Inspur

CN12700 Series

INOS Security Configuration Guide

（Release 8.x）

Inspur-Cisco Networking Technology Co.,Ltd. provides customers with comprehensive technical support and services. For any assistance, please contact our local office or company headquarters.
Website: http://www.inspur.com/
Technical Support Tel: 400-691-1766
Technical Support Email:icnt_service@inspur.com
Technical Document Support Email: icnt_service@inspur.com
Address: 1036 Langchao Road, Lixia District, Jinan City, Shandong Province
Postal code: 250101

Notice

Preface

Objectives

This guide describes main functions of the CN12700 Series. To have a quick grasp of the CN12700 Series, please read this manual carefully.

Versions

The following table lists the product versions related to this document.

Product name	Version
CN12700 Series	

Conventions

Symbol conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
Warning	Indicates a hazard with a medium or low level of risk which, if not avoided, could result in minor or moderate injury.
Caution	Indicates a potentially hazardous situation that, if not avoided, could cause equipment damage, data loss, and performance degradation, or unexpected results.
Note	Provides additional information to emphasize or supplement important points of the main text.
Tip	Indicates a tip that may help you solve a problem or save time.

General conventions

Convention	Description
Boldface	Names of files, directories, folders, and users are in **boldface**. For example, log in as user **root**.

Convention	Description
Italic	Book titles are in *italics*.
`Lucida Console`	Terminal display is in `Lucida Console`.

Command conventions

Convention	Description
Boldface	The keywords of a command line are in **boldface**.
Italic	Command arguments are in *italics*.
[]	Items (keywords or arguments) in square brackets [ ] are optional.
{ x | y | ... }	Alternative items are grouped in braces and separated by vertical bars. One is selected.
[x | y | ...]	Optional alternative items are grouped in square brackets and separated by vertical bars. One or none is selected.
{ x | y | ... } *	Alternative items are grouped in braces and separated by vertical bars. A minimum of one or a maximum of all can be selected.
[x | y | ...] *	The parameter before the & sign can be repeated 1 to n times.

GUI conventions

Convention	Description
Boldface	Buttons, menus, parameters, tabs, windows, and dialog titles are in **boldface**. For example, click **OK**.
>	Multi-level menus are in boldface and separated by the ">" signs. For example, choose **File** > **Create** > **Folder**.

Keyboard operation

Format	Description
Key	Press the key. For example, press **Enter** and press **Tab**.
Key 1+Key 2	Press the keys concurrently. For example, pressing **Ctrl+C** means the two keys should be pressed concurrently.
Key 1, Key 2	Press the keys in turn. For example, pressing **Alt**, **A** means the two keys should be pressed in turn.

Mouse operation

Action	Description
Click	Select and release the primary mouse button without moving the pointer.
Double-click	Press the primary mouse button twice continuously and quickly without moving the pointer.
Drag	Press and hold the primary mouse button and move the pointer to a certain position.

Change history

Updates between document versions are cumulative. Therefore, the latest document version contains all updates made to previous versions.

Issue 01 (2020-02-24)

Initial commercial release

Contents

Preface......iii
Objectives......iii
Versions......iii
Conventions......iii
Change history......v
CHAPTER 1 Overview......1
1.1 Authentication, Authorization, and Accounting......1
1.2 RADIUS and TACACS+ Security Protocols......2
1.3 SSH and Telnet......2
1.4 PKI......2
1.5 User Accounts and Roles......3
1.6 802.1X......3
1.7 NAC......3
1.8 Inspur TrustSec......3
1.9 IP ACLs......4
1.10 MAC ACLs......4
1.11 VACLs......4
1.12 Port Security......4
1.13 DHCP Snooping......4
1.14 Dynamic ARP Inspection......5
1.15 IP Source Guard......5
1.16 Keychain Management......5
1.17 Unicast RPF......6
1.18 Traffic Storm Control......6
1.19 Control Plane Policing......6
1.20 Rate Limits......6
CHAPTER 2 Configuring AAA......7
2.1 Finding Feature Information......7

2.2 Information About AAA........ 7
2.3 Licensing Requirements forAAA........ 10
2.4 Prerequisites for AAA........ 10
2.5 Guidelines and Limitations for AAA........ 11
2.6 Default Settings for AAA........ 11
2.7 Configuring AAA........ 11
2.8 Monitoring and Clearing the Local AAA Accounting Log........ 27
2.9 VerifyingtheAAAConfiguration........ 27
2.10 ConfigurationExamplesforAAA........ 28
2.11 Additional References for AAA........ 28
2.12 FeatureHistoryforAAA........ 28
CHAPTER 3 Configuring RADIUS........ 30
3.1 Finding Feature Information........ 30
3.2 Information About RADIUS........ 30
3.3 VirtualizationSupportforRADIUS........ 33
3.4 Licensing Requirements forRADIUS........ 34
3.5 Prerequisites forRADIUS........ 34
3.6 GuidelinesandLimitationsforRADIUS........ 34
3.7 DefaultSettingsforRADIUS........ 34
3.8 Configuring RADIUS Servers........ 34
3.9 Verifying the RADIUS Configuration........ 51
3.10 Monitoring RADIUSServers........ 51
3.11 Clearing RADIUS Server Statistics........ 52
3.12 ConfigurationExampleforRADIUS........ 52
3.13 WheretoGoNext........ 53
3.14 Additional References forRADIUS........ 53
3.15 Feature History for RADIUS........ 53
CHAPTER 4 Configuring TACACS+........ 55
4.1 Finding Feature Information........ 55
4.2 Information About TACACS+........ 55
4.3 Licensing Requirements for TACACS+........ 58

4.4 Prerequisites for TACACS+....59
4.5 Guidelines and Limitations forTACACS+....59
4.6 Default Settings for TACACS+....59
4.7 Configuring TACACS+.... 60
4.8 Monitoring TACACS+ Servers....82
4.9 Clearing TACACS+ Server Statistics....82
4.10 VerifyingtheTACACS+Configuration....83
4.11 Configuration Examples forTACACS+.... 83
4.12 WheretoGoNext.... 84
4.13 Additional References for TACACS+....84
4.14 Feature History for TACACS+....84
CHAPTER 5 Configuring LDAP.... 85
5.1 Finding Feature Information.... 85
5.2 Information About LDAP.... 85
5.3 Licensing Requirements forLDAP....88
5.4 Prerequisites forLDAP....88
5.5 Guidelines and Limitations for LDAP....88
5.6 Default Settings for LDAP.... 88
5.7 Configuring LDAP....89
5.8 Monitoring LDAPServers....102
5.9 Clearing LDAP Server Statistics.... 103
5.10 VerifyingtheLDAPConfiguration....103
5.11 Configuration Examples for LDAP....104
5.12 WheretoGoNext.... 104
5.13 Additional References forLDAP....105
5.14 FeatureHistoryforLDAP....105
CHAPTER 6 Configuring SSH and Telnet....106
6.1 Finding Feature Information.... 106
6.2 InformationAboutSSHandTelnet.... 106
6.3 Virtualization Support for SSH and Telnet.... 107
6.4 Licensing Requirements for SSH andTelnet....107

6.5 Prerequisites for SSH andTelnet....107
6.6 Guidelines and Limitations for SSH and Telnet....108
6.7 Default Settings for SSH and Telnet....108
6.8 Configuring SSH....108
6.9 Configuring Telnet....117
6.10 Verifying the SSH and Telnet Configuration....119
6.11 Configuration Example for SSH....119
6.12 Additional References for SSH and Telnet....120
CHAPTER 7 Configuring PKI....122
7.1 Finding Feature Information....122
7.2 Information About PKI....122
7.3 VirtualizationSupportforPKI....125
7.4 Licensing Requirements forPKI....125
7.5 Guidelines and Limitations for PKI....125
7.6 Default Settings for PKI....126
7.7 Configuring CAs and Digital Certificates....126
7.8 VerifyingthePKIConfiguration....140
7.9 Configuration Examples for PKI....140
7.10 Additional References forPKI....161
CHAPTER 8 Configuring User Accounts and RBAC....162
8.1 Finding Feature Information....162
8.2 InformationAboutUserAccountsandRBAC....162
8.3 VirtualizationSupportforRBAC....164
8.4 Licensing Requirements for User Accounts and RBAC....165
8.5 Guidelines and Limitations for User Accounts and RBAC....165
8.6 Default Settings for User Accounts and RBAC....165
8.7 Enabling Password-Strength Checking....166
8.8 Configuring User Accounts....167
8.9 Configuring Roles....168
8.10 Verifying User Accounts and RBAC Configuration....179
8.11 Configuration Examples for User Accounts and RBAC....180

8.12 Additional References for User Accounts and RBAC......181
8.13 Feature History for User Accounts and RBAC......182
CHAPTER 9 Configuring 802.1X...... 183
9.1 Finding Feature Information...... 183
9.2 InformationAbout802.1X......183
9.3 Licensing Requirements for 802.1X......188
9.4 Prerequisites for 802.1X......188
9.5 802.1X Guidelines and Limitations...... 189
9.6 Default Settings for 802.1X......189
9.7 Configuring 802.1X...... 190
9.8 Verifying the 802.1X Configuration...... 207
9.9 Monitoring802.1X......207
9.10 Configuration Example for 802.1X...... 208
9.11 Additional References for 802.1X...... 208
9.12 Feature History for802.1X...... 209
CHAPTER 10 Configuring NAC......210
10.1 Finding Feature Information...... 210
10.2 Information About NAC......210
10.3 VirtualizationSupportforNAC......218
10.4 Licensing Requirements forNAC...... 218
10.5 Prerequisites forNAC...... 218
10.6 NAC Guidelines andLimitations...... 219
10.7 DefaultSettingsforNAC......219
10.8 Configuring NAC......219
10.9 VerifyingtheNACConfiguration...... 245
10.10 ConfigurationExampleforNAC...... 246
10.11 Additional References forNAC...... 246
CHAPTER 11 Configuring Inspur TrustSec...... 247
11.1 Finding Feature Information...... 247
11.2 InformationAboutInspurTrustSec...... 247
11.3 Virtualization Support......265

11.4 Licensing Requirements for Inspur TrustSec.......... 265
11.5 Prerequisites for Inspur TrustSec.......... 266
11.6 Guidelines and Limitations for Inspur TrustSec.......... 266
11.7 Default Settings for Inspur TrustSec Parameters.......... 269
11.8 Configuring Inspur TrustSec.......... 270
11.9 Inspur TrustSec Support on Port-Channel Members.......... 321
11.10 Verifying the Inspur TrustSec Configuration.......... 322
11.11 Configuration Examples for Inspur TrustSec.......... 323
11.12 Troubleshooting Inspur TrustSec.......... 326
11.13 Additional References for Inspur TrustSec.......... 327
11.14 Feature History for Inspur TrustSec.......... 327
CHAPTER 12 Configuring Inspur TrustSec MACSec.......... 330
12.1 Finding Feature Information.......... 330
12.2 Information About MACsec.......... 330
12.3 Prerequisites for Inspur TrustSec MACSec.......... 336
12.4 Licensing Requirements for Inspur TrustSec MACSec.......... 336
12.5 Default Settings for Inspur TrustSec Parameters.......... 337
12.6 Feature History for Inspur TrustSec MACSec.......... 337
12.7 Guidelines and Limitations for Inspur TrustSec MACSec.......... 338
12.8 Configuring Inspur TrustSec MACSec.......... 338
12.9 Inspur TrustSec Support on Port-Channel Members.......... 353
12.10 Verifying the Inspur TrustSec MACSec Configuration.......... 354
12.11 Additional References for Inspur TrustSec MACSec.......... 354
CHAPTER 13 Configuring MACsec Key Agreement.......... 355
13.1 Finding Feature Information.......... 355
13.2 Information About MACsec.......... 355
13.3 Feature History for MKA.......... 361
13.4 Default Settings for MKA.......... 361
13.5 Guidelines and Limitations for MKA.......... 361
13.6 Configuring MKA.......... 362
13.7 Configuring a Non-standard Ethernet Type Value for EAPOL.......... 369

13.8 Configuring a Non-standard DMAC Address Value for EAPOL....370
13.9 Displaying MKA Statistics and Capability.... 372
13.10 Additional References for MKA.... 374
CHAPTER 14 Configuring IP ACLs.... 375
14.1 Finding Feature Information.... 375
14.2 Information About ACLs....375
14.3 Licensing Requirements for IP ACLs....389
14.4 Prerequisites for IP ACLs....389
14.5 Guidelines and Limitations for IP ACLs.... 390
14.6 Default Settings for IP ACLs.... 392
14.7 Configuring IP ACLs....392
14.8 Verifying the IP ACL Configuration.... 402
14.9 Monitoring and Clearing IP ACL Statistics.... 403
14.10 Configuration Examples for IP ACLs....403
14.11 Configuring Object Groups.... 403
14.12 Verifying the Object-Group Configuration.... 408
14.13 Configuring Time Ranges.... 408
14.14 Verifying the Time-Range Configuration.... 413
14.15 Troubleshooting Flexible ACL TCAM Bank Chaining....413
14.16 Additional References for IP ACLs.... 414
14.17 Feature History for IP ACLs.... 415
CHAPTER 15 Configuring MAC ACLs....416
15.1 Finding Feature Information.... 416
15.2 Information About MAC ACLs.... 416
15.3 Licensing Requirements for MAC ACLs.... 417
15.4 Prerequisites for MAC ACLs.... 417
15.5 Guidelines and Limitations for MAC ACLs....417
15.6 Default Settings for MAC ACLs....417
15.7 Configuring MAC ACLs.... 417
15.8 Verifying the MAC ACL Configuration....423
15.9 Monitoring and Clearing MAC ACL Statistics....424

15.10 Configuration Example for MAC ACLs........ 424
15.11 Additional References for MACACLs........424
15.12 Feature History for MAC ACLs........424
CHAPTER 16 Configuring VLAN ACLs........ 426
16.1 Finding Feature Information........ 426
16.2 InformationAboutVLANACLs........ 426
16.3 Licensing Requirements forVACLs........427
16.4 Prerequisites for VACLs........427
16.5 Guidelines and Limitations for VACLs........427
16.6 Default Settings forVACLs........428
16.7 Configuring VACLs........ 428
16.8 VerifyingtheVACLConfiguration........ 432
16.9 Monitoring and Clearing VACL Statistics........ 432
16.10 ConfigurationExampleforVACLs........432
16.11 Additional References forVACLs........432
16.12 Feature History for VLAN ACLs........ 433
CHAPTER 17 Configuring PortSecurity........434
17.1 Finding Feature Information........ 434
17.2 Information About Port Security........ 434
17.3 Licensing Requirements for Port Security........ 441
17.4 Prerequisites for Port Security........ 441
17.5 Default Settings for Port Security........441
17.6 Guidelines and Limitations for Port Security........442
17.7 Configuring PortSecurity........442
17.8 Verifying the Port Security Configuration........452
17.9 Displaying Secure MAC Addresses........ 453
17.10 ConfigurationExampleforPortSecurity........ 453
17.11 FeatureHistoryforPortSecurity........ 453
CHAPTER 18 Configuring DHCP........454
18.1 Finding Feature Information........ 454
18.2 Information About DHCP Snooping........ 454

18.3 InformationAbouttheDHCPRelayAgent........457
18.4 Information About the DHCPv6 Relay Agent........459
18.5 InformationAboutDHCPResponseRedirect........459
18.6 VirtualizationSupportforDHCP........459
18.7 Licensing Requirements forDHCP........459
18.8 Prerequisites forDHCP........460
18.9 Guidelines and Limitations for DHCP........460
18.10 Default Settings for DHCP........461
18.11 Configuring DHCP........461
18.12 ConfiguringDHCPv6........474
18.13 Configuring DHCP ResponseRedirect........478
18.14 VerifyingtheDHCPConfiguration........479
18.15 Displaying DHCP Bindings........479
18.16 Clearing the DHCP Snooping BindingDatabase........479
18.17 Clearing DHCP Relay Statistics........480
18.18 Clearing DHCPv6 Relay Statistics........480
18.19 Monitoring DHCP........481
18.20 Additional References for DHCP........481
18.21 Feature History for DHCP........481
CHAPTER 19 Configuring IPv6 First-Hop Security........483
19.1 Finding Feature Information........483
19.2 IntroductiontoFirst-HopSecurity........483
19.3 RAGuard........484
19.4 DHCPv6 Guard........484
19.5 IPv6Snooping........485
19.6 How to Configure IPv6 FHS........485
19.7 Configuration Examples........492
19.8 Additional References for IPv6 First-Hop Security........493
19.9 Feature History for IPv6 First-Hop Security........493
CHAPTER 20 Configuring Dynamic ARPInspection........494
20.1 Finding Feature Information........494

20.2 Information About DAI....494
20.3 VirtualizationSupportforDAI.... 497
20.4 Licensing Requirements for DAI....497
20.5 Prerequisites forDAI....498
20.6 GuidelinesandLimitationsforDAI.... 498
20.7 DefaultSettingsforDAI.... 498
20.8 Configuring DAI....499
20.9 VerifyingtheDAIConfiguration....504
20.10 Monitoring and Clearing DAIStatistics.... 504
20.11 ConfigurationExamplesforDAI....504
20.12 Configuring ARP ACLs....510
20.13 Verifying the ARP ACL Configuration....514
20.14 Additional References for DAI.... 514
20.15 Feature History for DAI....515

CHAPTER 21 Configuring IP SourceGuard....516

21.1 Finding Feature Information.... 516
21.2 InformationAboutIPSourceGuard....516
21.3 Licensing Requirements for IP Source Guard....517
21.4 Prerequisites for IP SourceGuard.... 517
21.5 Guidelines and Limitations for IP Source Guard.... 517
21.6 Default Settings for IP Source Guard.... 517
21.7 Configuring IP SourceGuard.... 518
21.8 Displaying IP Source Guard Bindings.... 519
21.9 Configuration Example for IP Source Guard....519
21.10 Additional References for IP Source Guard....520

CHAPTER 22 ConfiguringKeychainManagement....521

22.1 Finding Feature Information.... 521
22.2 Information About Keychain Management.... 521
22.3 Virtualization Support for Keychain Management....522
22.4 LicensingRequirementsforKeychainManagement.... 522
22.5 PrerequisitesforKeychainManagement....522

22.6 GuidelinesandLimitationsforKeychainManagement....522

22.7 DefaultSettingsforKeychainManagement.... 522

22.8 ConfiguringKeychainManagement.... 523

22.9 DeterminingActiveKeyLifetimes.... 529

22.10 Verifying the Keychain Management Configuration....529

22.11 ConfigurationExampleforKeychainManagement.... 529

22.12 WheretoGoNext.... 530

22.13 AdditionalReferencesforKeychainManagement.... 530

CHAPTER 23 ConfiguringTrafficStormControl.... 531

23.1 Finding Feature Information.... 531

23.2 Information About Traffic Storm Control.... 531

23.3 VirtualizationSupportforTrafficStormControl....532

23.4 LicensingRequirementsforTrafficStormControl.... 532

23.5 GuidelinesandLimitationsforTrafficStormControl.... 533

23.6 DefaultSettingsforTrafficStormControl....533

23.7 ConfiguringTrafficStormControl.... 533

23.8 Verifying Traffic Storm Control Configuration.... 534

23.9 MonitoringTrafficStormControlCounters.... 535

23.10 Configuration Example for Traffic Storm Control.... 535

23.11 AdditionalReferencesforTrafficStormControl....535

23.12 Feature History for Traffic Storm Control....535

CHAPTER 24 Configuring Unicast RPF.... 536

24.1 Finding Feature Information.... 536

24.2 Information About Unicast RPF....536

24.3 VirtualizationSupportforUnicastRPF....537

24.4 Licensing Requirements for UnicastRPF.... 537

24.5 Guidelines and Limitations for Unicast RPF....538

24.6 Default Settings for Unicast RPF....538

24.7 Configuring Unicast RPF.... 538

24.8 Configuration Examples for Unicast RPF.... 540

24.9 VerifyingtheUnicastRPFConfiguration....540

24.10 Additional References for Unicast RPF....541

24.11 Feature History for Unicast RPF....541

CHAPTER 25 Configuring Control Plane Policing.... 542

25.1 Finding Feature Information.... 542

25.2 Information About CoPP.... 542

25.3 Licensing Requirements forCoPP....556

25.4 Guidelines and Limitations for CoPP.... 556

25.5 Default Settings for CoPP....557

25.6 Configuring CoPP.... 557

25.7 Verifying the CoPP Configuration.... 565

25.8 Displaying the CoPP Configuration Status.... 566

25.9 Monitoring CoPP....566

25.10 Monitoring CoPP with SNMP....569

25.11 Clearing the CoPP Statistics.... 569

25.12 Configuration Examples for CoPP.... 570

25.13 Changing or Reapplying the Default CoPP Policy Using the Setup Utility....573

25.14 Additional References forCoPP....574

25.15 Feature History for CoPP.... 574

CHAPTER 26 Configuring RateLimits....576

26.1 Finding Feature Information.... 576

26.2 Information About Rate Limits.... 576

26.3 Virtualization Support for Rate Limits....576

26.4 Licensing Requirements for Rate Limits....577

26.5 Guidelines and Limitations for Rate Limits.... 577

26.6 Default Settings for Rate Limits....577

26.7 Configuring RateLimits....577

26.8 Monitoring Rate Limits.... 580

26.9 Clearing the Rate Limit Statistics.... 581

26.10 Verifying the Rate Limit Configuration....581

26.11 Configuration Examples for Rate Limits....581

26.12 Additional References for Rate Limits....581

26.13 Feature History for Rate Limits....582
CHAPTER 27 MonitoringSystemSecurity.... 583
27.1 Finding Feature Information.... 583
27.2 OverviewofSystemSecurityMonitoring.... 583
27.3 Additional References for Monitoring System Security....584
27.4 Feature History for Monitoring System Security....584
CHAPTER 28 Software IntegrityAssurance.... 586
28.1 Finding Feature Information.... 586
28.2 OverviewofRuntimeIntegrityAssurance.... 586
28.3 Additional References for Software Integrity Assurance....587
28.4 Feature History for Software Integrity Assurance....587

Figure

Figure 1： RADIUS Server States....32
Figure 2： TACACS+ Server States....57
Figure 3： Message Exchange....184
Figure 4： Posture Validation Devices.... 211
Figure 5： NAC Endpoint Device Posture Validation.... 212
Figure 6： Network Using LPIP Validation.... 214
Figure 7： NAC Device Roles....216
Figure 8： Inspur TrustSec Network Cloud Example.... 248
Figure 9： SA Protocol Negotiation....249
Figure 10： Inspur TrustSec Authentication.... 250
Figure 11： GACL Policy Example....253
Figure 12： SGT and SGACL in Inspur TrustSec Network....253
Figure 13： SGACL Policy Enforcement Per Interface Enabled....257
Figure 14： SGACL Policy Enforcement Per Interface Disabled....257
Figure 15： Using SXP to Propagate SGT Information....257
Figure 16： SXPv4 Loop Detection....260
Figure 17： Bidirectional SXP Connection.... 262
Figure 18： Example SXP Peer Connections.... 326
Figure 19： Inspur TrustSec Network Cloud Example.... 331
Figure 20： SA Protocol Negotiation....332
Figure 21： Inspur TrustSec Authentication.... 333
Figure 22： MACsec Frame.... 356
Figure 23： SecTag.... 357
Figure 24： MKA Encryption Process....358
Figure 25： Securing CE-to-CE.... 359
Figure 26： Securing DCI....360

Figure 27： Securing CE to multiple CEs.......360
Figure 28： Order of ACL Application.......377
Figure 29： ACLs and Packet Flow.......377
Figure 30： Suboption Packet Formats.......456
Figure 31： Circuit ID Suboption Frame Format for Regular and vPC Interfaces.......457
Figure 32： DHCP Relay Agent in a Metropolitan Ethernet Network.......458
Figure 33： ARP Cache Poisoning.......495
Figure 34： ARP Packet Validation on a VLAN Enabled for DAI.......496
Figure 35： Two Devices Supporting DAI.......505
Figure 36： One Device Supporting DAI.......509
Figure 37： Broadcast Suppression.......532
Figure 38： Runtime Integrity Assurance on Inspur CN12700 Series Switch.......586

Table

Table 1： AAA Service Configuration Commands........9
Table 2： AAA Authentication Methods for AAA Services........9
Table 3： Default AAA Parameter Settings........11
Table 4： MSCHAP and MSCHAP V2 RADIUS VSAs........16
Table 5： Feature History for AAA........28
Table 6： Default RADIUS Parameter Settings........34
Table 7： Feature History for RADIUS........53
Table 8： Default TACACS+ Parameters Settings........59
Table 9： Feature History for TACACS+........84
Table 10： LDAP Server States........87
Table 11： Default LDAP Parameters Settings........88
Table 12： Feature History for LDAP........105
Table 13： Default SSH and Telnet Parameters........108
Table 14： Default PKI Parameters........126
Table 15： Default User Accounts and RBAC Parameters........165
Table 16： Feature History for User Accounts and RBAC........182
Table 17： Default 802.1X Parameters........189
Table 18： Feature History for 802.1X........209
Table 19： Default NAC Parameter Settings........219
Table 20： SXP Version Negotiation Cases........259
Table 21： Licensing Requirements for Inspur TrustSec........266
Table 22： Support for port interoperability for the Inspur TrustSec-enabled links between the Inspur CN modules........269
Table 23： Default Inspur TrustSec Parameters Settings........269
Table 24： Feature History for Inspur TrustSec........327
Table 25： Default Inspur TrustSec Parameters Settings........337
Table 26： Feature History for Inspur TrustSec MACSec........337

Table 27： Supported EAPOL Packets with Ethernet Type and DMAC....361
Table 28： Feature History for MKA....361
Table 29： Default MKA Settings.... 361
Table 30： Security ACL Applications....376
Table 31： Protocol Number and Associated Layer 3 Protocol.... 391
Table 32： Default IP ACL Parameters.... 392
Table 33： Feature History for IP ACLs....415
Table 34： Default MAC ACLs Parameters.... 417
Table 35： Feature History for MAC ACLs....424
Table 36： Default VACL Parameters.... 428
Table 37： Feature History for VLAN ACLs....433
Table 38： Default Port Security Parameters....441
Table 39： Feature History for Port Security.... 453
Table 40： Default DHCP Parameters.... 461
Table 41： Feature History for DHCP.... 481
Table 42： Feature History for IPv6 First-Hop Security.... 493
Table 43： Default DAI Parameters....498
Table 44： Feature History for DAI....515
Table 45： Default IP Source Guard Parameters.... 517
Table 46： Feature History for IP Source Guard.... 520
Table 47： Default Keychain Management Parameters.... 523
Table 48： Default Traffic Storm Control Parameters....533
Table 49： Feature History for Traffic Storm Control....535
Table 50： Default Unicast RPF Parameter Settings....538
Table 51： Feature History for Unicast RPF.... 541
Table 52： Default CoPP Parameters Settings.... 557
Table 53： Feature History for CoPP....575
Table 54： Default Rate Limits Parameters Settings....577
Table 55： Feature History for Rate Limits.... 582
Table 56： Feature History for Monitoring System Security.... 584

Table 57： Feature History for Software Integrity Assurance.. 588

CHAPTER 1 Overview

The Inspur INOS software supports security features that can protect your network against degradation or failure and also against data loss or compromise resulting from intentional attacks and from unintended but damaging mistakes by well-meaning network users.

This chapter includes the following sections:

- Authentication, Authorization, and Accounting.
- RADIUS and TACACS+ Security Protocols.
- SSH and Telnet.
- PKI.
- User Accounts and Roles.
- 802.1X.
- NAC.
- Inspur TrustSec.
- IP ACLs.
- MAC ACLs.
- VACLs.
- Port Security.
- DHCP Snooping.
- Dynamic ARP Inspection.
- IP Source Guard.
- Keychain Management.
- Unicast RPF.
- Traffic Storm Control.
- Control Plane Policing.
- Rate Limits.

1.1 Authentication, Authorization, and Accounting

Authentication, authorization, and accounting (AAA) is an architectural framework for configuring a set of three independent security functions in a consistent, modular manner.

Authentication

Provides the method of identifying users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption. Authentication is the way a user is identified prior to being allowed access to the network and network services. You configure AAA authentication by defining a named list of authentication methods and then applying that list to various interfaces.

Authorization

Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet.

Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.

Accounting

Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes. Accounting enables you to track the services that users are accessing, as well as the amount of network resources that they are consuming.

Related Topics

Configuring AAA.

1.2 RADIUS and TACACS+ Security Protocols

AAA uses security protocols to administer its security functions. If your router or access server is acting as a network access server, AAA is the means through which you establish communication between your network access server and your RADIUS or TACACS+ security server.

The chapters in this guide describe how to configure the following security server protocols:

RADIUS

A distributed client/server system implemented through AAA that secures networks against unauthorized access. In the Inspur implementation, RADIUS clients run on Inspur routers and send authentication requests to a central RADIUS server that contains all user authentication and network service access information.

TACACS+

A security application implemented through AAA that provides a centralized validation of users who are attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation.

TACACS+ provides for separate and modular authentication, authorization, and accounting facilities.

Related Topics

Configuring RADIUS.
Configuring TACACS+.

1.3 SSH and Telnet

You can use the Secure Shell (SSH) server to enable an SSH client to make a secure, encrypted connection to a Inspur INOS device. SSH uses strong encryption for authentication. The SSH server in the Inspur INOS software can interoperate with publicly and commercially available SSH clients.

The SSH client in the Inspur INOS software works with publicly and commercially available SSH servers.

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

Related Topics

Configuring SSH and Telnet.

1.4 PKI

The Public Key Infrastructure (PKI) allows the device to obtain and use digital certificates for secure communication in the network and provides manageability and scalability for applications, such as SSH, that support digital certificates.

Related Topics

Configuring PKI.

1.5 User Accounts and Roles

You can create and manage user accounts and assign roles that limit access to operations on the Inspur INOS device. Role-based access control (RBAC) allows you to define the rules for an assign role that restrict the authorization that the user has to access management operations.

Related Topics

Configuring User Accounts and RBAC.

1.6 802.1X

802.1 X defines a client-server-based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. The authentication server authenticates each client connected to an Inspur INOS device port.

Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

Related Topics

Configuring 802.1X.

1.7 NAC

Network Admission Control (NAC) allows you to check endpoint devices for security compliancy and vulnerability before these devices are allowed access to the network. This security compliancy check is referred to as *posture validation*. Posture validation allows you to prevent the spread of worms, viruses, and other rogue applications across the network.

NAC validates that the posture, or state, of endpoint devices complies with security policies before the devices can access protected areas of the network. For devices that comply with the security policies, NAC allows access to protected services in the network. For devices that do not comply with security policies, NAC restricts access to the network that is sufficient only for remediation, which checks the posture of the device again.

Related Topics

Configuring NAC.

1.8 Inspur TrustSec

The Inspur TrustSec security architecture builds secure networks by establishing clouds of trusted network devices. Each device in the cloud is authenticated by its neighbors. Communication on the links between devices in the cloud is secured with a combination of encryption, message integrity checks, and replay protection mechanisms. Inspur TrustSec also uses the device and user identification information acquired during authentication for classifying, or coloring, the packets as they enter the network. This packet classification is maintained by tagging packets on ingress to the Inspur TrustSec network so that they can be properly identified for the purpose of applying security and other policy criteria along the data path. The tag, also called the security group tag (SGT), allows the network to enforce the access control policy by enabling the endpoint device to act upon the SGT to filter traffic. Inspur TrustSec uses ingress tagging and egress filtering to enforce access control policy in as a conversation.

Related Topics

Configuring Inspur TrustSec

1.9 IP ACLs

IP ACLs are ordered sets of rules that you can use to filter traffic based on IPv4 information in the Layer 3 header of packets. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the Inspur INOS software determines that an IP ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether a packet is permitted or denied, or if there is no match, the Inspur INOS software applies the applicable default rule. The Inspur INOS software continues processing packets that are permitted and drops packets that are denied.

Related Topics

Configuring IP ACLs.

1.10 MAC ACLs

MAC ACLs are ACLs that filter traffic using the information in the Layer 2 header of each packet. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the Inspur INOS software determines that a MAC ACL applies to a packet, it tests the packet against the conditions of all rules. The first match determines whether a packet is permitted or denied, or if there is no match, the INOS software applies the applicable default rule. The Inspur INOS software continues processing packets that are permitted and drops packets that are denied.

Related Topics

Configuring MAC ACLs.

1.11 VACLs

A VLAN ACL (VACL) is one application of an IP ACL or MAC ACL. You can configure VACLs to apply to all packets that are routed into or out of a VLAN or are bridged within a VLAN. VACLs are strictly for security packet filtering and for redirecting traffic to specific physical interfaces. VACLs are not defined by direction (ingress or egress).

Related Topics

Configuring VLAN ACLs.

1.12 Port Security

Port security allows you to configure Layer 2 interfaces that allow inbound traffic from only a restricted set of MAC addresses. The MAC addresses in the restricted set are called secure MAC addresses. In addition, the device does not allow traffic from these MAC addresses on another interface within the same VLAN. The number of MAC addresses that the device can secure is configurable per interface.

Related Topics

Configuring Port Security.

1.13 DHCP Snooping

DHCP snooping acts like a firewall between untrusted hosts and trusted DHCP servers. DHCP snooping performs the following activities:

- Validates DHCP messages received from untrusted sources and filters out invalid messages.
- Builds and maintains the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.
- Uses the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

Dynamic ARP inspection (DAI) and IP Source Guard also use information stored in the DHCP snooping binding database.

Related Topics

Configuring DHCP.

1.14 Dynamic ARP Inspection

Dynamic ARP inspection (DAI) ensures that only valid ARP requests and responses are relayed. When DAI is enabled and properly configured, a Inspur INOS device performs these activities:

- Intercepts all ARP requests and responses on untrusted ports.
- Verifies that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination.
- Drops invalid ARP packets.

DAI can determine the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a DHCP snooping binding database. This database is built by DHCP snooping if DHCP snooping is enabled on the VLANs and on the device. If the ARP packet is received on a trusted interface, the device forwards the packet without any checks. On untrusted interfaces, the device forwards the packet only if it is valid.

Related Topics

Configuring Dynamic ARP Inspection.

1.15 IP Source Guard

IP Source Guard is a per-interface traffic filter that permits IP traffic only when the IP address and MAC address of each packet matches one of two sources of IP and MAC address bindings:

- Entries in the DHCP snooping binding table.
- Static IP source entries that you configure.

Filtering on trusted IP and MAC address bindings helps prevent attacks that rely on spoofing the IP address of a valid host. To circumvent IP Source Guard, an attacker would have to spoof both the IP address and the MAC address of a valid host.

Related Topics

Configuring IP Source Guard.

1.16 Keychain Management

Keychain management allows you to create and maintain keychains, which are sequences of keys (sometimes called shared secrets). You can use keychains with features that secure communications with other devices by using key-based authentication. The device allows you to configure multiple keychains.

Some routing protocols that support key-based authentication can use a keychain to implement a hitless key rollover for authentication.

Related Topics

Configuring Keychain Management.

1.17 Unicast RPF

The Unicast Reverse Path Forwarding (RPF) feature reduces problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of Denial-of-Service (DoS) attacks, including Smurf and Tribal Flood Network (TFN) attacks, can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. Unicast RPF deflects attacks by forwarding only the packets that have source addresses that are valid and consistent with the IP routing table.

Related Topics

Configuring Unicast RPF.

1.18 Traffic Storm Control

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming traffic over a 1-second interval. During this interval, the traffic level, which is a percentage of the total available bandwidth of the port, is compared with the traffic storm control level that you configured. When the ingress traffic reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the interval ends.

Related Topics

Configuring Traffic Storm Control.

1.19 Control Plane Policing

The Inspur INOS device provides control plane policing to prevent denial-of-service (DoS) attacks from impacting performance. The supervisor module of the Inspur INOS device has both the management plane and control plane and is critical to the operation of the network. Any disruption to the supervisor module would result in serious network outages. Excessive traffic to the supervisor module could overload it and slow down the performance of the entire Inspur INOS device. Attacks on the supervisor module can be of various types such as, denial-of-service (DoS) attacks that generate IP traffic streams to the control plane at a very high rate. These attacks result in the control plane spending a large amount of time in handling these packets, which makes the control plane unable to process genuine traffic.

Related Topics

Configuring Control Plane Policing

1.20 Rate Limits

Rate limits can prevent redirected packets for egress exceptions from overwhelming the supervisor module on a Inspur INOS device.

Related Topics

Configuring Rate Limits

CHAPTER 2 Configuring AAA

This chapter describes how to configure authentication, authorization, and accounting (AAA) on Inspur INOS devices.

This chapter includes the following sections:

- Finding Feature Information.
- Information About AAA.
- Licensing Requirements for AAA.
- Prerequisites for AAA.
- Guidelines and Limitations for AAA.
- Default Settings for AAA.
- Configuring AAA.
- Monitoring and Clearing the Local AAA Accounting Log.
- Verifying the AAA Configuration.
- Configuration Examples for AAA.
- Additional References for AAA.
- Feature History for AAA.

2.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

2.2 Information About AAA

This section includes information about AAA on Inspur INOS devices.

2.2.1 AAA Security Services

The AAA feature allows you to verify the identity of, grant access to, and track the actions of users managing a Inspur INOS device. Inspur INOS devices support Remote Access Dial-In User Service (RADIUS) or Terminal Access Controller Access Control System Plus (TACACS+) protocols.

Based on the user ID and password combination that you provide, Inspur INOS devices perform local authentication or authorization using the local database or remote authentication or authorization using one or more AAA servers. A preshared secret key provides security for communication between the Inspur INOS device and AAA servers. You can configure a common secret key for all AAA servers or for only a specific AAA server.

AAA security provides the following services:

Authentication

Identifies users, including login and password dialog, challenge and response, messaging support, and, depending on the security protocol that you select, encryption.

Authentication is the process of verifying the identity of the person or device accessing the Inspur INOS device, which is based on the user ID and password combination provided by the entity trying to access the Inspur INOS device. Inspur INOS devices allow you to perform local authentication (using the local lookup database) or remote authentication (using one or more RADIUS or TACACS+ servers).

Authorization

Provides access control.AAA authorization is the process of assembling a set of attributes that describe what the user is authorized to perform. Authorization in the Inspur INOS software is provided by attributes that are downloaded from AAA servers. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights with the appropriate user.

Accounting

Provides the method for collecting information, logging the information locally, and sending the information to the AAA server for billing, auditing, and reporting.

The accounting feature tracks and maintains a log of every management session used to access the Inspur INOS device. You can use this information to generate reports for troubleshooting and auditing purposes. You can store accounting logs locally or send them to remote AAA servers.

Related Topics

Configuring Command Authorization on TACACS+ Servers.

2.2.2 Benefits of Using AAA

AAA provides the following benefits:

- Increased flexibility and control of access configuration
- Scalability
- Standardized authentication methods, such as RADIUS and TACACS+
- Multiple backup devices

2.2.3 Remote AAA Services

Remote AAA services provided through RADIUS and TACACS+ protocols have the following advantages over local AAA services:

- It is easier to manage user password lists for each Inspur INOS device in the fabric.
- AAA servers are already deployed widely across enterprises and can be easily used for AAA services.
- You can centrally manage the accounting log for all Inspur INOS devices in the fabric.
- It is easier to manage user attributes for each Inspur INOS device in the fabric than using the local databases on the Inspur INOS devices.
- AAA Server Groups

You can specify remote AAA servers for authentication, authorization, and accounting using server groups. A server group is a set of remote AAA servers that implement the same AAA protocol. The purpose of a server group is to provide for failover servers in case a remote AAA server fails to respond. If the first remote server in the group fails to respond, the next remote server in the group is tried until one of the servers sends a response. If all the AAA servers in the server group fail to respond, then that server group option is considered a failure. If required, you can specify multiple server groups. If the Inspur INOS device encounters errors from the servers in the first group, it tries the servers in the next server group.

2.2.4 AAA Service Configuration Options

The AAA configuration in Inspur INOS devices is service based, which means that you can have separate AAA configurations for the following services:

- User Telnet or Secure Shell (SSH) login authentication
- Console login authentication
- Inspur TrustSec authentication
- 802.1X authentication
- Extensible Authentication Protocol over User Datagram Protocol (EAPoUDP) authentication for Network Admission Control (NAC)
- User management session accounting

- 802.1X accounting

This table provides the related CLI command for each AAA service configuration option.

Table 1：AAA Service Configuration Commands

AAA Service Configuration Option	Related Command
Telnet or SSH login	**aaa authentication login default**
Fallback to local authentication for the default login.	**aaa authentication login default fallback error local**
Console login	**aaa authentication login console**
Inspur TrustSec authentication	**aaa authentication cts default**
802.1X authentication	**aaa authentication dot1x default**
EAPoUDP authentication	**aaa authentication eou default**
User session accounting	**aaa accounting default**
802.1X accounting	**aaa accounting dot1x default**

You can specify the following authentication methods for the AAA services:

All RADIUS servers

Uses the global pool of RADIUS servers for authentication.

Specified server groups Local

Uses the local username or password database for authentication.

None

Specifies that no AAA authentication be used.
This table shows the AAA authentication methods that you can configure for the AAA services.

Table 2：AAA Authentication Methods for AAA Services

AAA Service	AAA Methods
Console login authentication	Server groups, local, and none
User login authentication	Server groups, local, and none
Inspur TrustSec authentication	Server groups only
802.1X authentication	Server groups only
EAPoUDP authentication	Server groups only
User management session accounting	Server groups and local
802.1X accounting	Server groups and local

Note	For console login authentication, user login authentication, and user management session accounting, the Inspur INOS device tries each option in the order specified. The local option is the default method when other configured options fail.

Related Topics

Configuring Inspur TrustSec Configuring 802.1X.

Configuring NAC.

2.2.5 Authentication and Authorization Process for User Login

The following list explains the process:

- When you log in to the required Inspur INOS device, you can use the Telnet, SSH, or console login options.
- When you have configured the AAA server groups using the server group authentication method, the Inspur INOS device sends an authentication request to the first AAA server in the group as follows:
- If the AAA server fails to respond, the next AAA server is tried and so on until the remote server responds to the authentication request.
- If all AAA servers in the server group fail to respond, the servers in the next server group are tried.
- If all configured methods fail, the local database is used for authentication.
- If the Inspur INOS device successfully authenticates you through a remote AAA server, then the following possibilities apply:
- If the AAA server protocol is RADIUS, then user roles specified in the Inspur-av-pair attribute are downloaded with an authentication response.
- If the AAA server protocol is TACACS+, then another request is sent to the same server to get the user roles specified as custom attributes for the shell.
- If the user roles are not successfully retrieved from the remote AAA server, then the user is assigned with the vdc-operator role.
- If your username and password are successfully authenticated locally, the Inspur INOS device logs you in and assigns you the roles configured in the local database.

2.2.6 Virtualization Support for AAA

All AAA configuration and operations are local to the virtual device context (VDC), except the default console methods and the AAA accounting log. The configuration and operation of the AAA authentication methods for the console login apply only to the default VDC. The AAA accounting log is only in the default VDC. You can display the contents from any VDC, but you must clear it in the default VDC.

2.3 Licensing Requirements forAAA

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS	AAA requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

2.4 Prerequisites for AAA

Remote AAA servers have the following prerequisites:

- Ensure that the Inspur INOS device is configured as a client of the AAA servers.
- Ensure that the secret key is configured on the Inspur INOS device and the remote AAA servers.
- Ensure that the remote server responds to AAA requests from the Inspur INOS device.

Related Topics

Configuring RADIUS Server Hosts.
Configuring TACACS+ Server Hosts.
Manually Monitoring RADIUS Servers or Groups.

Manually Monitoring TACACS+ Servers or Groups.

2.5 Guidelines and Limitations for AAA

AAA has the following guidelines and limitations:

- If you have a user account configured on the local Inspur INOS device that has the same name as a remote user account on an AAA server, the Inspur INOS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.

2.6 Default Settings for AAA

This table lists the default settings for AAA parameters.

Table 3：Default AAA Parameter Settings

Parameters	Default
Console authentication method	local
Default authentication method	local
Login authentication failure messages	Disabled
MSCHAP authentication	Disabled
Default accounting method	local
Accounting log display length	250 KB

2.7 Configuring AAA

This section describes the tasks for configuring AAA on Inspur INOS devices.

2.7.1 Process for Configuring AAA

Follow these steps to configure AAA authentication and accounting:

1.
2. Configure console login authentication methods.
3. Configure default login authentication methods for user logins.
4. Configure default AAA accounting default methods.

Related Topics

Configuring RADIUS.
Configuring TACACS+.
Configuring Console Login Authentication Methods.
Configuring Default Login Authentication Methods.
Configuring AAA Accounting Default Methods.
Configuring AAA Authentication Methods for 802.1X.
Enabling the Default AAA Authenication Method for EAPoUDP.

2.7.2 Configuring Console Login Authentication Methods

This section describes how to configure the authentication methods for the console login. The authentication methods include the following:

- Global pool of RADIUS servers
- Local database on the Inspur INOS device
- Username only (none) The default method is local.

Before you begin

Ensure that you are in the default VDC.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**}
3. **exit**
4. (Optional) **show aaa authentication**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**} **Example:** `switch(config)# aaa authentication login console group radius`	Configures login authentication methods for the console. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: **radius** Uses the global pool of RADIUS servers for authentication. The **local** method uses the local database for authentication, and the **none** method specifies that no AAA authentication be used. The default console login method is **local**, which is used when no methods are configured or when all the configured methods fail to respond.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show aaa authentication** **Example:** `switch# show aaa authentication`	Displays the configuration of the console login authentication methods.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Groups.
Configuring TACACS+ Server Groups.

2.7.3 Configuring Default Login Authentication Methods

The authentication methods include the following:

- Global pool of RADIUS servers
- Local database on the Inspur INOS device
- Username only

The default method is local.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authentication login default** { **fallback error local** |**group** *group-list* [**none**] | **local** |**none**}
3. **exit**
4. (Optional) **show aaa authentication**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**aaa authentication login default** { **fallback error local** |**group** *group-list* [**none**] | **local** | **none**} **Example:** `switch(config)# aaa authentication login default group radius`	Configures the default authentication methods. The **fallback error local** enables fallback to local authentication for the default login if remote authentication is configured and all AAA servers are unreachable. Fallback to local authentication is enabled by default. **Note** Disabling fallback to local authentication can lock your Inspur INOS device, forcing you to perform a password recovery in order to gain access. To prevent being locked out of the device, we recommend disabling fallback to local authentication for only the default login or the console login, not both. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius**—Uses the global pool of RADIUS servers for authentication. The **local** method uses the local database for

	Command or Action	Purpose
		authentication, and the **none** method specifies that no AAA authentication be used. The default login method is **local**, which is used when no methods are configured or when all the configured methods fail to respond. You can configure one of the following: • AAA authentication groups • AAA authentication groups with no authentication • Local authentication • No authentication **Note** The **local** keyword is not supported (and is not required) when configuring AAA authentication groups because local authentication is the default if remote servers are unreachable. For example, if you configure **aaa authentication login default group g1**, local authentication is tried if you are unable to authenticate using AAA group g1. In contrast, if you configure **aaa authentication login default group g1 none**, no authentication is performed if you are unable to authenticate using AAA group g1.

Step 3	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show aaa authentication** **Example:** `switch# show aaa authentication`	Displays the configuration of the default login authentication methods.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Groups.
Configuring TACACS+ Server Groups.

2.7.4 Enabling the Default User Role for AAA Authentication

You can allow remote users who do not have a user role to log in to the Inspur INOS device through a RADIUS or TACACS+ remote authentication server using a default user role. When you disable the AAA default user role feature, remote users who do not have a user role cannot log in to the device.

You can enable or disable this feature for the VDC as needed. For the default VDC, the default role is network-operator. For nondefault VDCs, the default VDC is vdc-operator.

Before you begin

Make sure that you are in the correct VDC. To switch VDCs, use the **switchto vdc** command.

SUMMARY STEPS

1. **configure terminal**
2. **aaa user default-role**
3. **exit**
4. (Optional) **show aaa user default-role**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**aaa user default-role** **Example:** `switch(config)# aaa user default-role`	Enables the default user role for AAA authentication. The default is enabled. You can disable the default user role feature by using the **no** form of this command.

Step 3	**exit**
	Example:
	`switch(config)#`
	`exit switch#`
		Exits configuration mode.

Step 4	(Optional) **show aaa user default-role** **Example:** `switch# show aaa user default-role`	Displays the AAA default user role configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring User Accounts and RBAC.

2.7.5 Enabling Login Authentication Failure Messages

When you log in, the login is processed by rolling over to the local user database if the remote AAA servers do not respond. In such cases, the following messages display on the user's terminal if you have enabled login failure messages:

```
Remote AAA servers unreachable; local authentication done.
Remote AAA servers unreachable; local authentication
failed.
```

Before you begin

Make sure that you are in the correct VDC. To switch VDCs, use the **switchto vdc** command.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authentication login error-enable**
3. **exit**
4. (Optional) **show aaa authentication**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure erminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**aaa authentication login error-enable** **Example:** `switch(config)# aaa authentication login error-enable`	Enables login authentication failure messages. The default is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show aaa authentication** **Example:** `switch# show aaa authentication`	Displays the login failure message configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

2.7.6 Enabling MSCHAP or MSCHAP V2 Authentication

Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is the Microsoft version of CHAP. The Inspur INOS software also supports MSCHAP Version 2 (MSCHAP V2). You can use MSCHAP for user logins to a Inspur INOS device through a remote authentication server (RADIUS or TACACS+). MSCHAP V2 only supports user logins to a Inspur INOS device through remote authentication RADIUS servers. If you configure a TACACS+ group with MSCHAP V2, the AAA default login authentication uses the next configured method, or the local method, if no other server group is configured.

By default, the Inspur INOS device uses Password Authentication Protocol (PAP) authentication between the Inspur INOS device and the remote server. If you enable MSCHAP or MSCHAP V2, you need to configure your RADIUS server to recognize the MSCHAP and MSCHAP V2 vendor-specific attributes (VSAs).

This table shows the RADIUS VSAs required for MSCHAP.

Table 4：MSCHAP and MSCHAP V2 RADIUS VSAs

Vendor-ID Number	Vendor-Type Number	VSA	Description

311	11	MSCHAP-Challenge	Contains the challenge sent by an AAA server to an MSCHAP or MSCHAP V2 user. It can be used in both Access-Request and Access-Challenge packets.
211	11	MSCHAP-Response	Contains the response value provided by an MSCHAP or MSCHAP V2 user in response to the challenge. It is only used in Access-Request packets.

Before you begin

Disable AAA ASCII authentication for logins.

SUMMARY STEPS

1. **configure terminal**
2. **no aaa authentication login ascii-authentication**
3. **aaa authentication login** {**mschap** | **mschapv2**} **enable**
4. **exit**
5. (Optional) **show aaa authentication login** {**mschap** | **mschapv2**}
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**no aaa authentication login ascii-authentication** **Example:** `switch(config)# no aaa authentication login ascii-authentication`	Disables ASCII authentication.
Step 3	**aaa authentication login** {**mschap** | **mschapv2**} **enable** **Example:** `switch(config)# aaa authentication login mschap enable`	Enables MSCHAP or MSCHAP V2 authentication. The default is disabled. **Note** You cannot enable both MSCHAP and MSCHAP V2 on your Inspur INOS device.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show aaa authentication login** {**mschap** | **mschapv2**} **Example:** `switch# show aaa authentication login mschap`	Displays the MSCHAP or MSCHAP V2 configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Using AAA Server VSAs with Inspur INOS Devices.

2.7.7 Configuring AAA Accounting Default Methods

Inspur INOS software supports TACACS+ and RADIUS methods for accounting. Inspur INOS devices report user activity to TACACS+ or RADIUS security servers in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the AAA server.

When you activate AAA accounting, the Inspur INOS device reports these attributes as accounting records, which are then stored in an accounting log on the security server.

You can create default method lists defining specific accounting methods, which include the following:

RADIUS server group

Uses the global pool of RADIUS servers for accounting.

Specified server group

Uses a specified RADIUS or TACACS+ server group for accounting.

Local

Uses the local username or password database for accounting.

Before you begin

Configure RADIUS or TACACS+ server groups, as needed.

SUMMARY STEPS

1. **configure terminal**
2. **aaa accounting default** {**group** *group-list* |**local**}
3. **exit**
4. (Optional) **show aaa accounting**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**aaa accounting default** {**group** *group-list* |**local**} **Example:** `switch(config)# aaa accounting default group radius`	Configures the default accounting method. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius**—Uses the global pool of RADIUS servers for accounting. • *named-group*—Uses a named subset of TACACS+ or RADIUS servers for accounting.

	Command or Action	Purpose
		The **local** method uses the local database for accounting. The default method is **local**, which is used when no server groups are configured or when all the configured server groups fail to respond.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show aaa accounting** **Example:** `switch# show aaa accounting`	Displays the configuration AAA accounting default methods.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Groups.
Configuring TACACS+ Server Groups.

2.7.8 Using AAA Server VSAs with Inspur INOS Devices

You can use vendor-specific attributes (VSAs) to specify Inspur INOS user roles and SNMPv3 parameters on AAA servers.

About VSAs

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating VSAs between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Inspur RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Inspur vendor ID is 9, and the supported option is vendor type 1, which is named Inspur-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Inspur attribute for a particular type of authorization, the separator is = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use RADIUS servers for authentication on a Inspur INOS device, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

VSA Format

The following VSA protocol options are supported by the Inspur INOS software:

Shell

Protocol used in access-accept packets to provide user profile information.

Accounting

Protocol used in accounting-request packets. If a value contains any white spaces, put it within double quotation marks.

The following attributes are supported by the Inspur INOS software:

roles

Lists all the roles assigned to the user. The value field is a string that stores the list of group names delimited by white space. For example, if you belong to roles network-operator and vdc-admin, the value field would be network-operator vdc-admin. This subattribute is sent in the VSA portion of the

Access-Accept frames from the RADIUS server, and it can only be used with the shell protocol value. These examples use the roles attribute:

```
shell:roles=network-operator vdc-admin

shell:roles*network-operator vdc-admin
```

The following examples show the roles attribute as supported by FreeRADIUS:

```
Inspur-AVPair = shell:roles=\network-operator

vdc-admin\ Inspur-AVPair = shell:roles*\network-

operator vdc-admin\
```

accountinginfo

Stores accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch, and it can only be used with the accounting protocol-related PDUs.

Specifying Inspur INOS User Roles and SNMPv3 Parameters on AAA Servers

You can use the VSA Inspur-av-pair on AAA servers to specify user role mapping for the Inspur INOS device using this format:

```
shell:roles="roleA roleB …"
```

If you do not specify the role option in the Inspur-av-pair attribute, the default user role is network-operator. You can also specify your SNMPv3 authentication and privacy protocol attributes as follows:

```
shell:roles="roleA roleB..." snmpv3:auth=SHA priv=AES-128
```

The SNMPv3 authentication protocol options are SHA and MD5. The privacy protocol options are AES-128 and DES. If you do not specify these options in the Inspur-av-pair attribute, MD5 and DES are the default authentication protocols.

Related Topics

Configuring User Accounts and RBAC.

2.7.9 Secure Login Enhancements

The following secure login enhancements are supported in Inspur INOS:

Configuring Login Parameters

Use this task to configure your Inspur INOS device for login parameters that help detect suspected DoS attacks and slow down dictionary attacks.

All login parameters are disabled by default. You must enter the **login block-for** command, which enables default login functionality, before using any other login commands. After the **login block-for** command is enabled, the following default is enforced:

- All login attempts made through Telnet or SSH are denied during the quiet period; that is, no ACLs are exempt from the login period until the **login quiet-mode access-class** command is entered.

SUMMARY STEPS

1. **configure terminal**
2. **[no] login block-for** *seconds* **attempts** *tries* **within** *seconds*
3. **[no] login quiet-mode access-class** {*acl-name* | *acl-number*}
4. **exit**
5. **show login failures**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] login block-for** *seconds* **attempts** *tries* **within** *seconds* **Example:** `Switch(config)# login block-for 100 attempts 2 within 100`	Configures your Inspur INOS device for login parameters that help provide DoS detection. **Note** This command must be issued before any other login command can be used.
Step 3	**[no] login quiet-mode access-class** {*acl-name* | *acl-number*} **Example:** `Switch(config)# login quiet-mode access-class myacl`	(Optional) Although this command is optional, it is recommended that it be configured to specify an ACL that is to be applied to the device when the device switches to quiet mode. When the device is in quiet mode, all login requests are denied and the only available connection is through the console.
Step 4	**exit** **Example:** `Switch(config)# exit`	Exits to privileged EXEC mode.
Step 5	**show login failures** **Example:** `Switch# show login`	Displays login parameters. • **failures** --Displays information related only to failed login attempts.

Configuration Examples for Login Parameters

Setting Login Parameters Example

The following example shows how to configure your switch to enter a 100 second quiet period if 15 failed login attempts is exceeded within 100 seconds; all login requests are denied during the quiet period except hosts from the ACL "myacl."

```
Switch(config)# login block-for 100 attempts 15 within 100
Switch(config)# login quiet-mode access-class myacl
```

Showing Login Parameters Example

The following sample output from the **show login** command verifies that no login parameters have been specified:

```
Switch# show login


No Quiet-Mode access list has been configured, default ACL will be applied.

Switch is enabled to watch for login Attacks.
If more than 2 login failures occur in 45 seconds or less, logins will be disabled for 70
 seconds.

Switch presently in Normal-Mode.
Current Watch Window remaining time 10 seconds.
Present login failure count 0.
```

The following sample output from the **show login failures** command shows all failed login attempts on the switch:

```
Switch# show login failures

Information about last 20 login failures with the device.
--------------------------------------------------------------------------------
Username                                Line    Source               Appname
TimeStamp
--------------------------------------------------------------------------------
admin                                    pts/0    bgl-ads-728.Inspur.com login
        Wed Jun 10 04:56:16 2015
admin                                    pts/0    bgl-ads-728.Inspur.com login
        Wed Jun 10 04:56:19 2015
--------------------------------------------------------------------------------
```

The following sample output from the **show login failures** command verifies that no information is presently logged:

```
Switch# show login failures
*** No logged failed login attempts with the device.***
```

Configuring Login Block Per User

The Login Block Per User feature helps detect suspected Denial of Service (DoS) attacks and to slow down dictionary attacks. This feature is applicable only for local users. Use this task to configure login parameters to block an user after failed login attempts.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authentication rejected** *attempts* **in** *seconds* **ban** *seconds*
3. **exit**
4. **show running config**
5. **show aaa local user blocked**
6. **clear aaa local user blocked** {**username** *user* | **all**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**aaa authentication rejected** *attempts* **in** *seconds*	Configures login parameters to block an user.

	Command or Action	Purpose
	ban *seconds* **Example:** `switch(config)# aaa authentication rejected 3 in 20 ban 300`	**Note** Use the **no aaa authentication rejected** command to revert to the default login parameters.
Step 3	**exit** **Example:** `switch(config)# exit`	Exits to privileged EXEC mode.
Step 4	**show running config** **Example:** `switch# show running config`	(Optional) Displays the login parameters.
Step 5	**show aaa local user blocked** **Example:** `switch# show aaa local user blocked`	(Optional) Displays the blocked local users.
Step 6	**clear aaa local user blocked** {**username** *user* | **all**} **Example:** `switch# clear aaa local user blocked username testuser`	(Optional) Clears the blocked local users. • **all**–Clears all the blocked local users.

Configuration Examples for Login Block Per User

Setting Parameters for Login Block Per User

The following example shows how to configure the login parameters to block a user for 300 seconds when five login attempts fail within a period of 60 seconds:

```
switch(config)# aaa authentication rejected 5 in 60 ban 300
```

Showing Login Parameters

The following example shows the login parameters configured for a switch:

```
switch# show run | i rejected
aaa authentication rejected 5 in 60 ban 300
```

Showing Blocked Local Users

The following example shows the blocked local users:

```
switch# show aaa local user blocked
Local-user                    State

testuser                      Watched (till 11:34:42 IST Feb 5 2015)
```

Clearing Blocked Local Users

The following example shows how to clear the blocked local user testuser:

```
switch# clear aaa local user blocked username testuser
```

Restricting Sessions Per User—Per User Per Login

Use this task to restrict the maximum sessions per user.

SUMMARY STEPS

1. **configure terminal**
2. **[no] user max-logins** *max-logins*
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] user max-logins** *max-logins* **Example:** `Switch(config)# user max-logins 1`	Restricts the maximum sessions per user. The range is from 1 to 7. If you set the maximum login limit as 1, then only one session (telnet/SSH) is allowed per user.
Step 3	**exit** **Example:** `Switch(config)# exit`	Exits to privileged EXEC mode.

Configuring Passphrase and Locking User Accounts

Perform this task to configure passphrase lengths, time values, and locking user accounts.

SUMMARY STEPS

1. **userpassphrase** { **min-length** | **max-length** }
2. **userpassphrase** { **min-length** & **max-length** }
3. **show userpassphrase** {**min-length** | **max-length** | **length** }
4. **no userpassphrase** {**min-length** | **max-length** | **length** }
5. **show userpassphrase all**
6. **userpassphrase** { **default-lifetime** | **default-warntime** | **default-gracetime** }
7. **username** <*username*> **passphrase** { **lifetime** | **warntime** | **gracetime** }
8. **no username** <*username*> **passphrase** { **lifetime** | **warntime** | **gracetime** | **timevalues** }
9. **show username** <*username*> **passphrase timevalues**
10. **username** <*username*> **lock-user-account**
11. **username** <*username*> **expire-userpassphrase**
12. **show locked-users**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**userpassphrase** { **min-length** | **max-length** } **Example:** `Switch(config)# userpassphrase {  min-length`	Admin is allowed to configure either minimum or maximum passphrase length

	Command or Action	Purpose
	`<8 ? 127> | max-length <80 ? 127> }`	
Step 2	**userpassphrase** { **min-length** & **max-length** } **Example:** `Switch(config)# userpassphrase { min-length <8 ? 127> & max-length <80 ? 127> }`	Admin is allowed to configure both minimum and maximum passphrase length
Step 3	**show userpassphrase** {**min-length** | **max-length** | **length**} **Example:** `Switch(config)# show userpassphrase {min-length | max-length | length }`	Using min-length or max-length option, user is allowed to view either minimum or maximum passphrase length configuration .Using length option, they can view complete passphrase length configuration.
Step 4	**no userpassphrase** {**min-length** | **max-length** | **length** } **Example:** `Switch(config)# userpassphrase {min-length | max-length | length }`	To reset the passphrase length configuration to default configuration
Step 5	**show userpassphrase all** **Example:** `Switch(config)# show userpassphrase all`	To list all the parameter values under userpassphrase
Step 6	**userpassphrase** { **default-lifetime** | **default-warntime** | **default-gracetime** } **Example:** `Switch(config)# userpassphrase { default-lifetime | default-warntime | default-gracetime }`	Admin is allowed to update the default configurations
Step 7	**username** <*username*> **passphrase** { **lifetime** | **warntime** | **gracetime** } **Example:** `Switch(config)# username <user1> passphrase { lifetime | warntime | gracetime }`	Admin can configure passphrase lifetimes for any user
Step 8	**no username** <*username*> **passphrase** { **lifetime** | **warntime** | **gracetime** | **timevalues** } **Example:** `Switch(config)# username <user1> passphrase { lifetime | warntime | gracetime | timevalues }`	Admin can reset passphrase lifetimes to default values for any user
Step 9	**show username** <*username*> **passphrase timevalues** **Example:** `Switch(config)# show username <user1> passphrase timevalues`	Any user can view his/her passphrase lifetimes configured and admin can view for any user
Step 10	**username** <*username*> **lock-user-account** **Example:** `Switch(config)# username <user1> lock-user-account`	Admin can lock any user account

	Command or Action	Purpose
Step 11	**username** *<username>* **expire-userpassphrase** **Example:** `Switch(config)# username <user1> expire-userpassphrase`	Admin can set any userpassphrase to expire immediately
Step 12	**show locked-users** **Example:** `Switch(config)# show locked-users`	Admin can view and unlock all the locked users

Enabling the Password Prompt for User Name

SUMMARY STEPS

1. **configure terminal**
2. **[no] password prompt username**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] password prompt username** **Example:** `Switch(config)# password prompt username`	Enables the login knob. If this command is enabled and the user enters the **username** command without the password option, then the password is prompted. The password accepts hidden characters. Use the **no** form of this command to disable the login knob.
Step 3	**exit** **Example:** `Switch(config)# exit`	Exits to privileged EXEC mode.

Support over SHA-256 Algorithm for Verifying OS Integrity

Use the **show file bootflash:/ sha256sum** command to display the sha256sum of the file. The sample output for this command is shown below:

```
Switch# show file bootflash:/ sha256sum

abd9d40020538acc363df3d1bae7d1df16841e4903fca2c07c7898bf4f549ef5
```

Configuring Share Key Value for using RADIUS/TACACS+

The shared secret you configure for remote authentication and accounting must be hidden. For the **radius-server key** and **tacacs-server key** commands, a separate command to generate encrypted shared secret can be used.

SUMMARY STEPS

1. **configure terminal**
2. **generate type7_encrypted_secret**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 2	**generate type7_encrypted_secret** **Example:** `Switch(config)# generate ype7_encrypted_secret`	Configures RADIUS and TACACS shared secret with key type 7. While generating an encrypted shared secret, user input is hidden. **Note** You can generate encrypted equivalent of plain text separately and can configure the encrypted shared secret later.
Step 3	**exit** **Example:** `Switch(config)# exit`	Exits to privileged EXEC mode.

2.8 Monitoring and Clearing the Local AAA Accounting Log

The Inspur INOS device maintains a local log for the AAA accounting activity. You can monitor this log and clear it.

SUMMARY STEPS

1. **show accounting log** [*size* | **last-index** | **start-seqnum** *number* | **start-time** *year month day hh***:***mm***:***ss*]
2. (Optional) **clear accounting log**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show accounting log** [*size*	**last-index**	**start-seqnum** *number*	**start-time** *year month day hh***:***mm***:***ss*] **Example:** `switch# show accounting log`	Displays the accounting log contents. By default, the command output contains up to 250,000 bytes of the accounting log. You can use the *size* argument to limit command output. The range is from 0 to 250000 bytes. You can also specify a starting sequence number or a starting time for the log output.The range of the starting index is from 1 to 1000000. Use the **last-index** keyword to display the value of the last index number in the accounting log file.
Step 2	(Optional) **clear accounting log** **Example:** `switch# clear aaa accounting log`	Clears the accounting log contents.

2.9 Verifying the AAA Configuration

To display AAA configuration information, perform one of the following tasks:

Command	Purpose
show aaa accounting	Displays AAA accounting configuration.
show aaa authentication [**login** {**ascii-authentication** | **error-enable** | **mschap** | **mschapv2**}]	Displays AAA authentication login configuration information.
show aaa groups	Displays the AAA server group configuration.
show running-config aaa [**all**]	Displays the AAA configuration in the running configuration.
show startup-config aaa	Displays the AAA configuration in the startup configuration.

2.10 Configuration Examples for AAA

The following example shows how to configure AAA:

```
aaa authentication login default group radius
aaa authentication login console group radius
aaa accounting default group radius
```

2.11 Additional References for AAA

This section includes additional information related to implementing AAA.

Related Documents

Related Topic	Document Title
Inspur INOS Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*
SNMP	*Inspur CN12700 Series INOS System Management Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

2.12 Feature History for AAA

This table lists the release history for this feature.

Table 5： Feature History for AAA

Feature Name	Releases	Feature Information
Login Block Per User	8.2(3)	Added support for login block per user. Refer to the "Secure Login Enhancements" section.
Secure Login Enhancements	8.2(3)	Added enhancements for secure

Feature Name	Releases	Feature Information
		login. Refer to the "Secure Login Enhancements" section.
AAA	8.2(3)	No change from Release 8.2(3)
AAA	8.2(3)	Added support for the Inspur CN 3000 Series Switches.
AAA	8.2(3)	No change from Release 8.2(3)
AAA	8.2(3)	No change from Release 8.2(3)
AAA authentication	8.2(3)	Added support for enabling or disabling AAA authentication for user logins.
AAA authentication	8.2(3)	Added support for remote users who do not have a user role to log in to the Inspur INOS device through a RADIUS or TACACS+ remote authentication server using a default user role.
Login authentication	8.2(3)	Added support for enabling or disabling login authentication failure messages.
CHAP authentication	8.2(3)	Added support for enabling or disabling CHAP authentication.
Local authentication	8.2(3)	Added support for enabling fallback to local authentication when remote authentication fails.
Local authentication	8.2(3)	Added support for disabling fallback to local authentication.
MSCHAP V2 authentication	8.2(3)	Added support for enabling or disabling MSCHAP V2 authentication.
AAA	8.2(3)	No change from Release 8.2(3).

CHAPTER 3 Configuring RADIUS

This chapter describes how to configure the Remote Access Dial-In User Service (RADIUS) protocol on Inspur INOS devices.

This chapter includes the following sections:

- Finding Feature Information.
- Information About RADIUS.
- Virtualization Support for RADIUS.
- Licensing Requirements for RADIUS.
- Prerequisites for RADIUS.
- Guidelines and Limitations for RADIUS.
- Default Settings for RADIUS.
- Configuring RADIUS Servers.
- Verifying the RADIUS Configuration.
- Monitoring RADIUS Servers.
- Clearing RADIUS Server Statistics.
- Configuration Example for RADIUS.
- Where to Go Next .
- Additional References for RADIUS.
- Feature History for RADIUS.

3.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

3.2 Information About RADIUS

The RADIUS distributed client/server system allows you to secure networks against unauthorized access. In the Inspur implementation, RADIUS clients run on Inspur INOS devices and send authentication and accounting requests to a central RADIUS server that contains all user authentication and network service access information.

3.2.1 RADIUS Network Environments

RADIUS can be implemented in a variety of network environments that require high levels of security while maintaining network access for remote users.

You can use RADIUS in the following network environments that require access security:

- Networks with multiple-vendor network devices, each supporting RADIUS. For example, network devices from several vendors can use a single RADIUS server-based security database.
- Networks already using RADIUS. You can add a Inspur INOS device with RADIUS to the network. This action might be the first step when you make a transition to a AAA server.
- Networks that require resource accounting. You can use RADIUS accounting independent of RADIUS

authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, indicating the amount of resources (such as time, packets, bytes, and so on) used during the session. An Internet service provider (ISP) might use a freeware-based version of the RADIUS access control and accounting software to meet special security and billing needs.

- Networks that support authentication profiles. Using the RADIUS server in your network, you can configure AAA authentication and set up per-user profiles. Per-user profiles enable the Inspur INOS device to better manage ports using their existing RADIUS solutions and to efficiently manage shared resources to offer different service-level agreements.

3.2.2 RADIUS Operation

When a user attempts to log in and authenticate to a Inspur INOS device using RADIUS, the following process occurs:

- The user is prompted for and enters a username and password.
- The username and encrypted password are sent over the network to the RADIUS server.
- The user receives one of the following responses from the RADIUS server:

ACCEPT

The user is authenticated.

REJECT

The user is not authenticated and is prompted to reenter the username and password, or access is denied.

CHALLENGE

A challenge is issued by the RADIUS server. The challenge collects additional data from the user.

CHANGE PASSWORD

A request is issued by the RADIUS server, asking the user to select a new password.

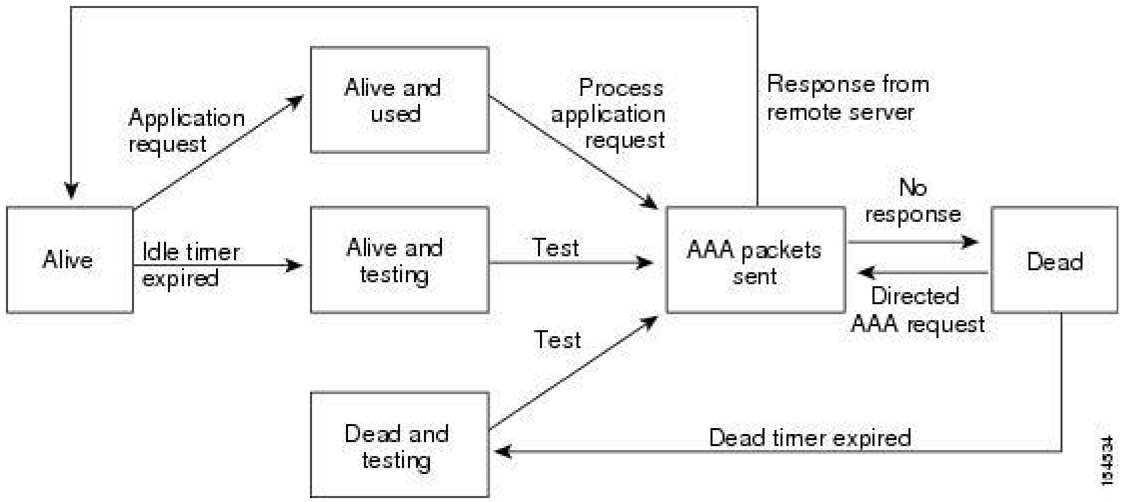
The ACCEPT or REJECT response is bundled with additional data that is used for EXEC or network authorization. You must first complete RADIUS authentication before using RADIUS authorization. The additional data included with the ACCEPT or REJECT packets consists of the following:

- Services that the user can access, including Telnet, rlogin, or local-area transport (LAT) connections, and Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services.
- Connection parameters, including the host or client IPv4 or IPv6 address, access list, and user timeouts.

3.2.3 RADIUS Server Monitoring

An unresponsive RADIUS server can cause a delay in processing AAA requests. You can configure the Inspur INOS device to periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. The Inspur INOS device marks unresponsive RADIUS servers as dead and does not send AAA requests to any dead RADIUS servers. The Inspur INOS device periodically monitors the dead RADIUS servers and brings them to the alive state once they respond. This monitoring process verifies that a RADIUS server is in a working state before real AAA requests are sent its way. Whenever a RADIUS server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Inspur INOS device displays an error message that a failure is taking place.

Figure 1 : RADIUS Server States



This figure shows the states for RADIUS server monitoring.

3.2.4 RADIUS Configuration Distribution

Inspur Fabric Services (CFS) allows the Inspur INOS device to distribute the RADIUS configuration to other Inspur INOS devices in the network. When you enable CFS distribution for a feature on your device, the device belongs to a CFS region containing other devices in the network that you have also enabled for CFS distribution for the feature. CFS distribution for RADIUS is disabled by default.

After you enable CFS distribution for RADIUS on your Inspur INOS device, the first RADIUS configuration command that you enter causes the Inspur INOS software to take the following actions:

- Creates a CFS session on your Inspur INOS device.
- Locks the RADIUS configuration on all Inspur INOS devices in the CFS region with CFS enabled for RADIUS.
- Saves the RADIUS configuration changes in a temporary buffer on the Inspur INOS device.

The changes stay in the temporary buffer on the Inspur INOS device until you explicitly commit them to be distributed to the devices in the CFS region. When you commit the changes, the Inspur INOS software takes the following actions:

- Applies the changes to the running configuration on your Inspur INOS device.
- Distributes the updated RADIUS configuration to the other Inspur INOS devices in the CFS region.
- Unlocks the RADIUS configuration in the devices in the CFS region.
- Terminates the CFS session.

CFS does not distribute the RADIUS server group configuration or server and global keys. The keys are unique to the Inspur INOS device and are not shared with other Inspur INOS devices.

3.2.5 Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating VSAs between the network access server and the RADIUS server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use. The Inspur RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Inspur vendor ID is 9, and the supported option is vendor type 1, which is named Inspur-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Inspur attribute for a particular type of authorization, the separator is = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use RADIUS servers for authentication on a Inspur INOS device, the RADIUS protocol directs the RADIUS server to return user attributes, such as authorization information, with authentication results. This authorization information is specified through VSAs.

The following VSA protocol options are supported by the Inspur INOS software:

Shell

Protocol used in access-accept packets to provide user profile information.

Accounting

Protocol used in accounting-request packets. If a value contains any white spaces, you should enclose the value within double quotation marks.

The Inspur INOS software supports the following attributes:

roles

Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space. For example, if the user belongs to roles network-operator and vdc-admin, the value field would be network-operator vdc-admin. This subattribute, which the RADIUS server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value. The following examples show the roles attribute that is supported by the Inspur Access Control Server (ACS):

```
shell:roles=network-operator vdc-admin

shell:roles*"network-operator vdc-admin
```

The following examples show the roles attribute that is supported by FreeRADIUS:

```
Inspur-AVPair = shell:roles=\network-operator vdc-

admin\ Inspur-AVPair = shell:roles*\network-operator

vdc-admin\
```

accountinginfo

Stores accounting information in addition to the attributes covered by a standard RADIUS accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the RADIUS client on the switch. It can be used only with the accounting protocol data units (PDUs).

3.3 Virtualization Support for RADIUS

RADIUS configuration and operation are local to the virtual device context (VDC). For more information on VDCs, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

The Inspur INOS device uses virtual routing and forwarding instances (VRFs) to access the RADIUS servers. For more information on VRFs, see the *Inspur CN12700 Series INOS Unicast Routing Configuration Guide*.

3.4 Licensing Requirements for RADIUS

This table shows the licensing requirements for this feature.

Product	License Requirement
Inspur INOS	RADIUS requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

3.5 Prerequisites for RADIUS

RADIUS has the following prerequisites:

- Obtain IPv4 or IPv6 addresses or hostnames for the RADIUS servers.
- Obtain keys from the RADIUS servers.
- Ensure that the Inspur INOS device is configured as a RADIUS client of the AAA servers.

3.6 Guidelines and Limitations for RADIUS

RADIUS has the following guidelines and limitations:

- You can configure a maximum of 64 RADIUS servers on the Inspur INOS device.
- If you have a user account configured on the local Inspur INOS device that has the same name as a remote user account on an AAA server, the Inspur INOS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.

3.7 Default Settings for RADIUS

This table lists the default settings for RADIUS parameters.

Table 6： Default RADIUS Parameter Settings

Parameters	Default
Server roles	Authentication and accounting
Dead timer interval	0 minutes
Retransmission count	1
Retransmission timer interval	5 seconds
Authentication port	1812
Accouinting port	1813
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

3.8 Configuring RADIUS Servers

This section describes how to configure RADIUS servers on a Inspur INOS device.

3.8.1 RADIUS Server Configuration Process

1. If needed, enable CFS configuration distribution for RADIUS.
2. Establish the RADIUS server connections to the Inspur INOS device.
3. Configure the RADIUS secret keys for the RADIUS servers.
4. If needed, configure RADIUS server groups with subsets of the RADIUS servers for AAA authentication methods.
5. If needed, configure any of the following optional parameters:

- Dead-time interval
- RADIUS server specification allowed at user login
- Timeout interval
- TCP port

6. (Optional) If RADIUS distribution is enabled, commit the RADIUS configuration to the fabric.

Related Topics

Configuring RADIUS Server Hosts.
Configuring Global RADIUS Keys.

3.8.2 Enabling RADIUS Configuration Distribution

Only Inspur INOS devices that have distribution enabled for RADIUS can participate in the distribution of the RADIUS configuration changes in the CFS region.

Before you begin

Ensure that CFS distribution is enabled.

SUMMARY STEPS

1. **configure terminal**
2. **radius distribute**
3. **exit**
4. (Optional) **show radius status**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius distribute** **Example:** `switch(config)# radius distribute`	Enable RADIUS configuration distribution. The default is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show radius status**	Displays the RADIUS CFS distribution configuration.

	Command or Action	Purpose
	Example: `switch(config)# show radius status`	
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

3.8.3 Configuring RADIUS Server Hosts

To access a remote RADIUS server, you must configure the IP address or hostname of a RADIUS server. You can configure up to 64 RADIUS servers.

Before you begin

Ensure that the server is already configured as a member of the server group. Ensure that the server is configured to authenticate RADIUS traffic.

Ensure that the Inspur INOS device is configured as a RADIUS client of the AAA servers.

SUMMARY STEPS

1. **configure terminal**
2. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*}
3. (Optional) **show radius** {**pending** | **pending-diff**}(Optional) radius commit
4. exit
5. (Optional) show radius-server
6. (Optional) copy running-config startup-config

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **Example:** `switch(config)# radius-server host 10.10.1.1`	Specifies the IPv4 or IPv6 address or hostname for a RADIUS server to use for authentication.
Step 3	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** `switch(config)# show radius pending`	Displays the RADIUS configuration pending for distribution.
Step 4	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration and distributes the RADIUS configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.
Step 5	**exit** **Example:**	Exits configuration mode.

	Command or Action	Purpose
	`switch(config)# exit` `switch#`	
Step 6	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring a Key for a Specific RADIUS Server.

3.8.4 Configuring Global RADIUS Keys

You can configure RADIUS keys for all servers used by the Inspur INOS device. A RADIUS key is a shared secret text string between the Inspur INOS device and the RADIUS server hosts.

Before you begin

Obtain the RADIUS key values for the remote RADIUS servers. Configure the RADIUS key on the remote RADIUS servers.

SUMMARY STEPS

1. **configure terminal**
2. **radius-server key** [**0** | **7**] *key-value*
3. **exit**
4. (Optional) **show radius-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server key** [**0** | **7**] *key-value* **Example:** `switch(config)# radius-server key 0 QsEfThUkO`	Specifies a RADIUS key for all RADIUS servers. You can specify that the *key-value* is in clear text format (**0**) or is type-7 encrypted (**7**). The Inspur INOS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters. By default, no RADIUS key is configured.
Step 3	**exit** **Example:** `switch(config)# exit`	Exits configuration mode.

	Command or Action	Purpose
	`switch#`	
Step 4	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration. **Note** The RADIUS keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted RADIUS keys.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Groups.
RADIUS Configuration Distribution.

3.8.5 Configuring a Key for a Specific RADIUS Server

You can configure a key on the Inspur INOS device for a specific RADIUS server. A RADIUS key is a secret text string shared between the Inspur INOS device and a specific RADIUS server.

Before you begin

Configure one or more RADIUS server hosts. Obtain the key value for the remote RADIUS server. Configure the key on the RADIUS server.

SUMMARY STEPS

1. **configure terminal**
2. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **7**] *key-value*
3. **exit**
4. (Optional) **show radius-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **7**] *key-value* **Example:** `switch(config)# radius-server host 10.10.1.1 key 0 PlIjUhYg`	Specifies a RADIUS key for a specific RADIUS server. You can specify that the *key-value* is in clear text format (**0**) or is type-7 encrypted (**7**). The Inspur INOS software encrypts a clear text key before saving it to the running configuration. The default format is clear text. The maximum length is 63 characters. This RADIUS key is used instead of the global RADIUS key.

	Command or Action	Purpose

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration. **Note** The RADIUS keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted RADIUS keys.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Hosts.

3.8.6 Configuring RADIUS ServerGroups

You can specify one or more remote AAA servers for authentication using server groups. All members of a group must belong to the RADIUS protocol. The servers are tried in the same order in which you configure them. You can configure up to 100 server groups in a VDC.

You can configure these server groups at any time but they only take effect when you apply them to an AAA service.

Before you begin

Ensure that all servers in the group are RADIUS servers.

SUMMARY STEPS

1. **configure terminal**
2. **aaa group server radius** *group-name*
3. **server** {*ipv4-address* | *ipv6-address* | *host-name*}
4. (Optional) **deadtime** *minutes*
5. (Optional) **server** {*ipv4-address* | *ipv6-address* | *host-name*}
6. (Optional) **use-vrf** *vrf-name*
7. **exit**
8. (Optional) **show radius-server groups** [*group-name*]
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa group server radius** *group-name*	Creates a RADIUS server group and enters the RADIUS server group configuration submode for

	Command or Action	Purpose
	Example: `switch(config)# aaa group server radius RadServer switch(config-radius)#`	that group. The *group-name* argument is a case-sensitive alphanumeric string with a maximum length of 127 characters.
Step 3	**server** {*ipv4-address* | *ipv6-address* | *host-name*} **Example:** `switch(config-radius)# server 10.10.1.1`	Configures the RADIUS server as a member of the RADIUS server group. If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.
Step 4	(Optional) **deadtime** *minutes* **Example:** `switch(config-radius)# deadtime 30`	Configures the monitoring dead time. The default is 0 minutes. The range is from 1 through 1440. **Note** If the dead-time interval for a RADIUS server group is greater than zero (0), that value takes precedence over the global dead-time value.
Step 5	(Optional) **server** {*ipv4-address* | *ipv6-address* | *host-name*} **Example:** `switch(config-radius)# server 10.10.1.1`	Configures the RADIUS server as a member of the RADIUS server group. **Tip** If the specified RADIUS server is not found, configure it using the **radius-server host** command and retry this command.
Step 6	(Optional) **use-vrf** *vrf-name* **Example:** `switch(config-radius)# use-vrf vrf1`	Specifies the VRF to use to contact the servers in the server group.
Step 7	**exit** **Example:** `switch(config-radius)# exit switch(config)#`	Exits configuration mode.
Step 8	(Optional) **show radius-server groups** [*group-name*] **Example:** `switch(config)# show radius-server groups`	Displays the RADIUS server group configuration.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring the RADIUS Dead-Time Interval.

3.8.7 Configuring the Global Source Interface for RADIUS Server Groups

You can configure a global source interface for RADIUS server groups to use when accessing RADIUS servers. You can also configure a different source interface for a specific RADIUS server group. By default, the Inspur INOS software uses any available interface.

SUMMARY STEPS

1. **configure terminal**
2. **ip radius source-interface** *interface*
3. **exit**
4. (Optional) **show radius-server**
5. (Optional) **copy running-config startup config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)`
		Enters global configuration mode.

Step 2	**ip radius source-interface** *interface* **Example:** `switch(config)# ip radius source-interface mgmt 0`	Configures the global source interface for all RADIUS server groups configured on the device.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration information.
Step 5	(Optional) **copy running-config startup config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Groups.

3.8.8 Allowing Users to Specify a RADIUS Server at Login

By default, the Inspur INOS device forwards an authentication request based on the default AAA authentication method. You can configure the Inspur INOS device to allow the user to specify a VRF and RADIUS server to send the authentication request by enabling the directed-request option. If you enable this option, the user can log in as *username@vrfname*:*hostname*, where *vrfname* is the VRF to use and **hostname** is the name of a configured RADIUS server.

SUMMARY STEPS

1. configure terminal
2. **radius-server directed-request**
3. (Optional) **show radius** {**pending** | **pending-diff**}
4. (Optional) **radius commit**
5. **exit**

6. (Optional) **show radius-server directed-request**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server directed-request** **Example:** `switch(config)# radius-server directed-request`	Allows users to specify a RADIUS server to send the authentication request when logging in. The default is disabled.
Step 3	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** `switch(config)# show radius pending`	Displays the RADIUS configuration pending for distribution.
Step 4	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration and distributes the RADIUS configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	(Optional) **show radius-server directed-request** **Example:** `switch# show radius-server directed-request`	Displays the directed request configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

RADIUS Configuration Distribution.

3.8.9 Configuring the Global RADIUS Transmission Retry Count and Timeout Interval

You can configure a global retransmission retry count and timeout interval for all RADIUS servers. By default, a Inspur INOS device retries transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. The timeout interval determines how long the Inspur INOS device waits for responses from RADIUS servers before declaring a timeout failure.

SUMMARY STEPS

1. **configure terminal**
2. **radius-server retransmit** *count*

3. **radius-server timeout** *seconds*
4. (Optional) **show radius** {**pending** | **pending-diff**}
5. (Optional) **radius commit**
6. **exit**
7. (Optional) **show radius-server**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server retransmit** *count* **Example:** `switch(config)# radius-server retransmit 3`	Specifies the retransmission count for all RADIUS servers. The default retransmission count is 1 and the range is from 0 to 5.
Step 3	**radius-server timeout** *seconds* **Example:** `switch(config)# radius-server timeout 10`	Specifies the transmission timeout interval for RADIUS servers. The default timeout interval is 5 seconds and the range is from 1 to 60 seconds.
Step 4	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** `switch(config)# show radius pending`	Displays the RADIUS configuration pending for distribution.
Step 5	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration and distributes the RADIUS configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 6	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 7	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

RADIUS Configuration Distribution.

3.8.10 Configuring the RADIUS Transmission Retry Count and Timeout Interval for a Server

By default, a Inspur INOS device retries a transmission to a RADIUS server only once before reverting to local authentication. You can increase this number up to a maximum of five retries per server. You can also set a timeout interval that the Inspur INOS device waits for responses from RADIUS servers before declaring a timeout failure.

Before you begin

Configure one or more RADIUS server hosts.

SUMMARY STEPS

1. **configure terminal**
2. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **retransmit** *count*
3. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds*
4. (Optional) **show radius** {**pending** | **pending-diff**}
5. (Optional) **radius commit**
6. **exit**
7. (Optional) **show radius-server**
8. (Optional) **copy running-config startup-config**

SUMMARY STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **retransmit** *count* **Example:** `switch(config)# radius-server host server1 retransmit 3`	Specifies the retransmission count for a specific server. The default is the global value. **Note** The retransmission count value specified for a RADIUS server overrides the count specified for all RADIUS servers.
Step 3	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds* **Example:** `switch(config)# radius-server host server1 timeout 10`	Specifies the transmission timeout interval for a specific server. The default is the global value. **Note** The timeout interval value specified for a RADIUS server overrides the interval value specified for all RADIUS servers.
Step 4	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** `switch(config)# show radius pending`	Displays the RADIUS configuration pending for distribution.
Step 5	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration and distributes RADIUS configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 6	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

	Command or Action	Purpose
Step 7	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Hosts.
RADIUS Configuration Distribution.

3.8.11 Configuring Accounting and Authentication Attributes for RADIUS Servers

You can specify that a RADIUS server is to be used only for accounting purposes or only for authentication purposes. By default, RADIUS servers are used for both accounting and authentication. You can also specify the destination UDP port numbers where RADIUS accounting and authentication messages should be sent if there is a conflict with the default port.

Before you begin

Configure one or more RADIUS server hosts.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **acct-port** *udp-port*
3. (Optional) **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **accounting**
4. (Optional) **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **auth-port** *udp-port*
5. (Optional) **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **authentication**
6. (Optional) **show radius** {**pending** | **pending-diff**}
7. (Optional) **radius commit**
8. **exit**
9. (Optional) **show radius-server**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **acct-port** *udp-port* **Example:** `switch(config)# radius-server host 10.10.1.1 acct-port 2004`	Specifies a UDP port to use for RADIUS accounting messages. The default UDP port is 1813. The range is from 0 to 65535.

	Command or Action	Purpose
Step 3	(Optional) **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **accounting** **Example:** `switch(config)# radius-server host 10.10.1.1 accounting`	Specifies to use the RADIUS server only for accounting purposes. The default is both accounting and authentication.
Step 4	(Optional) **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **auth-port** *udp-port* **Example:** `switch(config)# radius-server host 10.10.2.2 auth-port 2005`	Specifies a UDP port to use for RADIUS authentication messages. The default UDP port is 1812. The range is from 0 to 65535.
Step 5	(Optional) **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **authentication** **Example:** `switch(config)# radius-server host 10.10.2.2 authentication`	Specifies to use the RADIUS server only for authentication purposes. The default is both accounting and authentication.
Step 6	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** `switch(config)# show radius pending`	Displays the RADIUS configuration pending for distribution.
Step 7	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration and distributes the RADIUS configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 8	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 9	(Optional) **show radius-server** **Example:** `switch(config)# show radius-server`	Displays the RADIUS server configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Hosts.
RADIUS Configuration Distribution.

3.8.12 Configuring Periodic RADIUS Server Monitoring on Individual Servers

You can monitor the availability of individual RADIUS servers. The configuration parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval during which a RADIUS server receives no requests before the Inspur INOS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Before you begin

Enable RADIUS.
Add one or more RADIUS server hosts.

SUMMARY STEPS

1. **configure terminal**
2. **radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}
3. **radius-server deadtime** *minutes*
4. **exit**
5. (Optional) **show radius-server**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]} **Example:** `switch(config)# radius-server host 10.10.1.1 test username user1 password Ur2Gd2BH idle-time 3`	Specifies parameters for individual server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes. **Note** For periodic RADIUS server monitoring, you must set the idle timer to a value greater than 0.
Step 3	**radius-server deadtime** *minutes* **Example:** `switch(config)# radius-server deadtime 5`	Specifies the number of minutes before the Inspur INOS device checks a RADIUS server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 1 to 1440 minutes.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Hosts.

3.8.13 Configuring the RADIUS Dead-Time Interval

You can configure the dead-time interval for all RADIUS servers. The dead-time interval specifies the time that the Inspur INOS device waits after declaring a RADIUS server is dead, before sending out a test packet to determine if the server is now alive. The default value is 0 minutes.

SUMMARY STEPS

1. **configure terminal**
2. **radius-server deadtime** *minutes*
3. (Optional) **show radius** {**pending** | **pending-diff**}
4. (Optional) **radius commit**
5. **exit**
6. (Optional) **show radius-server**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server deadtime** *minutes* **Example:** `switch(config)# radius-server deadtime 5`	Configures the dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes.
Step 3	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** `switch(config)# show radius pending`	Displays the RADIUS configuration pending for distribution.
Step 4	(Optional) **radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration and distributes the RADIUS configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring RADIUS Server Groups.

RADIUS Configuration Distribution.

3.8.14 Committing the RADIUS Distribution

You can apply the RADIUS global and server-specific configuration stored in the temporary buffer to the running configuration across all devices in the fabric (including the originating device).

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **show radius** {**pending** | **pending-diff**}
3. **radius commit**
4. **exit**
5. (Optional) **show role session status**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** `switch(config)# show radius pending`	Displays the RADIUS configuration pending for distribution.
Step 3	**radius commit** **Example:** `switch(config)# radius commit`	Applies the RADIUS configuration changes in the temporary database to the running configuration and distributes the RADIUS configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show role session status** **Example:** `switch# show role session status`	Displays the user role CFS session status.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Applies the running configuration to the startup configuration.

3.8.15 Discarding the RADIUS Distribution Session

You can discard the temporary database of RADIUS changes and end the CFS distribution session.

SUMMARY STEPS

1. **configure terminal**

2. (Optional) **show radius** {**pending** | **pending-diff**}
3. **radius abort**
4. **exit**
5. (Optional) **show radius session status**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **show radius** {**pending** | **pending-diff**} **Example:** `switch(config)# show radius pending`	Displays the RADIUS configuration pending for distribution.
Step 3	**radius abort** **Example:** `switch(config)# radius abort`	Discards the RADIUS configuration in the temporary storage and ends the session.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show radius session status** **Example:** `switch# show radius session status`	Displays the RADIUS CFS session status.

3.8.16 Clearing the RADIUS Distribution Session

You can clear the ongoing Inspur Fabric Services distribution session (if any) and unlock the fabric for the RADIUS feature.

SUMMARY STEPS

1. **clear radius session**
2. (Optional) **show radius session status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear radius session** **Example:** `switch# clear radius session`	Clears the session and unlocks the fabric.
Step 2	(Optional) **show radius session status** **Example:** `switch# show radius session status`	Displays the RADIUS CFS session status.

3.8.17 Manually Monitoring RADIUS Servers or Groups

You can manually issue a test message to a RADIUS server or to a server group.

SUMMARY STEPS

1. **test aaa server radius** {*ipv4-address* | *ipv6-address* | *host-name*} [**vrf** *vrf-name*] *username password*
2. **test aaa group** *group-name username password*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**test aaa server radius** {*ipv4-address* | *ipv6-address* | *host-name*} [**vrf** *vrf-name*] *username password* **Example:** `switch# test aaa server radius 10.10.1.1 user1 Ur2Gd2BH`	Sends a test message to a RADIUS server to confirm availability.
Step 2	**test aaa group** *group-name username password* **Example:** `switch# test aaa group RadGroup user2 As3He3CI`	Sends a test message to a RADIUS server group to confirm availability.

3.9 Verifying the RADIUS Configuration

To display RADIUS configuration information, perform one of the following tasks:

Command	Purpose
show radius {**status** | **pending** | **pending-diff**}	Displays the RADIUS Inspur Fabric Services distribution status and other details.
show running-config radius [**all**]	Displays the RADIUS configuration in the running configuration.
show startup-config radius	Displays the RADIUS configuration in the startup configuration.
show radius-server [*host-name* | *ipv4-address* | *ipv6-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured RADIUS server parameters.

For detailed information about the fields in the output from this command, see the *Inspur CN12700 Series INOS Security Command Reference*.

3.10 Monitoring RADIUS Servers

You can monitor the statistics that the Inspur INOS device maintains for RADIUS server activity.

Before you begin

Configure one or more RADIUS server hosts.

SUMMARY STEPS

1. **show radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch# show radius-server statistics 10.10.1.1`	Displays the RADIUS statistics.

Related Topics

Configuring RADIUS Server Hosts.
Clearing RADIUS Server Statistics.

3.11 Clearing RADIUS Server Statistics

You can display the statistics that the Inspur INOS device maintains for RADIUS server activity.

Before you begin

Configure RADIUS servers on the Inspur INOS device.

SUMMARY STEPS

1. (Optional) **show radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}
2. **clear radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **show radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch# show radius-server statistics 10.10.1.1`	Displays the RADIUS server statistics on the Inspur INOS device.
Step 2	**clear radius-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch# clear radius-server statistics 10.10.1.1`	Clears the RADIUS server statistics.

Related Topics

Configuring RADIUS Server Hosts.

3.12 Configuration Example for RADIUS

The following example shows how to configure RADIUS:

```
radius-server key 7 "ToIkLhPpG"
radius-server host 10.10.1.1 key 7 "ShMoMhTl" authentication
accounting aaa group server radius RadServer
    server 10.10.1.1
```

3.13 Where to Go Next

You can now configure AAA authentication methods to include the server groups.

3.14 Additional References forRADIUS

This section describes additional information related to implementing RADIUS.

Related Documents

Related Topic	Document Title
Inspur INOS Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*
VRF configuration	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

3.15 Feature History for RADIUS

This table lists the release history for this feature.

Table 7： Feature History for RADIUS

Feature Name	Releases	Feature Information
RADIUS	8.2(3)	No change from Release 8.2(3)
RADIUS	8.2(3)	Added support for the Inspur CN 3000 Series Switches.
RADIUS	8.2(3)	Added type-6 encryption for RADIUS server keys.
RADIUS	8.2(3)	No change from Release 8.2(3)
RADIUS server groups	8.2(3)	Added support for configuring the global source interface for all RADIUS server groups.
RADIUS server groups	8.2(3)	Added support for configuring a source interface for a specific RADIUS server group.
Periodic server monitoring	8.2(3)	Added support for global periodic RADIUS server monitoring.
OTP	8.2(3)	Added support for one-time passwords.
RADIUS statistics	8.2(3)	Added support for clearing statistics for RADIUS server hosts.

Feature Name	Releases	Feature Information
RADIUS	8.2(3)	No change from Release8.2(3)

CHAPTER 4 Configuring TACACS+

This chapter describes how to configure the Terminal Access Controller Access Control System Plus (TACACS+) protocol on Inspur INOS devices.

This chapter includes the following sections:

- Finding Feature Information.
- Information About TACACS+.
- Licensing Requirements for TACACS+.
- Prerequisites for TACACS+.
- Guidelines and Limitations for TACACS+.
- Default Settings for TACACS+.
- Configuring TACACS+.
- Monitoring TACACS+ Servers.
- Clearing TACACS+ Server Statistics.
- Verifying the TACACS+ Configuration.
- Configuration Examples for TACACS+.
- Where to Go Next .
- Additional References for TACACS+.
- Feature History for TACACS+.

4.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

4.2 Information About TACACS+

The TACACS+ security protocol provides centralized validation of users attempting to gain access to a Inspur INOS device. TACACS+ services are maintained in a database on a TACACS+ daemon running, typically, on a UNIX or Windows NT workstation. You must have access to and must configure a TACACS+ server before the configured TACACS+ features on your Inspur INOS device are available.

TACACS+ provides for separate authentication, authorization, and accounting facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service—authentication, authorization, and accounting—independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The TACACS+ client/server protocol uses TCP (TCP port 49) for transport requirements. Inspur INOS devices provide centralized authentication using the TACACS+ protocol.

4.2.1 TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the Inspur INOS device can authorize access without authenticating.
- Uses the TCP transport protocol to send data between the AAA client and server, making reliable transfers with a connection-oriented protocol.
- Encrypts the entire protocol payload between the switch and the AAA server to ensure higher data

confidentiality. The RADIUS protocol only encrypts passwords.

4.2.2 TACACS+ Operation for User Login

When a user attempts a Password Authentication Protocol (PAP) login to a Inspur INOS device using TACACS+, the following actions occur:

1. When the Inspur INOS device establishes a connection, it contacts the TACACS+ daemon to obtain the username and password.
2. The Inspur INOS device will eventually receive one of the following responses from the TACACS+ daemon:

ACCEPT

User authentication succeeds and service begins. If the Inspur INOS device requires user authorization, authorization begins.

REJECT

User authentication failed. The TACACS+ daemon either denies further access to the user or prompts the user to retry the login sequence.

ERROR

An error occurred at some time during authentication either at the daemon or in the network connection between the daemon and the Inspur INOS device. If the Inspur INOS device receives an ERROR response, the Inspur INOS device tries to use an alternative method for authenticating the user.

After authentication, the user also undergoes an additional authorization phase if authorization has been enabled on the INOS device. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

3. If TACACS+ authorization is required, the Inspur INOS device again contacts the TACACS+ daemon and it returns an ACCEPT or REJECT authorization response. An ACCEPT response contains attributes that are used to direct the EXEC or NETWORK session for that user and determines the services that the user can access.

Services include the following:

- Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services
- Connection parameters, including the host or client IP address (IPv4 or IPv6), access list, and user timeouts

4.2.3 Default TACACS+ Server Encryption Type and Secret Key

- You must configure the TACACS+ secret key to authenticate the switch to the TACACS+ server. A secret key is a secret text string shared between the Inspur INOS device and the TACACS+ server host. The length of the key is restricted to 63 characters and can include any printable ASCII characters (white spaces are not allowed). You can configure a global secret key for all TACACS+ server configurations on the Inspur INOS device to use.

You can override the global secret key assignment by explicitly using the **key** option when configuring an individual TACACS+ server.

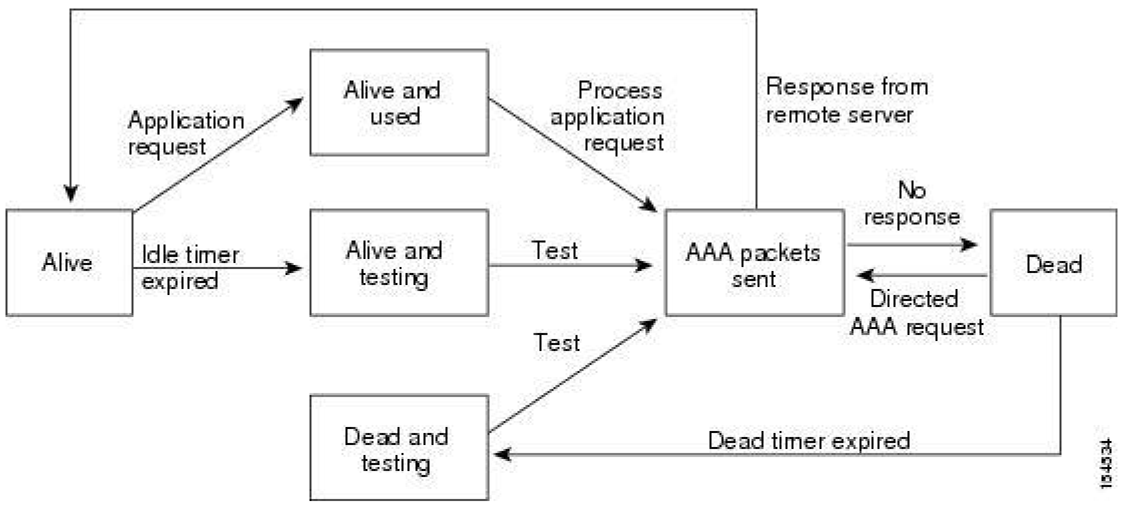
4.2.4 Command Authorization Support for TACACS+ Servers

By default, command authorization is done against a local database in the Inspur INOS software when an authenticated user enters a command at the command-line interface (CLI). You can also verify authorized commands for authenticated users using TACACS+.

4.2.5 TACACS+ Server Monitoring

An unresponsive TACACS+ server can delay the processing of AAA requests. A Inspur INOS device can periodically monitor a TACACS+ server to check whether it is responding (or alive) to save time in processing AAA requests. The Inspur INOS device marks unresponsive TACACS+ servers as dead and does not send AAA requests to any dead TACACS+ servers. A Inspur INOS device periodically monitors dead TACACS+ servers and brings them to the alive state once they are responding. This process verifies that a TACACS+ server is in a working state before real AAA requests are sent its way. Whenever a TACACS+ server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Inspur INOS device displays an error message that a failure is taking place before it can impact performance.

Figure 2： TACACS+ Server States



This figure shows the server states for TACACS+ server

4.2.6 TACACS+ Configuration Distribution

Inspur Fabric Services (CFS) allows the Inspur INOS device to distribute the TACACS+ configuration to other Inspur INOS devices in the network. When you enable CFS distribution for a feature on your device, the device belongs to a CFS region containing other devices in the network that you have also enabled for CFS distribution for the feature. CFS distribution for TACACS+ is disabled by default.

After you enable CFS distribution for TACACS+ on your Inspur INOS device, the first TACACS+ configuration command that you enter causes the Inspur INOS software to take the following actions:

- Creates a CFS session on your Inspur INOS device.
- Locks the TACACS+ configuration on all Inspur INOS devices in the CFS region with CFS enabled for TACACS+.
- Saves the TACACS+ configuration changes in a temporary buffer on the Inspur INOS device.

The changes stay in the temporary buffer on the Inspur INOS device until you explicitly commit them to be distributed to the devices in the CFS region. When you commit the changes, the Inspur INOS software takes the following actions:

- Applies the changes to the running configuration on your Inspur INOS device.
- Distributes the updated TACACS+ configuration to the other Inspur INOS devices in the CFS region.
- Unlocks the TACACS+ configuration in the devices in the CFS region.
- Terminates the CFS session.

CFS does not distribute the TACACS+ server group configuration, periodic TACACS+ server testing configurations, or server and global keys. The keys are unique to the Inspur INOS device and are not shared with other Inspur INOS devices.

4.2.7 Vendor-Specific Attributes forTACACS+

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific attributes (VSAs) between the network access server and the TACACS+ server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use.

Inspur VSA Format for TACACS+

The Inspur TACACS+ implementation supports one vendor-specific option using the format recommended in the IETF specification. The Inspur vendor ID is 9, and the supported option is vendor type 1, which is named Inspur-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Inspur attribute for a particular type of authorization, the separator is = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use TACACS+ servers for authentication on a Inspur INOS device, the TACACS+ protocol directs the TACACS+ server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

The following VSA protocol options are supported by the Inspur INOS software:

Shell

Protocol used in access-accept packets to provide user profile information.

Accounting

Protocol used in accounting-request packets. If a value contains any white spaces, you should enclose the value within double quotation marks.

The Inspur INOS software supports the following attributes:

roles

Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space. For example, if the user belongs to roles network-operator and vdc-admin, the value field would be network-operator vdc-admin. This subattribute, which the TACACS+ server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value. The following examples show the roles attribute as supported by Inspur ACS:

```
shell:roles=network-operator    vdc-admin

shell:roles*network-operator vdc-admin
```

accountinginfo

Stores accounting information in addition to the attributes covered by a standard TACACS+ accounting protocol. This attribute is sent only in the VSA portion of the Account-Request frames from the TACACS+ client on the switch. It can be used only with the accounting protocol data units (PDUs).

4.3 Licensing Requirements for TACACS+

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS	TACACS+ requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

4.4 Prerequisites for TACACS+

TACACS+ has the following prerequisites:

- Obtain the IPv4 or IPv6 addresses or hostnames for the TACACS+ servers.
- Obtain the secret keys from the TACACS+ servers, if any.
- Ensure that the Inspur INOS device is configured as a TACACS+ client of the AAA servers.

4.5 Guidelines and Limitations for TACACS+

TACACS+ has the following guidelines and limitations:

- You can configure a maximum of 64 TACACS+ servers on the Inspur INOS device.
- If you have a user account configured on the local Inspur INOS device that has the same name as a remote user account on an AAA server, the Inspur INOS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.
- Inspur recommends that you configure the dead-time interval if more than six servers are configured in a group. If you must configure more than six servers, make sure to set the dead-time interval to a value greater than 0 and enable dead server monitoring by configuring the test username and test password.
- For Inspur INOS Releases 8.2(3), command authorization on TACACS+ servers is available only for non-console sessions. If you use a console to login to the server, command authorization is disabled.

Beginning with Inspur INOS Release 8.2(3), command authorization on TACACS+ servers is available for both console and non-console sessions.

4.6 Default Settings for TACACS+

This table lists the default settings for TACACS+ parameters.

Table 8： Default TACACS+ Parameters Settings

Parameters	Default
TACACS+	Disabled
Dead timer interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

4.7 Configuring TACACS+

This section describes how to configure TACACS+ on a Inspur INOS device.

4.7.1 TACACS+ Server Configuration Process

Step 1	Enable TACACS+.
Step 2	If needed, enable CFS configuration distribution for TACACS+.
Step 3	Establish the TACACS+ server connections to the Inspur INOS device.
Step 4	Configure the secret keys for the TACACS+ servers.
Step 5	If needed, configure TACACS+ server groups with subsets of the TACACS+ servers for AAA authentication methods.
Step 6	(Optional) Configure the TCP port.
Step 7	(Optional) If needed, configure periodic TACACS+ server monitoring
Step 8	(Optional) If TACACS+ distribution is enabled, commit the TACACS+ configuration to the fabric.

Related Topics

Enabling TACACS+ .

4.7.2 Enabling TACACS+

By default, the TACACS+ feature is disabled on the Inspur INOS device. You must explicitly enable the TACACS+ feature to access the configuration and verification commands for authentication.

SUMMARY STEPS

1. **configure terminal**
2. **feature tacacs+**
3. **exit**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature tacacs+** **Example:** `switch(config)# feature tacacs+`	Enables TACACS+.

Step 3	**exit**
	Example:
	`switch(config)#`
	`exit switch#`
		Exits configuration mode.

	Command or Action	Purpose
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

4.7.3 Configuring TACACS+ Server Hosts

To access a remote TACACS+ server, you must configure the IP address or the hostname for the TACACS+ server on the Inspur INOS device. You can configure up to 64 TACACS+ servers.

Before you begin

Enable TACACS+.
Obtain the IPv4 or IPv6 addresses or the hostnames for the remote TACACS+ servers.

SUMMARY STEPS

1. **configure terminal**
2. **tacacs-server host** {*host-name* | *ipv4-address* | *ipv6-address*} [**key** [**0** | **6** | **7**] *shared-secret*] [**port** *port-number*] [**timeout** *seconds*] [**single-connection**]
3. (Optional) **show tacacs+** {**pending** | **pending-diff**}
4. (Optional) **tacacs+ commit**
5. **exit**
6. (Optional) **show tacacs-server**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**tacacs-server host** {*host-name* | *ipv4-address* | *ipv6-address*} [**key** [**0** | **6** | **7**] *shared-secret*][**port** *port-number*] [**timeout** *seconds*] [**single-connection**] **Example:** `switch(config)# tacacs-server host 10.10.2.2`	Specifies the IPv4 or IPv6 address or hostname for a TACACS+ server. Use the **single-connection** option to improve performance by configuring a single TACACS+ connection. Rather than have the device open and close a TCP connection to the daemon each time it must communicate, this option maintains a single open connection between the device and the daemon.
Step 3	(Optional) **show tacacs+** {**pending** | **pending-diff**} **Example:** `switch(config)# show tacacs+ pending`	Displays the TACACS+ configuration pending for distribution.
Step 4	(Optional) **tacacs+ commit** **Example:**	Applies the TACACS+ configuration changes in the temporary database to the running configuration and distributes TACACS+ configuration to other

	Command or Action	Purpose
	`switch(config)# tacacs+ commit`	INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	(Optional) **show tacacs-server** **Example:** `switch# show tacacs-server`	Displays the TACACS+ server configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+ .
Enabling TACACS+ Configuration Distribution.
Configuring TACACS+ Server Groups

4.7.4 Configuring Global TACACS+ Keys

You can configure secret TACACS+ keys at the global level for all servers used by the Inspur INOS device. A secret key is a shared secret text string between the Inspur INOS device and the TACACS+ server hosts.

Before you begin

Enable TACACS+.
Obtain the secret key values for the remote TACACS+ servers.

SUMMARY STEPS

1. **configure terminal**
2. **tacacs-server key** [**0** | **7**] *key-value*
3. **exit**
4. (Optional) **show tacacs-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**tacacs-server key** [**0** | **7**] *key-value* **Example:** `switch(config)# tacacs-server key 0 QsEfThUkO`	Specifies a TACACS+ key for all TACACS+ server. You can specify that the *key-value* is in clear text format (**0**) or is type-7 encrypted (**7**). The Inspur INOS software encrypts a clear text key before saving

	Command or Action	Purpose
		it to the running configuration. The default format is clear text. The maximum length is 63 characters. By default, no secret key is configured.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show tacacs-server** **Example:** `switch# show tacacs-server`	Displays the TACACS+ server configuration. **Note** The secret keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted secret keys.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+ .
Enabling TACACS+ Configuration Distribution.

4.7.5 Configuring a Key for a Specific TACACS+ Server

You can configure secret keys for a TACACS+ server. A secret key is a shared secret text string between the Inspur INOS device and the TACACS+ server host.

Before you begin

Enable TACACS+.
Obtain the secret key values for the remote TACACS+ servers.

SUMMARY STEPS

1. **configure terminal**
2. **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **6** | **7**] *key-value*
3. **exit**
4. (Optional) **show tacacs-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* |	Specifies a secret key for a specific TACACS+

	Command or Action	Purpose
	host-name} **key** [**0** | **6** | **7**] *key-value* **Example:** `switch(config)# tacacs-server host 10.10.1.1 key 0 PlIjUhYg`	server. You can specify the format of the secret key with the option **key**: • **key 0** specifies that the *key-value* entered is in clear text format • **key 6** specifies that the *key-value* entered is in type-6 encrypted format • **key 7** specifies that the *key-value* entered is in type-7 encrypted format If no key is specified, INOS software assumes the *key-value* to be clear text and encrypts it using type-7 encryption before saving it to running configuration. The maximum length of *key-value* is 63 characters This secret key is used instead of the global secret key. **Note** Type-6 encryption is done using AES cipher and a user-defined master key. Without this master key, type-6 keys are unusable. The master key is defined by the user and is never displayed in the configuration. Type-6 passwords are more secure. Type-7 encryption is done using a weak cipher and an encryption key that is hardwired into the OS. Type-7 passwords configured on one device can be decrypted on any other device because the encryption/decryption key is contained within the OS.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show tacacs-server** **Example:** `switch# show tacacs-server`	Displays the TACACS+ server configuration. **Note** The secret keys are saved in encrypted form in the running configuration. Use the **show running-config** command to display the encrypted secret keys.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring TACACS+ Server Groups

You can specify one or more remote AAA servers to authenticate users using server groups. All members of a group must belong to the TACACS+ protocol. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time but they only take effect when you apply them to an AAA service.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **configure terminal**
2. **tacacs-server host** {*host-name* | *ipv4-address* | *ipv6-address*} [**key** [**0** | **6** | **7**] *shared-secret*] [**port** *port-number*] [**timeout** *seconds*] [**single-connection**]
3. **aaa group server tacacs+** *group-name*
4. **server** {*ipv4-address* | *ipv6-address* | *host-name*}
5. **exit**
6. (Optional) **show tacacs-server groups**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**tacacs-server host** {*host-name* | *ipv4-address* | *ipv6-address*} [**key** [**0** | **6** | **7**] *shared-secret*] [**port** *port-number*] [**timeout** *seconds*] [**single-connection**]
	Example:
	`switch(config)# tacacs-server host 10.10.2.2`
	`switch(config-tacacs+)#`
		Specifies the IPv4 or IPv6 address or hostname for a TACACS+ server.
		Use the **single-connection** option to improve performance by configuring a single TACACS+ connection. Rather than have the device open and close a TCP connection to the daemon each time it must communicate, this option maintains a single open connection between the device and the daemon.

Step 3	**aaa group server tacacs+** *group-name*
	Example:
	`switch(config)# aaa group server tacacs+ TacServer`
	`switch(config-tacacs+)#`
		Creates a TACACS+ server group and enters the TACACS+ server group configuration mode for that group.

Step 4	**server** {*ipv4-address* | *ipv6-address* | *host-name*} **Example:** `switch(config-tacacs+)# server 10.10.2.2`	Configures the TACACS+ server as a member of the TACACS+ server group. If the specified TACACS+ server is not found, configure it using the **tacacs-server host** command and retry this command.

	Command or Action	Purpose

Step 5	**exit**
	Example:
	`switch(config-tacacs+)# exit`
	`switch(config)#`
		Exits TACACS+ server group configuration mode.

Step 6	(Optional) **show tacacs-server groups** **Example:** `switch(config)# show tacacs-server groups`	Displays the TACACS+ server group configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+ .
Remote AAA Services.
Configuring TACACS+ Server Hosts.
Configuring the TACACS+ Dead-Time Interval.

4.7.6 Configuring the Global Source Interface for TACACS+ Server Groups

You can configure a global source interface for TACACS+ server groups to use when accessing TACACS+ servers. You can also configure a different source interface for a specific TACACS+ server group. By default, the Inspur INOS software uses any available interface.

SUMMARY STEPS

1. **configure terminal**
2. **ip tacacs source-interface** *interface*
3. **exit**
4. (Optional) **show tacacs-server**
5. (Optional) **copy running-config startup config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)`
		Enters global configuration mode.

Step 2	**ip tacacs source-interface** *interface* **Example:** `switch(config)# ip tacacs source-interface mgmt 0`	Configures the global source interface for all TACACS+ server groups configured on the device.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

	Command or Action	Purpose
Step 4	(Optional) **show tacacs-server** **Example:** `switch# show tacacs-server`	Displays the TACACS+ server configuration information.
Step 5	(Optional) **copy running-config startup config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+ .
Configuring TACACS+ Server Groups.

4.7.7 Allowing Users to Specify a TACACS+ Server at Login

You can configure the switch to allow the user to specify which TACACS+ server to send the authentication request by enabling the directed-request option. By default, a Inspur INOS device forwards an authentication request based on the default AAA authentication method. If you enable this option, the user can log in as username@vrfname:hostname, where vrfname is the VRF to use and hostname is the name of a configured TACACS+ server.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **configure terminal**
2. **tacacs-server directed-request**
3. (Optional) **show tacacs+** {**pending** | **pending-diff**}
4. (Optional) **tacacs+ commit**
5. **exit**
6. (Optional) **show tacacs-server directed-request**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**tacacs-server directed-request** **Example:** `switch(config)# tacacs-server directed-request`	Allows users to specify a TACACS+ server to send the authentication request when logging in. The default is disabled.
Step 3	(Optional) **show tacacs+** {**pending** | **pending-diff**} **Example:** `switch(config)# show tacacs+ pending`	Displays the pending TACACS+ configuration.
Step 4	(Optional) **tacacs+ commit**	Applies the TACACS+ configuration changes in the

	Command or Action	Purpose
	Example: `switch(config)# tacacs+ commit`	temporary database to the running configuration and distributes TACACS+ configuration to other INOS devices if you have enabled CFS configuration distribution for the user role feature.
Step 5	**exit** **Example:** `switch(config)# exit switch#`	Exits configuration mode.
Step 6	(Optional) **show tacacs-server directed-request** **Example:** `switch# show tacacs-server directed-request`	Displays the TACACS+ directed request configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+ .
Enabling TACACS+ Configuration Distribution.

4.7.8 Configuring the Global TACACS+ Timeout Interval

You can set a global timeout interval that the device waits for responses from all TACACS+ servers before declaring a timeout failure. The timeout interval determines how long the device waits for responses from TACACS+ servers before declaring a timeout failure.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. From the Feature Selector pane, choose **Security > AAA > Server Groups**.
2. From the Summary pane, double-click the device to display the server groups.
3. Click **Default TACACS Server Group**.
4. From the Details pane, click the **Global Settings** tab.
5. In the Time out(secs) field, enter the number of seconds for the timeout interval.
6. From the menu bar, choose **File > Deploy** to apply your changes to the device.

DETAILED STEPS

Step 1 From the Feature Selector pane, choose **Security > AAA > Server Groups**

Step 2 From the Summary pane, double-click the device to display the server groups.

Step 3 Click **Default TACACS Server Group**.

Step 4 From the Details pane, click the **Global Settings** tab.

Step 5 In the Time out(secs) field, enter the number of seconds for the timeout interval.
The default is 5 seconds.

Step 6 From the menu bar, choose **File > Deploy** to apply your changes to the device.

Related Topics

Enabling TACACS+ .
Enabling TACACS+ Configuration Distribution.

4.7.9 Configuring the Timeout Interval for a TACACS+ Server

You can set a timeout interval that the Inspur INOS device waits for responses from a TACACS+ server before declaring a timeout failure. The timeout interval determines how long the Inspur INOS device waits for responses from a TACACS+ server before declaring a timeout failure.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **configure terminal**
2. **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds*
3. (Optional) **show tacacs+** {**pending** | **pending-diff**}
4. (Optional) **tacacs+ commit**
5. **exit**
6. (Optional) **show tacacs-server**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds* **Example:** `switch(config)# tacacs-server host server1 timeout 10`	Specifies the timeout interval for a specific server. The default is the global value. **Note** The timeout interval value specified for a TACACS+ server overrides the global timeout interval value specified for all TACACS+ servers.
Step 3	(Optional) **show tacacs+** {**pending** | **pending-diff**} **Example:** `switch(config)# show tacacs+ pending`	Displays the TACACS+ configuration pending for distribution.
Step 4	(Optional) **tacacs+ commit** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration and distributes the TACACS+ configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

	Command or Action	Purpose

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	(Optional) **show tacacs-server** **Example:** `switch# show tacacs-server`	Displays the TACACS+ server configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+ .
Enabling TACACS+ Configuration Distribution.

4.7.10 Configuring TCP Ports

You can configure another TCP port for the TACACS+ servers if there are conflicts with another application. By default, Inspur INOS devices use port 49 for all TACACS+ requests.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **configure terminal**
2. **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **port** *tcp-port*
3. (Optional) **show tacacs+** {**pending** | **pending-diff**}
4. (Optional) **tacacs+ commit**
5. **exit**
6. (Optional) **show tacacs-server**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **port** *tcp-port* **Example:** `switch(config)# tacacs-server host 10.10.1.1 port 2`	Specifies the TCP port to use for TACACS+ messages to the server. The default TCP port is 49. The range is from 1 to 65535.

	Command or Action	Purpose
Step 3	(Optional) **show tacacs+** {**pending** | **pending-diff**} **Example:** `switch(config)# show tacacs+ distribution pending`	Displays the TACACS+ configuration pending for distribution.
Step 4	(Optional) **tacacs+ commit** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration and distributes TACACS+ configuration to other INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	(Optional) **show tacacs-server** **Example:** `switch# show tacacs-server`	Displays the TACACS+ server configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+ .
Enabling TACACS+ Configuration Distribution.

4.7.11 Configuring Periodic TACACS+ Server Monitoring on Individual Servers

You can monitor the availability of individual TACACS+ servers. The configuration parameters include the username and password to use for the server and an idle timer. The idle timer specifies the interval in which a TACACS+ server receives no requests before the Inspur INOS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Before you begin

Enable TACACS+.
Add one or more TACACS+ server hosts.

SUMMARY STEPS

1. **configure terminal**
2. **tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]}
3. **tacacs-server dead-time** *minutes*
4. **exit**
5. (Optional) **show tacacs-server**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]} **Example:** `switch(config)# tacacs-server host 10.10.1.1 test username user1 password Ur2Gd2BH idle-time 3`	Specifies parameters for individual server monitoring. The default username is test, and the default password is test. The default value for the idle timer is 0 minutes, and the valid range is from 0 to 1440 minutes. **Note** For periodic TACACS+ server monitoring, the idle timer value must be greater than 0.
Step 3	**tacacs-server dead-time** *minutes* **Example:** `switch(config)# tacacs-server dead-time 5`	Specifies the number of minutes before the Inspur INOS device checks a TACACS+ server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 0 to 1440 minutes.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show tacacs-server** **Example:** `switch# show tacacs-server`	Displays the TACACS+ server configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring TACACS+ Server Hosts.
Enabling TACACS+ Configuration Distribution.

4.7.12 Configuring the TACACS+ Dead-Time Interval

You can configure the dead-time interval for all TACACS+ servers. The dead-time interval specifies the time that the Inspur INOS device waits, after declaring a TACACS+ server is dead, before sending out a test packet to determine if the server is now alive.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **configure terminal**

2. **tacacs-server deadtime** *minutes*
3. (Optional) **show tacacs+** {**pending** | **pending-diff**}
4. (Optional) **tacacs+ commit**
5. **exit**
6. (Optional) **show tacacs-server**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**tacacs-server deadtime** *minutes* **Example:** `switch(config)# tacacs-server deadtime 5`	Configures the global dead-time interval. The default value is 0 minutes. The range is from 1 to 1440 minutes.
Step 3	(Optional) **show tacacs+** {**pending** | **pending-diff**} **Example:** `switch(config)# show tacacs+ pending`	Displays the pending TACACS+ configuration.
Step 4	(Optional) **tacacs+ commit** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration and distributes TACACS+ configuration to other INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	(Optional) **show tacacs-server** **Example:** `switch# show tacacs-server`	Displays the TACACS+ server configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+ Configuration Distribution.

4.7.13 Configuring ASCII Authentication

You can enable ASCII authentication on the TACACS+ server.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authentication login ascii-authentication**
3. (Optional) **show tacacs+** {**pending** | **pending-diff**}
4. (Optional) **tacacs+ commit**
5. **exit**
6. (Optional) **show tacacs-server**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authentication login ascii-authentication** **Example:** `switch(config)# aaa authentication login ascii-authentication`	Enables ASCII authentication. The default is disabled.
Step 3	(Optional) **show tacacs+** {**pending** | **pending-diff**} **Example:** `switch(config)# show tacacs+ pending`	Displays the pending TACACS+ configuration.
Step 4	(Optional) **tacacs+ commit** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration and distributes TACACS+ configuration to the other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 6	(Optional) **show tacacs-server** **Example:** `switch# show tacacs-server`	Displays the TACACS+ server configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

4.7.14 Configuring Command Authorization on TACACS+ Servers

You can configure authorization for commands on TACACS+ servers.

Caution	Command authorization disables user role-based authorization control (RBAC), including the default

	roles.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authorization** {**commands** | **config-commands**} {**console** | **default**}
3. (Optional) **show tacacs+** {**pending** | **pending-diff**}
4. (Optional) **tacacs+ commit**
5. **exit**
6. (Optional) **show aaa authorization** [**all**]
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authorization** {**commands** | **config-commands**}
	{**console** | **default**}
	Example:
	`switch(config)# aaa authorization`
	`commands default group TacGroup`
	`Per command authorization will disable RBAC for all`
	`users. Proceed (y/n)?`
		Configures the command authorization method for specific roles on a TACACS+ server.
		The **commands** keyword configures authorization sources for all EXEC commands, and the **config-commands** keyword configures authorization sources for all configuration commands.
		The **console** keyword configures command authorization for a console session, and the **default** keyword configures command authorization for a non-console session.
		The *group-list* argument consists of a space-delimited list of TACACS+ server group names. Servers belonging to this group are contacted for command authorization. The **local** method uses the local role-based database for authorization.
		The **local** method is used only if all the configured server groups fail to respond and you have configured **local** as the fallback method. The default method is **local**.
		If you have not configured a fallback method after the TACACS+ server group method, authorization fails if all server groups fail to respond.
		If you press **Enter** at the confirmation prompt, the default action is **n**.

	Command or Action	Purpose
Step 3	(Optional) **show tacacs+** {**pending** | **pending-diff**} **Example:** `switch(config)# show tacacs+ pending`	Displays the pending TACACS+ configuration.
Step 4	(Optional) **tacacs+ commit** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration and distributes TACACS+ configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 6	(Optional) **show aaa authorization** [**all**] **Example:** `switch(config)# show aaa authorization`	Displays the AAA authorization configuration. The **all** keyword displays the default values.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+ , on page 78
Testing Command Authorization on TACACS+ Servers.

4.7.15 Testing Command Authorization on TACACS+ Servers

You can test the command authorization for a user on the TACACS+ servers.

Before you begin

Enable TACACS+.
Ensure that you have configured command authorization for the TACACS+ servers.

SUMMARY STEPS

1. **test aaa authorization command-type** {**commands** | **config-commands**} **user** *username* **command** *command-string*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**test aaa authorization command-type** {**commands** | **config-commands**} **user** *username* **command** *command-string* **Example:** `switch# test aaa authorization command-type commands user TestUser command reload`	Tests a user's authorization for a command on the TACACS+ servers. The **commands** keyword specifies only EXEC commands and the **config-commands** keyword specifies only configuration commands. **Note** Put double quotes (") before and after the

	Command or Action	Purpose
		command-string argument if it contains spaces.

Related Topics

Enabling TACACS+ .
Configuring Command Authorization on TACACS+ Servers.
Configuring User Accounts and RBAC.

4.7.16 Enabling and Disabling Command Authorization Verification

You can enable and disable command authorization verificaiton on the command-line interface (CLI) for the default user session or for another username.

SUMMARY STEPS

1. **terminal verify-only** [**username** *username*]
2. **terminal no verify-only** [**username** *username*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**terminal verify-only** [**username** *username*] **Example:** `switch# terminal verify-only`	Enables command authorization verification. After you enter this command, the Inspur INOS software indicates whether the commands you enter are authorized or not.
Step 2	**terminal no verify-only** [**username** *username*] **Example:** `switch# terminal no verify-only`	Disables command authorization verification.

4.7.17 Enabling TACACS+ Configuration Distribution

Only Inspur INOS devices that have distribution enabled can participate in the distribution of the TACACS+ configuration changes in the CFS region.

Before you begin

Ensure that CFS distribution is enabled.

SUMMARY STEPS

1. **configure terminal**
2. **tacacs+ distribute**
3. **exit**
4. (Optional) **show tacacs+ status**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose

	Example:
	`switch# configure terminal`
	`switch(config)#`	

Step 2	**tacacs+ distribute** **Example:** `switch(config)# tacacs+ distribute`	Enables TACACS+ configuration distribution. The default is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show tacacs+ status** **Example:** `switch(config)# show tacacs+ status`	Displays the TACACS+ CFS distribution configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+ .
Configuring TACACS+ Server Hosts.
TACACS+ Server Configuration Process.
Configuring TACACS+ Server Groups.

4.7.18 Committing the TACACS+ Configuration to Distribution

You can apply the TACACS+ global and server configuration stored in the temporary buffer to the running configuration across all Inspur INOS devices in the fabric (including the originating device).

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **show tacacs+** {**pending** | **pending-diff**}
3. **tacacs+ commit**
4. **exit**
5. (Optional) **show tacacs+ distribution status**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
	`switch(config)#`	
Step 2	(Optional) **show tacacs+** {**pending** | **pending-diff**} **Example:** `switch(config)# show tacacs+ pending`	Displays the TACACS+ configuration pending for distribution.
Step 3	**tacacs+ commit** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration and distributes the TACACS+ configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show tacacs+ distribution status** **Example:** `switch(config)# show tacacs+ distribution status`	Displays the TACACS distribution configuration and status.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Applies the running configuration to the startup configuration.

Related Topics

Enabling TACACS+ Configuration Distribution.

4.7.19 Discarding the TACACS+ Distribution Session

You can discard the temporary database of TACACS+ changes and end the CFS distribution session.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **show tacacs+** {**pending** | **pending-diff**}
3. **tacacs+ abort**
4. **exit**
5. (Optional) **show tacacs+ distribution status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

	`switch(config)#`	
Step 2	(Optional) **show tacacs+** {**pending** | **pending-diff**} **Example:** `switch(config)# show tacacs+ pending`	Displays the TACACS+ configuration pending for distribution.
Step 3	**tacacs+ abort** **Example:** `switch(config)# tacacs+ abort`	Discards the TACACS+ configuration in the temporary storage and ends the session.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show tacacs+ distribution status** **Example:** `switch(config)# show tacacs+ distribution status`	Displays the TACACS distribution configuration and status.

Related Topics

Enabling TACACS+ Configuration Distribution.

4.7.20 Clearing the TACACS+ Distribution Session

You can clear an active CFS distribution session and unlock TACACS+ configuration in the network.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **clear tacacs+ session**
2. (Optional) **show tacacs+ distribution status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear tacacs+ session** **Example:** `switch# clear tacacs+ session`	Clears the CFS session for TACACS+ and unlocks the fabric.
Step 2	(Optional) **show tacacs+ distribution status** **Example:** `switch(config)# show tacacs+ distribution status`	Displays the TACACS distribution configuration and status.

Related Topics

Enabling TACACS+ Configuration Distribution.

4.7.21 Manually Monitoring TACACS+ Servers or Groups

You can manually issue a test message to a TACACS+ server or to a server group.

Before you begin

Enable TACACS+.

SUMMARY STEPS

1. **test aaa server tacacs+** {*ipv4-address* | *ipv6-address* | *host-name*} [**vrf** *vrf-name*] *username password*
2. **test aaa group** *group-name username password*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**test aaa server tacacs+** {*ipv4-address* | *ipv6-address* | *host-name*} [**vrf** *vrf-name*] *username password* **Example:** `switch# test aaa server tacacs+ 10.10.1.1 user1 Ur2Gd2BH`	Sends a test message to a TACACS+ server to confirm availability.
Step 2	**test aaa group** *group-name username password* **Example:** `switch# test aaa group TacGroup user2 As3He3CI`	Sends a test message to a TACACS+ server group to confirm availability.

Related Topics

Configuring TACACS+ Server Hosts.
Configuring TACACS+ Server Groups.

4.7.22 Disabling TACACS+

You can disable TACACS+.

Caution	When you disable TACACS+, all related configurations are automatically discarded.

SUMMARY STEPS

1. **configure terminal**
2. **no feature tacacs+**
3. **exit**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no feature tacacs+** **Example:** `switch(config)# no feature tacacs+`	Disables TACACS+.
Step 3	**exit**	Exits configuration mode.

	Command or Action	Purpose

	Example:
	`switch(config)# exit`
	`switch#`	

Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

4.8 Monitoring TACACS+ Servers

You can monitor the statistics that the Inspur INOS device maintains for TACACS+ server activity.

Before you begin

Configure TACACS+ servers on the Inspur INOS device.

SUMMARY STEPS

1. **show tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch# show tacacs-server statistics 10.10.1.1`	Displays the TACACS+ statistics.

Related Topics

Configuring TACACS+ Server Hosts.
Clearing TACACS+ Server Statistics.

4.9 Clearing TACACS+ Server Statistics

You can display the statistics that the Inspur INOS device maintains for TACACS+ server activity.

Before you begin

Configure TACACS+ servers on the Inspur INOS device.

SUMMARY STEPS

1. (Optional) **show tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}
2. **clear tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **show tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}	Displays the TACACS+ server statistics on the Inspur INOS device.

	Command or Action	Purpose
	Example: `switch# show tacacs-server statistics 10.10.1.1`	
Step 2	**clear tacacs-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch# clear tacacs-server statistics 10.10.1.1`	Clears the TACACS+ server statistics.

Related Topics

Configuring TACACS+ Server Hosts.

4.10 Verifying the TACACS+ Configuration

To display the TACACS+ configuration, perform one of the following tasks:

Command	Purpose
show tacacs+ { **status** | **pending** | **pending-diff**}	Displays the TACACS+ Inspur Fabric Services distribution status and other details.
show running-config tacacs+ [**all**]	Displays the TACACS+ configuration in the running configuration.
show startup-config tacacs	Displays the TACACS+ configuration in the startup configuration.
show tacacs-server [*host-name* | *ipv4-address* | *ipv6-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured TACACS+ server parameters.

4.11 Configuration Examples for TACACS+

The following example shows how to configure a TACACS+ server host and server group:

```
feature tacacs+
tacacs-server key 7 "ToIkLhPpG"
tacacs-server  host  10.10.2.2  key  7  "ShMoMhTl"
aaa group server tacacs+ TacServer
     server 10.10.2.2
```

The following example shows how to configure and use command authorization verification:

```
switch# terminal verify-only
switch# show interface ethernet 7/2 brief
%Success
switch# terminal no verify-only
switch# show interface ethernet 7/2 brief

--------------------------------------------------------------------------------
Ethernet      VLAN   Type Mode   Status  Reason                   Speed     Port
Interface                                                                   Ch #
--------------------------------------------------------------------------------
Eth7/2        1      eth  access down    SFP not inserted           auto(D) --
```

4.12 Where to Go Next

You can now configure AAA authentication methods to include the server groups.

4.13 Additional References for TACACS+

This section includes additional information related to implementing TACACS+.

Related Documents

Related Topic	Document Title
Inspur INOS licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*
VRF configuration	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

4.14 Feature History for TACACS+

This table lists the release history for this feature.

Table 9：Feature History for TACACS+

Feature Name	Releases	Feature Information	
TACACS+	8.2(3)	Added support for a single TACACS+ connection.	
TACACS+	8.2(3)	Added the ability to configure command authorization for a console session.	

CHAPTER 5 Configuring LDAP

This chapter describes how to configure the Lightweight Directory Access Protocol (LDAP) on Inspur INOS devices.

This chapter includes the following sections:
- Finding Feature Information.
- Information About LDAP.
- Licensing Requirements for LDAP.
- Prerequisites for LDAP.
- Guidelines and Limitations for LDAP.
- Default Settings for LDAP.
- Configuring LDAP.
- Monitoring LDAP Servers.
- Clearing LDAP Server Statistics.
- Verifying the LDAP Configuration.
- Configuration Examples for LDAP.
- Where to Go Next.
- Additional References for LDAP.
- Feature History for LDAP.

5.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

5.2 Information About LDAP

The Lightweight Directory Access Protocol (LDAP) provides centralized validation of users attempting to gain access to a Inspur INOS device. LDAP services are maintained in a database on an LDAP daemon running, typically, on a UNIX or Windows NT workstation. You must have access to and must configure an LDAP server before the configured LDAP features on your Inspur INOS device are available.

LDAP provides for separate authentication and authorization facilities. LDAP allows for a single access control server (the LDAP daemon) to provide each service—authentication and authorization—independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The LDAP client/server protocol uses TCP (TCP port 389) for transport requirements. Inspur INOS devices provide centralized authentication using the LDAP protocol.

5.2.1 LDAP Authentication and Authorization

Clients establish a TCP connection and authentication session with an LDAP server through a simple bind (username and password). As part of the authorization process, the LDAP server searches its database to retrieve the user profile and other information.

You can configure the bind operation to first bind and then search, where authentication is performed first and authorization next, or to first search and then bind. The default method is to first search and then bind.

The advantage of searching first and binding later is that the distinguished name (DN) received in the search result can be used as the user DN during binding rather than forming a DN by prepending the username (cn attribute) with the baseDN. This method is especially helpful when the user DN is different from the username plus the baseDN. For the user bind, the bindDN is constructed as baseDN + append-with-baseDN, where append-with-baseDN has a default value of cn=$userid.

5.2.2 LDAP Operation for User Login

When a user attempts a Password Authentication Protocol (PAP) login to a Inspur INOS device using LDAP, the following actions occur:

1. When the Inspur INOS device establishes a connection, it contacts the LDAP daemon to obtain the username and password.

2. The Inspur INOS device eventually receives one of the following responses from the LDAP daemon:

ACCEPT

User authentication succeeds and service begins. If the Inspur INOS

device requires user authorization, authorization begins. REJECT

User authentication fails. The LDAP daemon either denies further access to the user or prompts the user to retry the login sequence.

ERROR

An error occurs at some time during authentication either at the daemon or in the network connection between the daemon and the Inspur INOS device. If the Inspur INOS device receives an ERROR response, the Inspur INOS device tries to use an alternative method for authenticating the user.

After authentication, the user also undergoes an additional authorization phase if authorization has been enabled on the Inspur INOS device. Users must first successfully complete LDAP authentication before proceeding to LDAP authorization.

3. If LDAP authorization is required, the Inspur INOS device again contacts the LDAP daemon and it returns an ACCEPT or REJECT authorization response. An ACCEPT response contains attributes that are used to direct the EXEC or NETWORK session for that user and determines the services that the user can access.

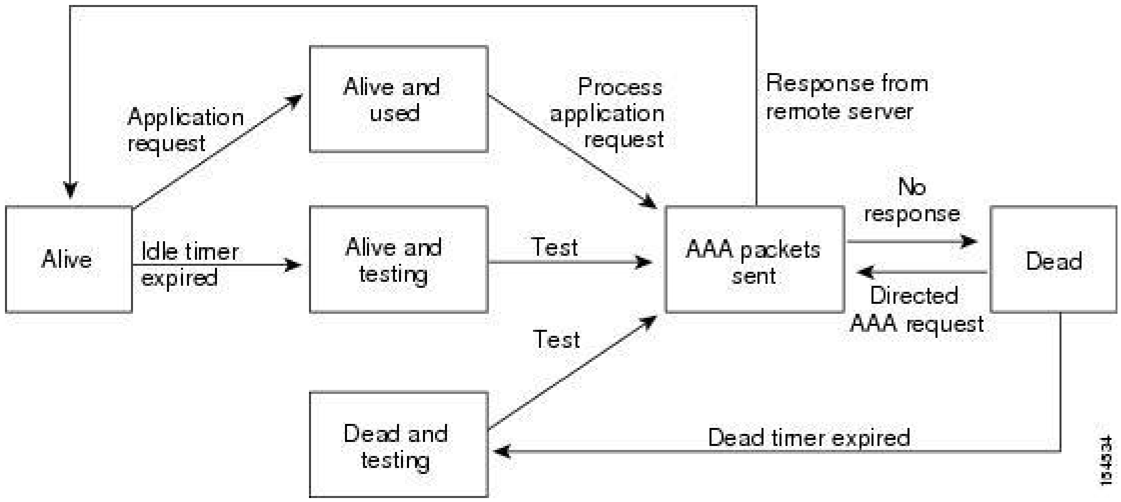
Services include the following:

- Telnet, rlogin, Point-to-Point Protocol (PPP), Serial Line Internet Protocol (SLIP), or EXEC services
- Connection parameters, including the host or client IP address (IPv4 or IPv6), access list, and user timeouts

5.2.3 LDAP Server Monitoring

An unresponsive LDAP server can delay the processing of AAA requests. A Inspur INOS device can periodically monitor an LDAP server to check whether it is responding (or alive) to save time in processing AAA requests. The Inspur INOS device marks unresponsive LDAP servers as dead and does not send AAA requests to any dead LDAP servers. A Inspur INOS device periodically monitors dead LDAP servers and brings them to the alive state once they are responding. This process verifies that an LDAP server is in a working state before real AAA requests are sent its way. Whenever an LDAP server changes to the dead or alive state, a Simple Network Management Protocol (SNMP) trap is generated and the Inspur INOS device displays an error message that a failure is taking place before it can impact performance.

Table 10： LDAP Server States



This figure shows the server states for LDAP server

5.2.4 Vendor-Specific Attributes for LDAP

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific attributes (VSAs) between the network access server and the LDAP server. The IETF uses attribute 26. VSAs allow vendors to support their own extended attributes that are not suitable for general use.

Inspur VSA Format for LDAP

The Inspur LDAP implementation supports one vendor-specific option using the format recommended in the IETF specification. The Inspur vendor ID is 9, and the supported option is vendor type 1, which is named Inspur-av-pair. The value is a string with the following format:

```
protocol : attribute separator value *
```

The protocol is a Inspur attribute for a particular type of authorization, the separator is = (equal sign) for mandatory attributes, and * (asterisk) indicates optional attributes.

When you use LDAP servers for authentication on a Inspur INOS device, LDAP directs the LDAP server to return user attributes, such as authorization information, along with authentication results. This authorization information is specified through VSAs.

The following VSA protocol options are supported by the Inspur INOS software:

Shell

Protocol used in access-accept packets to provide user profile information.

The Inspur INOS software supports the following attributes:

roles

Lists all the roles to which the user belongs. The value field is a string that lists the role names delimited by white space. For example, if the user belongs to roles network-operator and vdc-admin, the value field would be network-operator vdc-admin. This subattribute, which the LDAP server sends in the VSA portion of the Access-Accept frames, can only be used with the shell protocol value. The following examples show the roles attribute as supported by Inspur ACS:

```
shell:roles=network-operator vdc-admin

shell:roles*network-operator vdc-admin
```

5.2.5 Virtualization Support for LDAP

LDAP configuration and operation are local to the virtual device context (VDC). For more information on VDCs, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

The Inspur INOS device uses virtual routing and forwarding instances (VRFs) to access the LDAP servers. For more information on VRFs, see the *Inspur CN12700 Series INOS Unicast Routing Configuration Guide*.

5.3 Licensing Requirements for LDAP

The following table shows the licensing requirements for this feature:

Product	**License Requirement**
Inspur INOS	LDAP requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

5.4 Prerequisites for LDAP

LDAP has the following prerequisites:

- Obtain the IPv4 or IPv6 addresses or hostnames for the LDAP servers.
- Ensure that the Inspur INOS device is configured as an LDAP client of the AAA servers.

5.5 Guidelines and Limitations for LDAP

LDAP has the following guidelines and limitations:

- You can configure a maximum of 64 LDAP servers on the Inspur INOS device.
- Inspur INOS supports only LDAP version 3.
- Inspur INOS supports only these LDAP servers:
- OpenLDAP
- Microsoft Active Directory
- LDAP over Secure Sockets Layer (SSL) supports only SSL version 3 and Transport Layer Security (TLS) version 1.
- If you have a user account configured on the local Inspur INOS device that has the same name as a remote user account on an AAA server, the Inspur INOS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.

5.6 Default Settings for LDAP

This table lists the default settings for LDAP parameters.

Table 11 : Default LDAP Parameters Settings

Parameters	**Default**
LDAP	Disabled

Parameters	**Default**
LDAP authentication method	First search and then bind
LDAP authentication mechanism	Plain
Dead-time interval	0 minutes
Timeout interval	5 seconds
Idle timer interval	60 minutes
Periodic server monitoring username	test
Periodic server monitoring password	Inspur

5.7 Configuring LDAP

This section describes how to configure LDAP on a Inspur INOS device.

5.7.1 LDAP Server Configuration Process

You can configure LDAP servers by following this configuration process.

Step 1 Enable LDAP.

Step 2 Establish the LDAP server connections to the Inspur INOS device.

Step 3 If needed, configure LDAP server groups with subsets of the LDAP servers for AAA authentication methods.

Step 4 (Optional) Configure the TCP port.

Step 5 (Optional) Configure the default AAA authorization method for the LDAP server.

Step 6 (Optional) Configure an LDAP search map.

Step 7 (Optional) If needed, configure periodic LDAP server monitoring.

Related Topics

Enabling LDAP.
Configuring LDAP Server Hosts.
Configuring the RootDN for an LDAP Server.
Configuring LDAP Server Groups.
Configuring TCP Ports.
Configuring LDAP Search Maps.
Configuring Periodic LDAP Server Monitoring.

5.7.2 Enabling LDAP

By default, the LDAP feature is disabled on the Inspur INOS device. You must explicitly enable the LDAP feature to access the configuration and verification commands for authentication.

SUMMARY STEPS

1. **configure terminal**
2. **feature ldap**
3. **exit**

4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature ldap** **Example:** `switch(config)# feature ldap`	Enables LDAP.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

5.7.3 Configuring LDAP Server Hosts

To access a remote LDAP server, you must configure the IP address or the hostname for the LDAP server on the Inspur INOS device. You can configure up to 64 LDAP servers.

Before you begin

Enable LDAP.

Obtain the IPv4 or IPv6 addresses or the hostnames for the remote LDAP servers.

If you plan to enable the Secure Sockets Layer (SSL) protocol, make sure that the LDAP server certificate is manually configured on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} [**enable-ssl**]
3. **exit**
4. (Optional) **show ldap-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} [**enable-ssl**]	Specifies the IPv4 or IPv6 address or hostname for an LDAP server.

	Command or Action	Purpose
	Example: `switch(config)# ldap-server host 10.10.2.2 enable-ssl`	The **enable-ssl** keyword ensures the integrity and confidentiality of the transferred data by causing the LDAP client to establish a Secure Sockets Layer (SSL) session prior to sending the bind or search request.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show ldap-server** **Example:** `switch# show ldap-server`	Displays the LDAP server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling LDAP.
Configuring LDAP Server Groups.

5.7.4 Configuring the RootDN for an LDAP Server

You can configure the root designated name (DN) for the LDAP server database. The rootDN is used to bind to the LDAP server to verify its state.

Before you begin

Enable LDAP.
Obtain the IPv4 or IPv6 addresses or the hostnames for the remote LDAP servers.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **rootDN** *root-name* [**password** *password*] [**port** *tcp-port* [**timeout** *seconds*] | [**timeout** *seconds*]]
3. **exit**
4. (Optional) **show ldap-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address*	Specifies the rootDN for the LDAP server database

	Command or Action	Purpose
	| *host-name*} **rootDN** *root-name* [**password** *password*] [**port** *tcp-port* [**timeout** *seconds*] | [**timeout** *seconds*]] **Example:** `switch(config)# ldap-server host 10.10.1.1 rootDN cn=manager,dc=acme,dc=com password Ur2Gd2BH timeout 60`	and the bind password for the root. Optionally specifies the TCP port to use for LDAP messages to the server. The range is from 1 to 65535, and the default TCP port is the global value or 389 if a global value is not configured. Also specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show ldap-server** **Example:** `switch# show ldap-server`	Displays the LDAP server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling LDAP.
Configuring LDAP Server Hosts.

5.7.5 Configuring LDAP Server Groups

You can specify one or more remote AAA servers to authenticate users using server groups. All members of a group must be configured to use LDAP. The servers are tried in the same order in which you configure them.

You can configure these server groups at any time, but they take effect only when you apply them to an AAA service.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **aaa group server ldap** *group-name*
3. [**no**] **server** {*ipv4-address* | *ipv6-address* | *host-name*}
4. (Optional) [**no**] **authentication** {**bind-first** [**append-with-baseDN** *DNstring*] | **compare** [**password-attribute** *password*]}
5. (Optional) [**no**] **enable user-server-group**
6. (Optional) [**no**] **enable Cert-DN-match**
7. (Optional) [**no**] **use-vrf** *vrf-name*
8. **exit**
9. (Optional) **show ldap-server groups**

10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **aaa group server ldap** *group-name*
	Example:
	`switch(config)# aaa group server ldap LDAPServer1`
	`switch(config-ldap)#`
		Creates an LDAP server group and enters the LDAP server group configuration mode for that group.

Step 3	[**no**] **server** {*ipv4-address* | *ipv6-address* | *host-name*} **Example:** `switch(config-ldap)# server 10.10.2.2`	Configures the LDAP server as a member of the LDAP server group. If the specified LDAP server is not found, configure it using the **ldap-server host** command and retry this command.
Step 4	(Optional) [**no**] **authentication** {**bind-first** [**append-with-baseDN** *DNstring*] | **compare** [**password-attribute** *password*]} **Example:** `switch(config-ldap)# authentication compare password-attribute TyuL8r`	Performs LDAP authentication using the bind or compare method. The default LDAP authentication method is the bind method using first search and then bind.
Step 5	(Optional) [**no**] **enable user-server-group** **Example:** `switch(config-ldap)# enable user-server-group`	Enables group validation. The group name should be configured in the LDAP server. Users can login through public-key authentication only if the username is listed as a member of this configured group in the LDAP server.
Step 6	(Optional) [**no**] **enable Cert-DN-match** **Example:** `switch(config-ldap)# enable Cert-DN-match`	Enables users to login only if the user profile lists the subject-DN of the user certificate as authorized for login.
Step 7	(Optional) [**no**] **use-vrf** *vrf-name* **Example:** `switch(config-ldap)# use-vrf vrf1`	Specifies the VRF to use to contact the servers in the server group. **Note** This command is supported only on Inspur CN12700 Series Switches.

Step 8	**exit**
	Example:
	`switch(config-ldap)# exit`
	`switch(config)#`
		Exits LDAP server group configuration mode.

Step 9	(Optional) **show ldap-server groups** **Example:**	Displays the LDAP server group configuration.

	Command or Action	Purpose
	`switch(config)# show ldap-server groups`	
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling LDAP.
Configuring LDAP Server Hosts.

5.7.6 Configuring the Global LDAP Timeout Interval

You can set a global timeout interval that determines how long the Inspur INOS device waits for responses from all LDAP servers before declaring a timeout failure.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server timeout** *seconds*
3. **exit**
4. (Optional) **show ldap-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server timeout** *seconds* **Example:** `switch(config)# ldap-server timeout 10`	Specifies the timeout interval for LDAP servers. The default timeout interval is 5 seconds. The range is from 1 to 60 seconds.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show ldap-server** **Example:** `switch# show ldap-server`	Displays the LDAP server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch# copy running-config startup-config`	

Related Topics

Enabling LDAP.
Configuring the Timeout Interval for an LDAP Server.

5.7.7 Configuring the Timeout Interval for an LDAP Server

You can set a timeout interval that determines how long the Inspur INOS device waits for responses from an LDAP server before declaring a timeout failure.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds*
3. **exit**
4. (Optional) **show ldap-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **timeout** *seconds* **Example:** `switch(config)# ldap-server host server1 timeout 10`	Specifies the timeout interval for a specific server. The default is the global value. **Note** The timeout interval value specified for an LDAP server overrides the global timeout interval value specified for all LDAP servers.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show ldap-server** **Example:** `switch# show ldap-server`	Displays the LDAP server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling LDAP.
Configuring the Global LDAP Timeout Interval.

5.7.8 Configuring the Global LDAP Server Port

You can configure a global LDAP server port through which clients initiate TCP connections. By default, Inspur INOS devices use port 389 for all LDAP requests.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server port** *tcp-port*
3. **exit**
4. (Optional) **show ldap-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server port** *tcp-port* **Example:** `switch(config)# ldap-server port 2`	Specifies the global TCP port to use for LDAP messages to the server. The default TCP port is 389. The range is from 1 to 65535.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show ldap-server** **Example:** `switch# show ldap-server`	Displays the LDAP server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling LDAP.
Configuring TCP Ports.

5.7.9 Configuring TCP Ports

You can configure another TCP port for the LDAP servers if there are conflicts with another application. By default, Inspur INOS devices use port 389 for all LDAP requests.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **port** *tcp-port* [**timeout** *seconds*]
3. **exit**
4. (Optional) **show ldap-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **port** *tcp-port* [**timeout** *seconds*] **Example:** `switch(config)# ldap-server host 10.10.1.1 port 200 timeout 5`	Specifies the TCP port to use for LDAP messages to the server. The default TCP port is 389. The range is from 1 to 65535. Optionally specifies the timeout interval for the server. The range is from 1 to 60 seconds, and the default timeout is the global value or 5 seconds if a global value is not configured. **Note** The timeout interval value specified for an LDAP server overrides the global timeout interval value specified for all LDAP servers.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show ldap-server** **Example:** `switch# show ldap-server`	Displays the LDAP server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling LDAP.
Configuring the Global LDAP Server Port.

5.7.10 Configuring LDAP Search Maps

You can configure LDAP search maps to send a search query to the LDAP server. The server searches its database for data meeting the criteria specified in the search map.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. **ldap search-map** *map-name*
3. (Optional) [**userprofile** | **trustedCert** | **CRLLookup** | **user-certdn-match** | **user-pubkey-match** | **user-switch-bind**] **attribute-name** *attribute-name* **search-filter** *filter* **base-DN** *base-DN-name*
4. **exit**
5. (Optional) **show ldap-search-map**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**ldap search-map** *map-name*
	Example:
	`switch(config)# ldap search-map map1`
	`switch(config-ldap-search-map)#`
		Configures an LDAP search map.

Step 3	(Optional) [**userprofile** | **trustedCert** | **CRLLookup** | **user-certdn-match** | **user-pubkey-match** | **user-switch-bind**] **attribute-name** *attribute-name* **search-filter** *filter* **base-DN** *base-DN-name* **Example:** `switch(config-ldap-search-map)# userprofile attribute-name att-name search-filter (&(objectClass=inetOrgPerson)(cn=$userid)) base-DN dc=acme,dc=com`	Configures the attribute name, search filter, and base-DN for the user profile, trusted certificate, CRL, certificate DN match, public key match, or user-switchgroup lookup search operation. These values are used to send a search query to the LDAP server. The *attribute-name* argument is the name of the attribute in the LDAP server that contains the CN role definition.

Step 4	**exit**
	Example:
	`switch(config-ldap-search-map)# exit`
	`switch(config)#`
		Exits LDAP search map configuration mode.

Step 5	(Optional) **show ldap-search-map** **Example:** `switch(config)# show ldap-search-map`	Displays the configured LDAP search maps.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling LDAP.

5.7.11 Configuring Periodic LDAP Server Monitoring

You can monitor the availability of LDAP servers. The configuration parameters include the username and password to use for the server, the rootDN to bind to the server to verify its state, and an idle timer. The idle timer specifies the interval in which an LDAP server receives no requests before the Inspur INOS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test rootDN** *root-name* [**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]]
3. [**no**] **ldap-server deadtime** *minutes*
4. **exit**
5. (Optional) **show ldap-server**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **test rootDN** *root-name* [**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]] **Example:** `switch(config)# ldap-server host 10.10.1.1 test rootDN root1 username user1 password Ur2Gd2BH idle-time 3`	Specifies the parameters for server monitoring. The default username is test, and the default password is Inspur. The default value for the idle timer is 60 minutes, and the valid range is from 1 to 1440 minutes. **Note** We recommend that the user not be an existing user in the LDAP server database.
Step 3	[**no**] **ldap-server deadtime** *minutes* **Example:** `switch(config)# ldap-server deadtime 5`	Specifies the number of minutes before the Inspur INOS device checks an LDAP server that was previously unresponsive. The default value is 0 minutes, and the valid range is from 1 to 60 minutes.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

	Command or Action	Purpose
Step 5	(Optional) **show ldap-server** **Example:** `switch# show ldap-server`	Displays the LDAP server configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling LDAP.
Configuring LDAP Server Hosts.

5.7.12 Configuring the LDAP Dead-Time Interval

You can configure the dead-time interval for all LDAP servers. The dead-time interval specifies the time that the Inspur INOS device waits, after declaring that an LDAP server is dead, before sending out a test packet to determine if the server is now alive.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ldap-server deadtime** *minutes*
3. **exit**
4. (Optional) **show ldap-server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ldap-server deadtime** *minutes* **Example:** `switch(config)# ldap-server deadtime 5`	Configures the global dead-time interval. The default value is 0 minutes. The range is from 1 to 60 minutes.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show ldap-server** **Example:** `switch# show ldap-server`	Displays the LDAP server configuration.

	Command or Action	Purpose
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling LDAP.

5.7.13 Configuring AAA Authorization on LDAP Servers

You can configure the default AAA authorization method for LDAP servers.

Before you begin

Enable LDAP.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authorization** {**ssh-certificate** | **ssh-publickey**} **default** {**group** *group-list* | **local**}
3. **exit**
4. (Optional) **show aaa authorization** [**all**]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authorization** {**ssh-certificate** | **ssh-publickey**} **default** {**group** *group-list* | **local**} **Example:** `switch(config)# aaa authorization ssh-certificate default group LDAPServer1 LDAPServer2`	Configures the default AAA authorization method for the LDAP servers. The **ssh-certificate** keyword configures LDAP or local authorization with certificate authentication, and the **ssh-publickey** keyword configures LDAP or local authorization with the SSH public key. The default authorization is local authorization, which is the list of authorized commands for the user's assigned role. The *group-list* argument consists of a space-delimited list of LDAP server group names. Servers that belong to this group are contacted for AAA authorization. The **local** method uses the local database for authorization.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

	Command or Action	Purpose
Step 4	(Optional) **show aaa authorization** [**all**] **Example:** `switch(config)# show aaa authorization`	Displays the AAA authorization configuration. The **all** keyword displays the default values.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling LDAP.

5.7.14 Disabling LDAP

You can disable LDAP.

Caution	When you disable LDAP, all related configurations are automatically discarded.

SUMMARY STEPS

1. **configure terminal**
2. **no feature ldap**
3. **exit**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no feature ldap** **Example:** `switch(config)# no feature ldap`	Disables LDAP.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

5.8 Monitoring LDAP Servers

You can monitor the statistics that the Inspur INOS device maintains for LDAP server activity.

Before you begin

Configure LDAP servers on the Inspur INOS device.

SUMMARY STEPS

1. **show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch# show ldap-server statistics 10.10.1.1`	Displays the LDAP statistics.

Related Topics

Configuring LDAP Server Hosts.
Clearing LDAP Server Statistics.

5.9 Clearing LDAP Server Statistics

You can display the statistics that the Inspur INOS device maintains for LDAP server activity.

Before you begin

Configure LDAP servers on the Inspur INOS device.

SUMMARY STEPS

1. (Optional) **show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}
2. **clear ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **show ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch# show ldap-server statistics 10.10.1.1`	Displays the LDAP server statistics on the Inspur INOS device.
Step 2	**clear ldap-server statistics** {*hostname* | *ipv4-address* | *ipv6-address*} **Example:** `switch# clear ldap-server statistics 10.10.1.1`	Clears the LDAP server statistics.

Related Topics

Configuring LDAP Server Hosts.
Monitoring LDAP Servers.

5.10 Verifying the LDAP Configuration

To display LDAP configuration information, perform one of the following tasks:

Command	Purpose
show running-config ldap [**all**]	Displays the LDAP configuration in the running configuration.
show startup-config ldap	Displays the LDAP configuration in the startup configuration.
show ldap-server	Displays LDAP configuration information.
show ldap-server groups	Displays LDAP server group configuration information.
show ldap-server statistics {*host-name*| *ipv4-address* | *ipv6-address*}	Displays LDAP statistics.
show ldap-search-map	Displays information about the configured LDAP attribute maps.

For detailed information about the fields in the output from this command, see the *Inspur CN12700 Series INOS Security Command Reference*.

5.11 Configuration Examples for LDAP

The following example shows how to configure an LDAP server host and server group:

```
feature ldap
ldap-server host 10.10.2.2 enable-ssl
aaa group server ldap LdapServer
    server 10.10.2.2
exit
show ldap-server
show ldap-server groups
```

The following example shows how to configure an LDAP search map:

```
ldap search-map s0
userprofile attribute-name description search-filter
(&(objectClass=inetOrgPerson)(cn=$userid)) base-DN dc=acme,dc=com
exit
show ldap-search-map
```

The following example shows how to configure AAA authorization with certificate authentication for an LDAP server:

```
aaa authorization ssh-certificate default group LDAPServer1 LDAPServer2
exit
show aaa authorization
```

5.12 Where to Go Next

You can now configure AAA authentication methods to include the server groups.

5.13 Additional References forLDAP

This section includes additional information related to implementing LDAP.

Related Documents

Related Topic	Document Title
Inspur INOS licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*
VRF configuration	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
• INSPUR-AAA-SERVER-MIB • INSPUR-AAA-SERVER-EXT-MIB	-

5.14 Feature History for LDAP

This table lists the release history for this feature.

Table 12: Feature History for LDAP

Feature Name	Releases	Feature Information	
LDAP	8.2(3)	No change from Release 8.2(3).	

CHAPTER 6 Configuring SSH and Telnet

This chapter describes how to configure Secure Shell Protocol (SSH) and Telnet on Inspur INOS devices. This chapter includes the following sections:

- Finding Feature Information.
- Information About SSH and Telnet.
- Virtualization Support for SSH and Telnet.
- Licensing Requirements for SSH and Telnet.
- Prerequisites for SSH and Telnet.
- Guidelines and Limitations for SSH and Telnet.
- Default Settings for SSH and Telnet.
- Configuring SSH .
- Configuring Telnet.
- Verifying the SSH and Telnet Configuration.
- Configuration Example for SSH.
- Additional References for SSH and Telnet.

6.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

6.2 Information About SSH and Telnet

This section includes information about SSH and Telnet.

6.2.1 SSH Server

You can use the SSH server to enable an SSH client to make a secure, encrypted connection to a Inspur INOS device. SSH uses strong encryption for authentication. The SSH server in the Inspur INOS software can interoperate with publicly and commercially available SSH clients.

The user authentication mechanisms supported for SSH are RADIUS, TACACS+, LDAP, and the use of locally stored usernames and passwords.

6.2.2 SSH Client

The SSH client feature is an application that runs over the SSH protocol to provide device authentication and encryption. The SSH client enables a Inspur INOS device to make a secure, encrypted connection to another Inspur INOS device or to any other device that runs the SSH server. This connection provides an outbound connection that is encrypted. With authentication and encryption, the SSH client allows for a secure communication over an insecure network.

The SSH client in the Inspur INOS software works with publicly and commercially available SSH servers.

6.2.3 SSH Server Keys

SSH requires server keys for secure communications to the Inspur INOS device. You can use SSH server keys for the following SSH options:

- SSH version 2 using Rivest, Shamir, and Adelman (RSA) public-key cryptography

- SSH version 2 using the Digital System Algrorithm (DSA) Be sure to have an SSH server key-pair with the appropriate version before enabling the SSH service. You can generate the SSH server key-pair according to the SSH client version used. The SSH service accepts two types of key-pairs for use by SSH version 2:
- The **dsa** option generates the DSA key-pair for the SSH version 2 protocol.
- The **rsa** option generates the RSA key-pair for the SSH version 2 protocol. By default, the Inspur INOS software generates an RSA key using 1024 bits. SSH supports the following public key formats:
- OpenSSH
- IETF Secure Shell (SECSH)
- Public Key Certificate in Privacy-Enhanced Mail (PEM)

Caution	If you delete all of the SSH keys, you cannot start the SSH services.

6.2.4 SSH Authentication Using Digital Certificates

SSH authentication on Inspur INOS devices provide X.509 digital certificate support for host authentication. An X.509 digital certificate is a data item that ensures the origin and integrity of a message. It contains encryption keys for secured communications and is signed by a trusted certification authority (CA) to verify the identity of the presenter. The X.509 digital certificate support provides either DSA or RSA algorithms for authentication.

The certificate infrastructure uses the first certificate that supports the Secure Socket Layer (SSL) and is returned by the security infrastructure, either through a query or a notification. Verification of certificates is successful if the certificates are from any of the trusted CAs configured and if not revoked or expired.

You can configure your device for either SSH authentication using an X.509 certificate or SSH authentication using a Public Key Certificate, but not both. If either of them is configured and the authentication fails, you are prompted for a password

From Inspur INOS Release 8.2(3), you can configure SSH authentication using X.509v3 certificates (RFC 6187). X.509v3 certificate-based SSH authentication uses certificates combined with a smartcard to enable two-factor authentication for Inspur device access. The SSH client is provided by Inspur partner Pragma Systems.

6.2.5 Telnet Server

The Telnet protocol enables TCP/IP connections to a host. Telnet allows a user at one site to establish a TCP connection to a login server at another site and then passes the keystrokes from one device to the other. Telnet can accept either an IP address or a domain name as the remote device address.

The Telnet server is disabled by default on the Inspur INOS device.

6.3 Virtualization Support for SSH and Telnet

SSH and Telnet configuration and operation are local to the virtual device context (VDC). For more information on VDCs, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

6.4 Licensing Requirements for SSH and Telnet

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS	SSH and Telnet require no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

6.5 Prerequisites for SSH and Telnet

SSH and Telnet have the following prerequisites:

- You have configured IP on a Layer 3 interface, out-of-band on the mgmt 0 interface, or inband on an Ethernet interface.

6.6 Guidelines and Limitations for SSH and Telnet

SSH and Telnet have the following configuration guidelines and limitations:

- The Inspur INOS software supports only SSH version 2 (SSHv2).
- You can configure your device for either SSH authentication using an X.509 certificate or SSH authentication using a public key certificate but not both. If either of them is configured and the authentication fails, you are prompted for a password.
- Static CRL is the only supported revocation check method.
- You need to follow the Open SSL format for the SSH X.509 certificate distinguished name.
- To obtain the Bash shell, only non-root users can login by using Telnet and SSH, and use the **run bash** command. If you want to run any command in the Bash shell with the root privilege, you need to use **sudo** *command-name*.
- Starting from Inspur INOS Release 8.2(3), you can use 4096 bit RSA keys to secure SSH, SCP and SFTP sessions.
- The SSH public and private keys imported into user accounts that are remotely authenticated through a AAA protocol (such as RADIUS or TACACS+) for the purpose of SSH passwordless file copy will not persist when the Inspur CN device is reloaded.

6.7 Default Settings for SSH and Telnet

This table lists the default settings for SSH and Telnet parameters.

Table 13：Default SSH and Telnet Parameters

Parameters	**Default**
SSH server	Enabled
SSH server key	RSA key generated with 1024 bits
RSA key bits for generation	1024
Telnet server	Disabled
Telnet port number	23

6.8 Configuring SSH

This section describes how to configure SSH.

6.8.1 Generating SSH Server Keys

You can generate an SSH server key based on your security requirements. The default SSH server key is an RSA key that is generated using 1024 bits.

SUMMARY STEPS

1. **configure terminal**
2. **no feature ssh**
3. **ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]}

4. feature ssh
5. **exit**
6. (Optional) **show ssh key**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.
Step 3	**ssh key** {**dsa** [**force**] | **rsa** [*bits* [**force**]]} **Example:** `switch(config)# ssh key rsa 2048`	Generates the SSH server key. The *bits* argument is the number of bits used to generate the RSA key. The range is from 768 to 2048. Starting from Inspur INOS Release 8.2(3), the range is from 1024 to 4096. The default value is 1024. You cannot specify the size of the DSA key. It is always set to 1024 bits. Use the **force** keyword to replace an existing key.
Step 4	**feature ssh** **Example:** `switch(config)# feature ssh`	Enables SSH.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 6	(Optional) **show ssh key** **Example:** `switch# show ssh key`	Displays the SSH server keys.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

6.8.2 Specifying the SSH Public Keys for User Accounts

You can configure an SSH public key to log in using an SSH client without being prompted for a password. You can specify the SSH public key in one of these formats:

- OpenSSH format
- IETF SECSH format

- Public Key Certificate in PEM format

Specifying the SSH Public Keys in IETF SECSH Format

You can specify the SSH public keys in IETF SECSH format for user accounts.

Before you begin

Generate an SSH public key in IETF SCHSH format.

SUMMARY STEPS

1. **copy** *server-file* **bootflash:***filename*
2. **configure terminal**
3. **username** *username* **sshkey file bootflash:***filename*
4. **exit**
5. (Optional) **show user-account**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**copy** *server-file* **bootflash:***filename* **Example:** `switch# copy tftp://10.10.1.1/secsh_file.pub bootflash:secsh_file.pub`	Downloads the file containing the SSH key in IETF SECSH format from a server. The server can be FTP, secure copy (SCP), secure FTP (SFTP), or TFTP.

Step 2	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 3	**username** *username* **sshkey file bootflash:***filename* **Example:** `switch(config)# username User1 sshkey file bootflash:secsh_file.pub`	Configures the SSH public key in IETF SECSH format.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 5	(Optional) **show user-account** **Example:** `switch# show user-account`	Displays the user account configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Specifying the SSH Public Keys in OpenSSH Format

You can specify the SSH public keys in OpenSSH format for user accounts.

Before you begin

Generate an SSH public key in OpenSSH format.

SUMMARY STEPS

1. **configure terminal**
2. **username** *username* **sshkey** *ssh-key*
3. **exit**
4. (Optional) **show user-account**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.
Step 2	**username** *username* **sshkey** *ssh-key* **Example:** `switch(config)# username User1 sshkey ssh-rsa`	Configures the SSH public key in OpenSSH format.
Step 3	**exit** **Example:** `switch(config)# exit switch#`	Exits global configuration mode.
Step 4	(Optional) **show user-account** **Example:** `switch# show user-account`	Displays the user account configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

6.8.3 Configuring a Login Grace Time for SSH Connections

You can configure the login grace time for SSH connections from remote devices to your Inspur INOS device. This configures the grace time for clients to authenticate themselves. If the time to login to the SSH session exceeds the specified grace time, the session disconnects and you will need to attempt logging in again.

SUMMARY STEPS

1. **configure terminal**
2. **feature ssh**
3. **ssh login-gracetime** *number*
4. (Optional) **exit**
5. (Optional) **show running-config security**
6. (Optional) **show running-config security all**

7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature ssh**
	Example:
	`switch# feature ssh`
	`switch(config)#`
		Enables SSH.

Step 3	**ssh login-gracetime** *number* **Example:** `switch(config)# ssh login-gracetime 120`	Configures the login grace time in seconds for SSH connections from remote devices to your Inspur INOS device. The default login grace time is 120 seconds. The range is from 1 to 2147483647. **Note** The **no** form of this command removes the configured login grace time and resets it to the default value of 120 seconds.
Step 4	(Optional) **exit** **Example:** `switch(config)# exit`	Exits global configuration mode.
Step 5	(Optional) **show running-config security** **Example:** `switch(config)# show running-config security`	Displays the configured SSH login grace time.
Step 6	(Optional) **show running-config security all** **Example:** `switch(config)# show running-config security all`	Displays the configured or default SSH login grace time.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Starting SSH Sessions

You can start SSH sessions using IPv4 or IPv6 to connect to remote devices from the Inspur INOS device.

Before you begin

Obtain the hostname for the remote device and, if needed, the username on the remote device. Enable the SSH server on the remote device.

SUMMARY STEPS

1. **ssh** [*username@*]{*ipv4-address* | *hostname*} [**vrf** *vrf-name*]
2. **ssh6** [*username@*]{*ipv6-address* | *hostname*} [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ssh** [*username@*]{*ipv4-address*|*hostname*} [**vrf** *vrf-name*] **Example:** `switch# ssh 10.10.1.1`	Creates an SSH IPv4 session to a remote device using IPv4. The default VRF is the default VRF.
Step 2	**ssh6** [*username@*]{*ipv6-address* | *hostname*} [**vrf** *vrf-name*] **Example:** `switch# ssh6 HostA`	Creates an SSH IPv6 session to a remote device using IPv6.

6.8.4 Configuring X.509v3 Certificate-Based SSH Authentication

Use this task to configure X.509v3 certificate-based SSH authentication.

Before you begin

Enable the SSH server on the remote device.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Configure a user account:

switch(config)# **username** *user-id* [**password [0|5]** *password*]

The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 28 characters. Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hyphen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames, but not in local usernames.

Usernames must begin with an alphanumeric character. The default password is undefined. The 0 option indicates that the password is clear text, and the 5 option indicates that the password is encrypted. The default is 0 (clear text).

Note When a password is not configured, the user can login to the Inspur INOS switch only via X.509v3 based user certificates.

Note If you create a user account with the encrypted password option, the corresponding SNMP user will not be created.

Step 3 Specify an SSH X.509 certificate distinguished name and RSA algorithm to use for authentication for an existing user account:

switch(config)# **username** *user-id* **ssh-cert-dn** *dn-name* **rsa**

The distinguished name can be up to 512 characters and must follow the Open SSL format shown in the example after this procedure. Make sure that the email address and state are configured as emailAddress and ST, respectively.

Step 4 Configure a trustpoint:

switch(config)# **[no] crypto ca trustpoint** *trustpoint*

Note Before you delete a trustpoint using the no form of this command, you must first delete the CRL and CA certificate, using the **delete crl** and **delete ca-certificate** commands.

Step 5 Configure a CA certificate for the trustpoint:

switch(config-trustpoint)# **crypto ca authenticate** *trustpoint*

Note To delete a CA certificate, enter the **delete ca-certificate** command in the trustpoint configuration mode.

Step 6 (Optional) Configure the certificate revocation list (CRL) for the trustpoint:

switch(config-trustpoint)# **crypto ca crl request** *trustpoint* **bootflash:***static-crl***.crl**

This command is optional but highly recommended.

The CRL file is a snapshot of the list of revoked certificates by the trustpoint. This static CRL list is manually copied to the device from the Certification Authority (CA).

Note Static CRL is the only supported revocation check

Note method. To delete the CRL, enter the **delete crl** command.

Step 7 Exit global configuration mode:

switch(config-trustpoint)# **exit**

switch(config)# **exit**

Step 8 (Optional) Display the configured certificate chain and associated trustpoint:

switch# **show crypto ca certificates**

Step 9 (Optional) Display the contents of the CRL list of the specified trustpoint:
switch# **show crypto ca crl** *trustpoint*

Step 10 (Optional) Display configured user account details:
switch# **show user-account**

Step 11 (Optional) Display the users logged into the device:
switch# **show users**

Step 12 (Optional) Copy the running configuration to the startup configuration:
switch# **copy running-config startup-config**

Example: Configuring X.509v3 Certificate-Based SSH Authentication

The following running configuration shows how to configure X.509v3 certificate-based SSH authentication. Replace the *placeholders* with relevant values for your setup.

```
configure terminal
 username <jsmith> password <4Ty18Rnt>
 username <jsmith> ssh-cert-dn <"/O = ABCcompany, OU = ABC1,
 emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith">
 rsa
 crypto ca trustpoint <tp1>
  crypto ca authenticate <tp1>
  crypto ca crl request <tp1> bootflash:<crl1>.crl
  exit
 exit
```

The following example shows how to check information about the CA certificates and user accounts.

```
switch# show crypto ca certificates
Trustpoint: tp1
CA certificate 0:
subject= /CN=SecDevCA
issuer= /CN=SecDevCA
serial=01AB02CD03EF04GH05IJ06KL07MN
```

```
notBefore=Jun 29 12:36:26 2016 GMT
notAfter=Jun 29 12:46:23 2021 GMT
SHA1 Fingerprint=47:29:E3:00:C1:C1:47:F2:56:8B:AC:B2:1C:64:48:FC:F4:8D:53:AF
purposes: sslserver sslclient

switch# show crypto ca crl tp1
Trustpoint: tp1 CRL: Certificate Revocation List (CRL):
Version 2 (0x1)
Signature Algorithm: sha1WithRSAEncryption
Issuer: /CN=SecDevCA
Last Update: Aug 8 20:03:15 2016 GMT
Next Update: Aug 16 08:23:15 2016 GMT
CRL extensions:
X509v3 Authority Key Identifier:
keyid:30:43:AA:80:10:FE:72:00:DE:2F:A2:17:E4:61:61:44:CE:78:FF:2A

switch# show user-account
user:user1
this user account has no expiry date
roles:network-operator
ssh cert DN : /C = US, ST = New York, L = Metropolis, O = Inspur , OU = csg, CN =
user1; Algo: x509v3-sign-rsa

switch# show users
NAME LINE TIME IDLE PID COMMENT
user1 pts/1 Jul 27 18:43 00:03 18796 (10.10.10.1) session=ssh
```

6.8.5 Clearing SSH Hosts

When you download a file from a server using SCP or SFTP, or when you start an SSH session from this device to a remote host, you establish a trusted SSH relationship with that server. You can clear the list of trusted SSH servers for your user account.

SUMMARY STEPS

1. **clear ssh hosts**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear ssh hosts** **Example:** `switch# clear ssh hosts`	Clears the SSH host sessions and the known host file.

Disabling the SSH Server

By default, the SSH server is enabled on the Inspur INOS device. You can disable the SSH server to prevent SSH access to the switch.

SUMMARY STEPS

1. **configure terminal**
2. **no feature ssh**
3. **exit**
4. (Optional) **show ssh server**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show ssh server** **Example:** `switch# show ssh server`	Displays the SSH server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

6.8.6 Deleting SSH Server Keys

You can delete SSH server keys on the Inspur INOS device after you disable the SSH server.

SUMMARY STEPS

1. **configure terminal**
2. **no feature ssh**
3. **no ssh key** [**dsa** | **rsa**]
4. **exit**
5. (Optional) **show ssh key**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no feature ssh** **Example:** `switch(config)# no feature ssh`	Disables SSH.
Step 3	**no ssh key** [**dsa** | **rsa**] **Example:** `switch(config)# no ssh key rsa`	Deletes the SSH server key. The default is to delete all the SSH keys.

	Command or Action	Purpose

Step 4	**exit**
	Example:
	`switch(config)#`
	`exit switch#`
		Exits global configuration mode.

Step 5	(Optional) **show ssh key** **Example:** `switch# show ssh key`	Displays the SSH server key configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Generating SSH Server Keys.

6.8.7 Clearing SSH Sessions

You can clear SSH sessions from the Inspur INOS device.

SUMMARY STEPS

1. **show users**
2. **clear line** *vty-line*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	Displays user session information.
Step 2	**clear line** *vty-line* **Example:** `switch(config)# clear line pts/12`	Clears a user SSH session.

6.9 Configuring Telnet

This section describes how to configure Telnet on the Inspur INOS device.

6.9.1 Enabling the Telnet Server

You can enable the Telnet server on the Inspur INOS device. By default, the Telnet server is disabled.

SUMMARY STEPS

1. **configure terminal**
2. **feature telnet**
3. **exit**
4. (Optional) **show telnet server**

5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature telnet** **Example:** `switch(config)# feature telnet`	Enables the Telnet server. The default is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show telnet server** **Example:** `switch# show telnet server`	Displays the Telnet server configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

6.9.2 Starting Telnet Sessions to Remote Devices

You can start Telnet sessions to connect to remote devices from the Inspur INOS device. You can start Telnet sessions using either IPv4 or IPv6.

Before you begin

Obtain the hostname or IP address for the remote device and, if needed, the username on the remote device. Enable the Telnet server on the Inspur INOS device.

Enable the Telnet server on the remote device.

SUMMARY STEPS

1. **telnet** {*ipv4-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*]
2. **telnet6** {*ipv6-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**telnet** {*ipv4-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*] **Example:** `switch# telnet 10.10.1.1`	Starts a Telnet session to a remote device using IPv4. The default port number is 23. The range is from 1 to 65535. The default VRF is the default VRF.
Step 2	**telnet6** {*ipv6-address* | *host-name*} [*port-number*] [**vrf** *vrf-name*]	Starts a Telnet session to a remote device using IPv6. The default port number is 23. The range is from 1 to 65535. The default VRF is the default

	Command or Action	Purpose
	Example: `switch# telnet6 2001:0DB8::ABCD:1 vrf management`	VRF.

Related Topics

Enabling the Telnet Server.

6.9.3 Clearing Telnet Sessions

You can clear Telnet sessions from the Inspur INOS device.

Before you begin

Enable the Telnet server on the Inspur INOS device.

SUMMARY STEPS

1. **show users**
2. **clear line** *vty-line*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	Displays user session information.
Step 2	**clear line** *vty-line* **Example:** `switch(config)# clear line pts/12`	Clears a user Telnet session.

6.10 Verifying the SSH and Telnet Configuration

To display the SSH and Telnet configuration information, perform one of the following tasks:

For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

6.11 Configuration Example for SSH

The following example shows how to configure SSH with an OpenSSH key:

Step 1 Disable the SSH server.

Example:

```
switch# configure terminal
switch(config)# no feature ssh
```

Step 2 Generate an SSH server key.

Example:

```
switch(config)#  ssh  key  rsa
generating rsa key(1024 bits)......
generated rsa key
```

Step 3 Enable the SSH server.

Example:

```
switch(config)# feature ssh
```

Step 4 Display the SSH server key.

Example:

```
switch(config)# show ssh key
rsa Keys generated:Sat Sep 29 00:10:39 2007

ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAvWhEBsF55oaPHNDBnpXOTw6+/OdHoLJZKr
+MZm99n2U0ChzZG4svRWmHuJY4PeDWl0e5yE3g3EO3pjDDmt923siNiv5aSga60K36lr39
HmXL6VgpRVn1XQFiBwn4na+H1d3Q0hDt+uWEA0tka2uOtXlDhliEmn4HVXOjGhFhoNE=

bitcount:1024
fingerprint:
51:6d:de:1c:c3:29:50:88:df:cc:95:f0:15:5d:9a:df
**************************************
could not retrieve dsa key information
**************************************
```

Step 5 Specify the SSH public key in OpenSSH format.

Example:

```
switch(config)#  username  User1  sshkey  ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAy19oF6QaZl9G+3f1XswK3OiW4H7YyUyuA50r
v7gsEPjhOBYmsi6PAVKui1nIf/DQhum+lJNqJP/eLowb7ubO+lVKRXFY/G+lJNIQ
W3g9igG30c6k6+XVn+NjnI1B7ihvpVh7dLddMOXwOnXHYshXmSiH3UD/vKyziEh5
4Tplx8=
```

Step 6 Save the configuration.

Example:

```
switch(config)# copy running-config startup-config
```

6.12 Additional References for SSH and Telnet

This section describes additional information related to implementing SSH and Telent.

Related Documents

Related Topic	Document Title
Inspur INOS licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*

VRF configuration	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
• INSPUR-SECURE-SHELL-MIB	-

CHAPTER 7 Configuring PKI

This chapter describes the Public Key Infrastructure (PKI) support on the Inspur INOS device. PKI allows the device to obtain and use digital certificates for secure communication in the network and provides manageability and scalability for Secure Shell (SSH).

This chapter includes the following sections:

- Finding Feature Information.
- Information About PKI.
- Virtualization Support for PKI.
- Licensing Requirements for PKI.
- Guidelines and Limitations for PKI.
- Default Settings for PKI.
- Configuring CAs and Digital Certificates.
- Verifying the PKI Configuration.
- Configuration Examples for PKI.
- Additional References for PKI.

7.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

7.2 Information About PKI

This section provides information about PKI.

7.2.1 CAs and Digital Certificates

Certificate authorities (CAs) manage certificate requests and issue certificates to participating entities such as hosts, network devices, or users. The CAs provide centralized key management for the participating entities.

Digital signatures, based on public key cryptography, digitally authenticate devices and individual users. In public key cryptography, such as the RSA encryption system, each device or user has a key pair that contains both a private key and a public key. The private key is kept secret and is known only to the owning device or user only. However, the public key is known to everybody. Anything encrypted with one of the keys can be decrypted with the other. A signature is formed when data is encrypted with a sender's private key. The receiver verifies the signature by decrypting the message with the sender's public key. This process relies on the receiver having a copy of the sender's public key and knowing with a high degree of certainty that it really does belong to the sender and not to someone pretending to be the sender.

Digital certificates link the digital signature to the sender. A digital certificate contains information to identify a user or device, such as the name, serial number, company, department, or IP address. It also contains a copy of the entity's public key. The CA that signs the certificate is a third party that the receiver explicitly trusts to validate identities and to create digital certificates.

To validate the signature of the CA, the receiver must first know the CA's public key. Typically, this process is handled out of band or through an operation done at installation. For instance, most web browsers are configured with the public keys of several CAs by default.

7.2.2 Trust Model, Trust Points, and Identity CAs

The PKI trust model is hierarchical with multiple configurable trusted CAs. You can configure each participating device with a list of trusted CAs so that a peer certificate obtained during the security protocol exchanges can be authenticated if it was issued by one of the locally trusted CAs. The Inspur INOS software locally stores the self-signed root certificate of the trusted CA (or certificate chain for a subordinate CA). The process of securely obtaining a trusted CA's root certificate (or the entire chain in the case of a subordinate CA) and storing it locally is called *CA authentication*.

The information about a trusted CA that you have configured is called the *trust point* and the CA itself is called a *trust point CA*. This information consists of a CA certificate (or certificate chain in case of a subordinate CA) and certificate revocation checking information.

The Inspur INOS device can also enroll with a trust point to obtain an identity certificate to associate with a key pair. This trust point is called an *identity CA*.

7.2.3 RSA Key Pairs and Identity Certificates

You can obtain an identity certificate by generating one or more RSA key pairs and associating each RSA key pair with a trust point CA where the Inspur INOS device intends to enroll. The Inspur INOS device needs only one identity per CA, which consists of one key pair and one identity certificate per CA.

The Inspur INOS software allows you to generate RSA key pairs with a configurable key size (or modulus). The default key size is 512. You can also configure an RSA key-pair label. The default key label is the device fully qualified domain name (FQDN).

The following list summarizes the relationship between trust points, RSA key pairs, and identity certificates:

- A trust point corresponds to a specific CA that the Inspur INOS device trusts for peer certificate verification for any application (such as SSH).
- A Inspur INOS device can have many trust points and all applications on the device can trust a peer certificate issued by any of the trust point CAs.
- A trust point is not restricted to a specific application.
- A Inspur INOS device enrolls with the CA that corresponds to the trust point to obtain an identity certificate. You can enroll your device with multiple trust points which means that you can obtain a separate identity certificate from each trust point. The identity certificates are used by applications depending upon the purposes specified in the certificate by the issuing CA. The purpose of a certificate is stored in the certificate as a certificate extension.
- When enrolling with a trust point, you must specify an RSA key pair to be certified. This key pair must be generated and associated to the trust point before generating the enrollment request. The association between the trust point, key pair, and identity certificate is valid until it is explicitly removed by deleting the certificate, key pair, or trust point.
- The subject name in the identity certificate is the fully qualified domain name for the Inspur INOS device.
- You can generate one or more RSA key pairs on a device and each can be associated to one or more trust points. But no more than one key pair can be associated to a trust point, which means only one identity certificate is allowed from a CA.
- If the Inspur INOS device obtains multiple identity certificates (each from a distinct CA), the certificate that an application selects to use in a security protocol exchange with a peer is application specific.
- You do not need to designate one or more trust points for an application. Any application can use any certificate issued by any trust point as long as the certificate purpose satisfies the application requirements.
- You do not need more than one identity certificate from a trust point or more than one key pair to be associated to a trust point. A CA certifies a given identity (or name) only once and does not issue multiple certificates with the same name. If you need more than one identity certificate for a CA and if the CA allows multiple certificates with the same names, you must define another trust point for the same CA, associate another key pair to it, and have it certified.

7.2.4 Multiple Trusted CA Support

The Inspur INOS device can trust multiple CAs by configuring multiple trust points and associating each with a distinct CA. With multiple trusted CAs, you do not have to enroll a device with the specific CA that issued the certificate to a peer. Instead, you can configure the device with multiple trusted CAs that the peer trusts. The Inspur INOS device can then use a configured trusted CA to verify certificates received from a peer that were not issued by the same CA defined in the identity of the peer device.

7.2.5 PKI Enrollment Support

Enrollment is the process of obtaining an identity certificate for the device that is used for applications like SSH. It occurs between the device that requests the certificate and the certificate authority.

The Inspur INOS device performs the following steps when performing the PKI enrollment process:

- Generates an RSA private and public key pair on the device.
- Generates a certificate request in standard format and forwards it to the CA.
- Receives the issued certificate back from the CA, signed with the CA's private key.
- Writes the certificate into a nonvolatile storage area on the device (bootflash).
- Manual Enrollment Using Cut-and-Paste

The Inspur INOS software supports certificate retrieval and enrollment using manual cut-and-paste.

Cut-and-paste enrollment means that you must cut and paste the certificate requests and resulting certificates between the device and the CA.

You must perform the following steps when using cut and paste in the manual enrollment process:

- Create an enrollment certificate request, which the Inspur INOS device displays in base64-encoded text form.
- Cut and paste the encoded certificate request text in an e-mail or in a web form and send it to the CA.
- Receive the issued certificate (in base64-encoded text form) from the CA in an e-mail or in a web browser download.
- Cut and paste the issued certificate to the device using the certificate import facility.

7.2.6 Multiple RSA Key Pair and Identity CA Support

Multiple identity CAs enable the device to enroll with more than one trust point, which results in multiple identity certificates, each from a distinct CA. With this feature, the Inspur INOS device can participate in SSH and other applications with many peers using certificates issued by CAs that are acceptable to those peers.

The multiple RSA key-pair feature allows the device to maintain a distinct key pair for each CA with which it is enrolled. It can match policy requirements for each CA without conflicting with the requirements specified by the other CAs, such as the key length. The device can generate multiple RSA key pairs and associate each key pair with a distinct trust point. Thereafter, when enrolling with a trust point, the associated key pair is used to construct the certificate request.

7.2.7 Peer Certificate Verification

The PKI support on a Inspur INOS device can verify peer certificates. The Inspur INOS software verifies certificates received from peers during security exchanges for applications, such as SSH. The applications verify the validity of the peer certificates. The Inspur INOS software performs the following steps when verifying peer certificates:

- Verifies that the peer certificate is issued by one of the locally trusted CAs.
- Verifies that the peer certificate is valid (not expired) with respect to current time.
- Verifies that the peer certificate is not yet revoked by the issuing CA.

For revocation checking, the Inspur INOS software supports the certificate revocation list (CRL). A trust point CA can use this method to verify that the peer certificate has not been revoked.

7.2.8 Certificate Revocation Checking

The Inspur INOS software can check the revocation status of CA certificates. The applications can use the revocation checking mechanisms in the order that you specify. The choices are CRL, none, or a combination of these methods.

CRL Support

The CAs maintain certificate revocation lists (CRLs) to provide information about certificates revoked prior to their expiration dates. The CAs publish the CRLs in a repository and provide the download public URL in all issued certificates. A client verifying a peer’s certificate can obtain the latest CRL from the issuing CA and use it to determine if the certificate has been revoked. A client can cache the CRLs of some or all of its trusted CAs locally and use them later if necessary until the CRLs expire.

The Inspur INOS software allows the manual configuration of predownloaded CRLs for the trust points, and then caches them in the device bootflash (cert-store). During the verification of a peer certificate, the Inspur INOS software checks the CRL from the issuing CA only if the CRL has already been cached locally and the revocation checking is configured to use the CRL. Otherwise, the Inspur INOS software does not perform CRL checking and considers the certificate to be not revoked unless you have configured other revocation checking methods.

7.2.9 Import and Export Support for Certificates and Associated Key Pairs

As part of the CA authentication and enrollment process, the subordinate CA certificate (or certificate chain) and identity certificates can be imported in standard PEM (base64) format.

The complete identity information in a trust point can be exported to a file in the password-protected PKCS#12 standard format. It can be later imported to the same device (for example, after a system crash) or to a replacement device. The information in a PKCS#12 file consists of the RSA key pair, the identity certificate, and the CA certificate (or chain).

7.3 Virtualization Support for PKI

The configuration and operation of the PKI feature is local to the virtual device context (VDC). For more information on VDCs, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

7.4 Licensing Requirements for PKI

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS	The PKI feature requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

7.5 Guidelines and Limitations for PKI

PKI has the following configuration guidelines and limitations:

- The maximum number of key pairs you can configure on a Inspur INOS device is 16.
- The maximum number of trust points you can declare on a Inspur INOS device is 16.
- The maximum number of identify certificates you can configure on a Inspur INOS device is 16.
- The maximum number of certificates in a CA certificate chain is 10.
- The maximum number of trust points you can authenticate to a specific CA is 10.
- Configuration rollbacks do not support the PKI configuration.

- The Inspur INOS software does not support OSCP.

7.6 Default Settings for PKI

This table lists the default settings for PKI parameters.

Table 14：Default PKI Parameters

Parameters	Default
Trust point	None
RSA key pair	None
RSA key-pair label	Device FQDN
RSA key-pair modulus	512
RSA key-pair exportable	Enabled
Revocation check method	CRL

7.7 Configuring CAs and Digital Certificates

This section describes the tasks that you must perform to allow CAs and digital certificates on your Inspur INOS device to interoperate.

7.7.1 Configuring the Hostname and IP Domain Name

You must configure the hostname and IP domain name of the device if you have not yet configured them because the Inspur INOS software uses the fully qualified domain name (FQDN) of the device as the subject in the identity certificate. Also, the Inspur INOS software uses the device FQDN as a default key label when you do not specify a label during key-pair generation. For example, a certificate named DeviceA.example.com is based on a device hostname of DeviceA and a device IP domain name of example.com.

SUMMARY STEPS

1. **configure terminal**
2. **hostname** *hostname*
3. **ip domain-name** *name* [**use-vrf** *vrf-name*]
4. **exit**
5. (Optional) **show hosts**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**hostname** *hostname* **Example:**	Configures the hostname of the device.

	Command or Action	Purpose
	`switch(config)# hostname DeviceA`	
Step 3	**ip domain-name** *name* [**use-vrf** *vrf-name*] **Example:** `DeviceA(config)# ip domain-name example.com`	Configures the IP domain name of the device. If you do not specify a VRF name, the command uses the default VRF.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show hosts** **Example:** `switch# show hosts`	Displays the IP domain name.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

7.7.2 Generating an RSA Key Pair

You can generate an RSA key pairs to sign and/or encrypt and decrypt the security payload during security protocol exchanges for applications. You must generate the RSA key pair before you can obtain a certificate for your device.

SUMMARY STEPS

1. **configure terminal**
2. **crypto key generate rsa** [**label** *label-string*] [**exportable**] [**modulus** *size*]
3. **exit**
4. (Optional) **show crypto key mypubkey rsa**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto key generate rsa** [**label** *label-string*] [**exportable**] [**modulus** *size*] **Example:** `switch(config)# crypto key generate rsa exportable`	Generates an RSA key pair. The maximum number of key pairs on a device is 16. The label string is alphanumeric, case sensitive, and has a maximum length of 64 characters. The default label string is the hostname and the FQDN separated by a period character (.). Valid modulus values are 512, 768, 1024, 1536, and 2048. Starting from Inspur INOS Release 8.2(3),

	Command or Action	Purpose
		4096 is also a valid modulus value. The default modulus size is 512. **Note** The security policy on the Inspur INOS device and on the CA (where enrollment is planned) should be considered when deciding the appropriate key modulus. By default, the key pair is not exportable. Only exportable key pairs can be exported in the PKCS#12 format. **Caution** You cannot change the exportability of a key pair.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show crypto key mypubkey rsa** **Example:** `switch# show crypto key mypubkey rsa`	Displays the generated key.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

7.7.3 Creating a Trust Point CA Association

You must associate the Inspur INOS device with a trust point CA. Starting from Inspur INOS Release 8.2(3), you can associate a 4096 bit RSA key with a trust point.

Before you begin

Generate the RSA key pair.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca trustpoint** *name*
3. **enrollment terminal**
4. **rsakeypair** *label*
5. **exit**
6. (Optional) **show crypto ca trustpoints**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose

	Example:
	`switch# configure terminal`
	`switch(config)#`	

Step 2	**crypto ca trustpoint** *name*
	Example:
	`switch(config)# crypto ca trustpoint admin-ca`
	`switch(config-trustpoint)#`
		Declares a trust point CA that the device should trust and enters trust point configuration mode.
		Note The maximum number of trust points that you can configure on a device is 16.

Step 3	**enrollment terminal** **Example:** `switch(config-trustpoint)# enrollment terminal`	Enables manual cut-and-paste certificate enrollment. The default is enabled. **Note** **The** Inspur INOS software supports only the manual cut-and-paste method for certificate enrollment.
Step 4	**rsakeypair** *label* **Example:** `switch(config-trustpoint)# rsakeypair SwitchA`	Specifies the label of the RSA key pair to associate to this trust point for enrollment. **Note** You can specify only one RSA key pair per CA.

Step 5	**exit**
	Example:
	`switch(config-trustpoint)# exit`
	`switch(config)#`
		Exits trust point configuration mode.

Step 6	(Optional) **show crypto ca trustpoints** **Example:** `switch(config)# show crypto ca trustpoints`	Displays trust point information.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Generating an RSA Key Pair, on page 160

7.7.4 Authenticating the CA

The configuration process of trusting a CA is complete only when the CA is authenticated to the Inspur INOS device. You must authenticate your Inspur INOS device to the CA by obtaining the self-signed certificate of the CA in PEM format, which contains the public key of the CA. Because the certificate of the CA is self-signed (the CA signs its own certificate) the public key of the CA should be manually authenticated by contacting the CA administrator to compare the fingerprint of the CA certificate.

Before you begin

Create an association with the CA.
Obtain the CA certificate or CA certificate chain.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca authenticate** *name*
3. **exit**
4. (Optional) **show crypto ca trustpoints**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td>Step 1</td><td>configure terminal
Example:
<pre>
switch# configure terminal
switch(config)#
</pre></td><td>Enters global configuration mode.</td></tr>
<tr><td>Step 2</td><td>crypto ca authenticate name
Example:
<pre>
switch(config)# crypto ca authenticate admin-ca input (cut &
paste) CA certificate (chain) in PEM
 format;
end the input with a line containing only END OF INPUT :
-----BEGIN CERTIFICATE-----
MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRIljK0ZejANBgkqhkiG9w0BAQUFADCB
kDEgMB4GCSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklO
MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UE
ChMFQ2lzY28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBD
QTAeFw0wNTA1MDMyMjQ2MzdaFw0wNzA1MDMyMjU1MTdaMIGQMSAwHgYJKoZIhvcN
AQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth
cm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4wDAYDVQQKEwVDaXNjbzETMBEG
A1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBhcm5hIENBMFwwDQYJKoZIhvcN
AQEBBQADSwAwSAJBAMW/7b3+DXJPANBsIHHzluNccNM87ypyzwuoSNZXOMpeRXXI
OzyBAgiXT2ASFuUOwQ1iDM8rO/41jf8RxvYKvysCAwEAAaOBvzCBvDALBgNVHQ8E
BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJyjyRoMbrCNMRU2OyRhQ
GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs
L0FwYXJuYSUyMENBLmNybDAwoC6gLIYqZmlsZTovL1xcc3NlLTA4XENlcnRFbnJv
bGxcQXBhcm5hJTIwQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB
BQUAA0EAHv6UQ+8nE399Tww+KaGr0g0NIJaqNgLh0AFcT0rEyuyt/WYGPzksF9Ea
NBG7E0oN66zex0EOEfG1Vs6mXp1//w==
-----END CERTIFICATE----- END OF INPUT
Fingerprint(s): MD5
Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12
Do you accept this certificate? [yes/no]: yes
</pre></td><td>Prompts you to cut and paste the certificate of the CA. Use the same name that you used when declaring the CA.

The maximum number of trust points that you can authenticate to a specific CA is 10.

Note For subordinate CA authentication, the Inspur INOS software requires the full chain of CA certificates ending in a self-signed CA because the CA chain is needed for certificate verification as well as for PKCS#12 format export.</td></tr>
<tr><td>Step 3</td><td>exit
Example:
<pre>
switch(config)# exit
switch#
</pre></td><td>Exits configuration mode.</td></tr>
<tr><td>Step 4</td><td>(Optional) show crypto ca trustpoints
Example:
<pre>
switch# show crypto ca trustpoints
</pre></td><td>Displays the trust point CA information.</td></tr>
<tr><td>Step 5</td><td>(Optional) copy running-config startup-config
Example:
<pre>
switch# copy running-config startup-config
</pre></td><td>Copies the running configuration to the startup configuration.</td></tr>
</table>

Related Topics

Creating a Trust Point CA Association.

7.7.5 Configuring Certificate Revocation Checking Methods

During security exchanges with a client (for example, an SSH user), the Inspur INOS device performs the certificate verification of the peer certificate sent by the client. The verification process may involve certificate revocation status checking.

You can configure the device to check the CRL downloaded from the CA. Downloading the CRL and checking locally does not generate traffic in your network. However, certificates can be revoked between downloads and your device would not be aware of the revocation.

Before you begin

Authenticate the CA.

Ensure that you have configured the CRL if you want to use CRL checking.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca trustpoint** *name*
3. **revocation-check** {**crl** [**none**] | **none**}
4. **exit**
5. (Optional) **show crypto ca trustpoints**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca trustpoint** *name*
	Example:
	`switch(config)# crypto ca trustpoint admin-ca`
	`switch(config-trustpoint)#`
		Specifies a trust point CA and enters trust point configuration mode.

Step 3	**revocation-check** {**crl** [**none**] | **none**} **Example:** `switch(config-trustpoint)# revocation-check none`	Configures the certificate revocation checking methods. The default method is **crl**. The Inspur INOS software uses the certificate revocation methods in the order that you specify.

Step 4	**exit**
	Example:
	`switch(config-trustpoint)# exit`
	`switch(config)#`
		Exits trust point configuration mode.

Step 5	(Optional) **show crypto ca trustpoints** **Example:** `switch(config)# show crypto ca trustpoints`	Displays the trust point CA information.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the

	Command or Action	Purpose
	Example: `switch(config)# copy running-config startup-config`	startup configuration.

Related Topics

Authenticating the CA.
Configuring a CRL.

7.7.6 Generating Certificate Requests

You must generate a request to obtain identity certificates from the associated trust point CA for each of your device's RSA key pairs. You must then cut and paste the displayed request into an e-mail or in a website form for the CA.

Before you begin

Create an association with the CA.
Obtain the CA certificate or CA certificate chain.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca enroll** *name*
3. **exit**
4. (Optional) **show crypto ca certificates**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca enroll** *name*
	Example:
	`switch(config)# crypto ca enroll admin-ca Create the certificate request ..`
	` Create a challenge password. You will need to verbally provide this`
	`  password to the CA Administrator in order to revoke your certificate.`
	`  For security reasons your password will not be saved in the configuration.`
	`  Please make a note of it.`
	`  Password:nbv123`
	` The subject name in the certificate will be: DeviceA.Inspur.com`
	` Include the switch serial number in the subject name? [yes/no]: no`
	` Include an IP address in the subject name [yes/no]: yes`
	`ip address:172.22.31.162`
	` The certificate request will be displayed...`
		Generates a certificate request for an authenticated CA.
		Note You must remember the challenge password. It is not saved with the configuration. You must enter this password if your certificate needs to be revoked.

	Command or Action	Purpose
	-----BEGIN CERTIFICATE REQUEST----- MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUJ1DVaSMqNIgJ2kt8rl4lKY 0JC6ManNy4qxk8VeMXZSiLJ4JgTzKWdxbLDkTTysnjuCXGvjb+wj0hEhv/y51T9y P2NJJ8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rargZvHtGJ91XTq4WoVkSCzXv8S VqyH0vEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDYGCSqGSIb3DQEJ DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ KoZIhvcNAQEEBQADgYEAkT60KER6Qo8nj0sDXZVHSfJZh6K6JtDz3Gkd99GlFWgt PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6Ul88nTOjglXMjja8 8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPNTZacJCUS6ZqKCMetbKytUx0= -----END CERTIFICATE REQUEST-----	

Step 3	**exit**
	Example:
	`switch(config-trustpoint)# exit`
	`switch(config)#`
		Exits trust point configuration mode.

Step 4	(Optional) **show crypto ca certificates** **Example:** `switch(config)# show crypto ca certificates`	Displays the CA certificates.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Creating a Trust Point CA Association.

7.7.7 Installing Identity Certificates

You can receive the identity certificate from the CA by e-mail or through a web browser in base64 encoded text form. You must install the identity certificate from the CA by cutting and pasting the encoded text.

Before you begin

Create an association with the CA.
Obtain the CA certificate or CA certificate chain.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca import** *name* **certificate**
3. **exit**
4. (Optional) **show crypto ca certificates**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**crypto ca import** *name* **certificate**
	
	Example:
	`switch(config)# crypto ca import admin-ca certificate`
	`input (cut & paste) certificate in PEM format:`
	`-----BEGIN CERTIFICATE-----`
	`MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G`
	`CSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD`
	`VQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UEChMFQ2lz`
	`Y28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBDQTAeFw0w`
	`NTExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEu`
	`Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C`
	`dQ1WkjKjSICdpLfK5eJSmNCQujGpzcuKsZPFXjF2UoiyeCYE8ylncWyw5E08rJ47`
	`glxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWYAu8rDfz8jMCnIW1aY/q2q4Gb`
	`x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw`
	`GYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR`
	`bhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjTEVNjskYUBoLFmxxoYGW`
	`pIGTMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE`
	`BhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w`
	`DAYDVQQKEwVDaXNjbzETMBEGA1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBh`
	`cm5hIENBghAFYNKJrLQZlE9JEiWMrRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6`
	`Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5jcmwwMKAuoCyGKmZpbGU6`
	`Ly9cXHNzZS0wOFxDZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH`
	`AQEEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTA4L0NlcnRFbnJvbGwvc3Nl`
	`LTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4`
	`XENlcnRFbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF`
	`AANBADbGBGsbe7GNLh9xeOTWBNb4U69ZSuDDcOcUZUUTgrpnTqVpPyejtsyflw`
	`E36cIZu4WsExREqxbTk8ycx7V5o=`
	`-----END CERTIFICATE-----`
		Prompts you to cut and paste the identity certificate for the CA named admin-ca.
		
		The maximum number of identify certificates that you can configure on a device is 16.

Step 3	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show crypto ca certificates** **Example:** `switch# show crypto ca certificates`	Displays the CA certificates.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Creating a Trust Point CA Association.

7.7.8 Ensuring Trust Point Configurations Persist Across Reboots

You can ensure that the trustpoint configuration persists across Inspur INOS device reboots.

The trust point configuration is a normal Inspur INOS device configuration that persists across system reboots only if you copy it explicitly to the startup configuration. The certificates, key pairs, and CRL associated with a trust point are automatically persistent if you have already copied the trust point configuration in the startup configuration. Conversely, if the trust point configuration is not copied to the startup configuration, the certificates, key pairs, and CRL associated with it are not persistent since they require the corresponding trust point configuration after a reboot. Always copy the running configuration to the startup configuration to ensure that the configured certificates, key pairs, and CRLs are persistent. Also, save the running configuration after deleting a certificate or key pair to ensure that the deletions permanent.

The certificates and CRL associated with a trust point automatically become persistent when imported (that is, without explicitly copying to the startup configuration) if the specific trust point is already saved in startup configuration.

We recommend that you create a password-protected backup of the identity certificates and save it to an external server.

Related Topics

Exporting Identity Information in PKCS 12 Format.

7.7.9 Exporting Identity Information in PKCS 12 Format

You can export the identity certificate along with the RSA key pair and CA certificate (or the entire chain in the case of a subordinate CA) of a trust point to a PKCS#12 file for backup purposes. You can import the certificate and RSA key pair to recover from a system crash on your device or when you replace the supervisor modules.

Before you begin

Authenticate the CA.
Install an identity certificate.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca export** *name* **pkcs12 bootflash:***filename password*
3. **exit**
4. **copy booflash:***filename scheme***://***server***/** [*url* **/**]*filename*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca export** *name* **pkcs12 bootflash:***filename password* **Example:** `switch(config)# crypto ca export admin-ca pkcs12 bootflash:adminid.p12 nbv123`	Exports the identity certificate and associated key pair and CA certificates for a trust point CA. The password is alphanumeric, case sensitive, and has a maximum length of 128 characters.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	**copy booflash:***filename scheme***://***server***/** [*url* **/**]*filename* **Example:** `switch# copy bootflash:adminid.p12 tftp:adminid.p12`	Copies the PKCS#12 format file to a remote server. For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case

	Command or Action	Purpose
		sensitive.

Related Topics

Generating an RSA Key Pair.
Authenticating the CA.
Installing Identity Certificates.

7.7.10 Importing Identity Information in PKCS 12 Format

You can import the certificate and RSA key pair to recover from a system crash on your device or when you replace the supervisor modules.

Before you begin

Ensure that the trust point is empty by checking that no RSA key pair is associated with it and no CA is associated with the trust point using CA authentication.

SUMMARY STEPS

1. **copy** *scheme*://*server*/[*url* /]*filename* **bootflash:***filename*
2. **configure terminal**
3. **crypto ca import** *name* **pksc12 bootflash:***filename*
4. **exit**
5. (Optional) **show crypto ca certificates**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**copy** *scheme*:// *server*/[*url* /]*filename* **bootflash:***filename* **Example:** `switch# copy tftp:adminid.p12 bootflash:adminid.p12`	Copies the PKCS#12 format file from the remote server. For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case sensitive.

Step 2	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 3	**crypto ca import** *name* **pksc12 bootflash:***filename* **Example:** `switch(config)# crypto ca import admin-ca pkcs12 bootflash:adminid.p12 nbv123`	Imports the identity certificate and associated key pair and CA certificates for trust point CA.
Step 4	**exit** **Example:**	Exits configuration mode.

	Command or Action	Purpose
	`switch(config)# exit` `switch#`	
Step 5	(Optional) **show crypto ca certificates** **Example:** `switch# show crypto ca certificates`	Displays the CA certificates.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

7.7.11 Configuring a CRL

You can manually configure CRLs that you have downloaded from the trust points. The Inspur INOS software caches the CRLs in the device bootflash (cert-store). During the verification of a peer certificate, the Inspur INOS software checks the CRL from the issuing CA only if you have downloaded the CRL to the device and you have configured certificate revocation checking to use the CRL.

Before you begin

Ensure that you have enabled certificate revocation checking.

SUMMARY STEPS

1. **copy** *scheme*:[//*server*/[*url* /]]*filename* **bootflash:***filename*
2. **configure terminal**
3. **crypto ca crl request** *name* **bootflash:***filename*
4. **exit**
5. (Optional) **show crypto ca crl** *name*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**copy** *scheme*:[//*server*/[*url* /]]*filename* **bootflash:***filename*
	Example:
	`switch# copy tftp:adminca.crl`
	`bootflash:adminca.crl`
		Downloads the CRL from a remote server.
		For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.
		The *server*, *url*, and *filename* arguments are case sensitive.

Step 2	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 3	**crypto ca crl request** *name* **bootflash:***filename*
	Example:
	`switch(config)# crypto ca crl request admin-ca`
	`bootflash:adminca.crl`
		Configures or replaces the current CRL with the one specified in the file.

	Command or Action	Purpose

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show crypto ca crl** *name* **Example:** `switch# show crypto ca crl admin-ca`	Displays the CA CRL information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

7.7.12 Deleting Certificates from the CA Configuration

You can delete the identity certificates and CA certificates that are configured in a trust point. You must first delete the identity certificate, followed by the CA certificates. After deleting the identity certificate, you can disassociate the RSA key pair from a trust point. You must delete certificates to remove expired or revoked certificates, certificates that have compromised (or suspected to be compromised) key pairs, or CAs that are no longer trusted.

SUMMARY STEPS

1. **configure terminal**
2. **crypto ca trustpoint** *name*
3. **delete ca-certificate**
4. **delete certificate** [**force**]
5. **exit**
6. (Optional) **show crypto ca certificates** [*name*]
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto ca trustpoint** *name*
	Example:
	`switch(config)# crypto ca trustpoint admin-ca`
	`switch(config-trustpoint)#`
		Specifies a trust point CA and enters trust point configuration mode.

Step 3	**delete ca-certificate** **Example:** `switch(config-trustpoint)# delete ca-certificate`	Deletes the CA certificate or certificate chain.
Step 4	**delete certificate** [**force**] **Example:**	Deletes the identity certificate. You must use the **force** option if the identity

	Command or Action	Purpose
	`switch(config-trustpoint)# delete certificate`	certificate you want to delete is the last certificate in a certificate chain or only identity certificate in the device. This requirement ensures that you do not mistakenly delete the last certificate in a certificate chain or only the identity certificate and leave the applications (such as SSH) without a certificate to use.

Step 5	**exit**
	Example:
	`switch(config-trustpoint)# exit`
	`switch(config)#`
		Exits trust point configuration mode.

Step 6	(Optional) **show crypto ca certificates** [*name*] **Example:** `switch(config)# show crypto ca certificates admin-ca`	Displays the CA certificate information.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

7.7.13 Deleting RSA Key Pairs from a Inspur INOS Device

You can delete the RSA key pairs from a Inspur INOS device if you believe the RSA key pairs were compromised in some way and should no longer be used.

SUMMARY STEPS

1. **configure terminal**
2. **crypto key zeroize rsa** *label*
3. **exit**
4. (Optional) **show crypto key mypubkey rsa**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**crypto key zeroize rsa** *label* **Example:** `switch(config)# crypto key zeroize rsa MyKey`	Deletes the RSA key pair.
Step 3	**exit** **Example:** `switch(config)# exit`	Exits configuration mode.

	Command or Action	Purpose
	`switch#`	
Step 4	(Optional) **show crypto key mypubkey rsa** **Example:** `switch# show crypto key mypubkey rsa`	Displays the RSA key pair configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Generating Certificate Requests.

7.8 Verifying the PKI Configuration

To display PKI configuration information, perform one of the following tasks:

Command	Purpose
show crypto key mypubkey rsa	Displays information about the RSA public keys generated on the Inspur INOS device.
show crypto ca certificates	Displays information about CA and identity certificates.
show crypto ca crl	Displays information about CA CRLs.
show crypto ca trustpoints	Displays information about CA trust points.

7.9 Configuration Examples for PKI

This section shows examples of the tasks that you can use to configure certificates and CRLs on Inspur INOS devices using a Microsoft Windows Certificate server.

7.9.1 Configuring Certificates on a Inspur INOS Device

To configure certificates on a Inspur INOS device, follow these steps:

Step 1 Configure the device FQDN.

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# hostname Device-1
Device-1(config)#
```

Step 2 Configure the DNS domain name for the device.

```
Device-1(config)# ip domain-name Inspur.com
```

Step 3 Create a trust point.

```
Device-1(config)# crypto ca trustpoint myCA
Device-1(config-trustpoint)# exit
Device-1(config)# show crypto ca trustpoints
trustpoint:   myCA;   key:
revokation methods:  crl
```

Step 4 Create an RSA key pair for the device.

```
Device-1(config)# crypto key generate rsa label myKey exportable modulus 1024
Device-1(config)# show crypto key mypubkey rsa
key  label:  myKey
key   size:   1024
exportable: yes
```

Step 5 Associate the RSA key pair to the trust point.

```
Device-1(config)#  crypto  ca  trustpoint  myCA
Device-1(config-trustpoint)#  rsakeypair  myKey
Device-1(config-trustpoint)# exit
Device-1(config)# show crypto ca trustpoints
trustpoint:  myCA;  key:  myKey
revokation methods:  crl
```

Step 6 Download the CA certificate from the Microsoft Certificate Service web interface.

Step 7 Authenticate the CA that you want to enroll to the trust point.

```
Device-1(config)# crypto ca authenticate myCA
input (cut & paste) CA certificate (chain) in PEM format;
end the input with a line containing only END OF INPUT :
-----BEGIN CERTIFICATE-----
MIIC4jCCAoygAwIBAgIQBWDSiay0GZRPSRIljK0ZejANBgkqhkiG9w0BAQUFADCB
kDEgMB4GCSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklO
MRIwEAYDVQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UE
ChMFQ2lzY28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBD
QTAeFw0wNTA1MDMyMjQ2MzdaFw0wNzA1MDMyMjU1MTdaMIGQMSAwHgYJKoZIhvcN
AQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UEBhMCSU4xEjAQBgNVBAgTCUth
cm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4wDAYDVQQKEwVDaXNjbzETMBEG
A1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBhcm5hIENBMFwwDQYJKoZIhvcN
AQEBBQADSwAwSAJBAMW/7b3+DXJPANBsIHHzluNccNM87ypyzwuoSNZXOMpeRXXI
OzyBAgiXT2ASFuUOwQ1iDM8rO/41jf8RxvYKvysCAwEAAaOBvzCBvDALBgNVHQ8E
BAMCAcYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJyjyRoMbrCNMRU2OyRhQ
GgsWbHEwawYDVR0fBGQwYjAuoCygKoYoaHR0cDovL3NzZS0wOC9DZXJ0RW5yb2xs
L0FwYXJuYSUyMENBLmNybDAwoC6gLIYqZmlsZTovL1xcc3NlLTA4XENlcnRFbnJv
bGxcQXBhcm5hJTIwQ0EuY3JsMBAGCSsGAQQBgjcVAQQDAgEAMA0GCSqGSIb3DQEB
BQUAA0EAHv6UQ+8nE399Tww+KaGr0g0NIJaqNgLh0AFcT0rEyuyt/WYGPzksF9Ea
NBG7E0oN66zex0EOEfG1Vs6mXp1//w==
-----END CERTIFICATE-----
END OF INPUT
Fingerprint(s): MD5 Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12
Do you accept this certificate? [yes/no]:y

Device-1(config)# show crypto ca certificates
Trustpoint:  myCA
CA certificate 0:
subject= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/
L=Bangalore/O=Yourcompany/OU=netstorage/CN=Aparna CA
issuer= /emailAddress=admin@yourcompany.com/C=IN/ST=Karnataka/
L=Bangalore/O=Yourcompany/OU=netstorage/CN=Aparna           CA
serial=0560D289ACB419944F4912258CAD197A
```

```
notBefore=May  3 22:46:37 2005 GMT
notAfter=May  3 22:55:17 2007 GMT
MD5 Fingerprint=65:84:9A:27:D5:71:03:33:9C:12:23:92:38:6F:78:12
purposes: sslserver sslclient ike
```

Step 8 Generate a request certificate to use to enroll with a trust point.

```
Device-1(config)# crypto ca enroll myCA
 Create the certificate request ..
 Create a challenge password. You will need to verbally provide this
  password to the CA Administrator in order to revoke your certificate.
  For security reasons your password will not be saved in the configuration.
  Please make a note of it.
  Password: nbv123
 The subject name in the certificate will be: Device-1.Inspur.com
 Include the switch serial number in the subject name? [yes/no]: no
 Include an IP address in the subject name [yes/no]: yes
ip address: 10.10.1.1
 The certificate request will be displayed...
-----BEGIN CERTIFICATE REQUEST-----
MIIBqzCCARQCAQAwHDEaMBgGA1UEAxMRVmVnYXMtMS5jaXNjby5jb20wgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAL8Y1UAJ2NC7jUJ1DVaSMqNIgJ2kt8rl4lKY
0JC6ManNy4qxk8VeMXZSiLJ4JgTzKWdxbLDkTTysnjuCXGvjb+wj0hEhv/y51T9y
P2NJJ8ornqShrvFZgC7ysN/PyMwKcgzhbVpj+rargZvHtGJ91XTq4WoVkSCzXv8S
VqyH0vEvAgMBAAGgTzAVBgkqhkiG9w0BCQcxCBMGbmJ2MTIzMDYGCSqGSIb3DQEJ
DjEpMCcwJQYDVR0RAQH/BBswGYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwDQYJ
KoZIhvcNAQEEBQADgYEAkT60KER6Qo8nj0sDXZVHSfJZh6K6JtDz3Gkd99GlFWgt
PftrNcWUE/pw6HayfQl2T3ecgNwel2d15133YBF2bktExiI6Ul88nTOjglXMjja8
8a23bNDpNsM8rklwA6hWkrVL8NUZEFJxqbjfngPNTZacJCUS6ZqKCMetbKytUx0=
-----END CERTIFICATE REQUEST-----
```

Step 9 Request an identity certificate from the Microsoft Certificate Service web interface.

Step 10 Import the identity certificate.

```
Device-1(config)# crypto ca import myCA certificate
input (cut & paste) certificate in PEM format:
-----BEGIN CERTIFICATE-----
MIIEADCCA6qgAwIBAgIKCjOOoQAAAAAAdDANBgkqhkiG9w0BAQUFADCBkDEgMB4G
CSqGSIb3DQEJARYRYW1hbmRrZUBjaXNjby5jb20xCzAJBgNVBAYTAklOMRIwEAYD
VQQIEwlLYXJuYXRha2ExEjAQBgNVBAcTCUJhbmdhbG9yZTEOMAwGA1UEChMFQ2lz
Y28xEzARBgNVBAsTCm5ldHN0b3JhZ2UxEjAQBgNVBAMTCUFwYXJuYSBDQTAeFw0w
NTExMTIwMzAyNDBaFw0wNjExMTIwMzEyNDBaMBwxGjAYBgNVBAMTEVZlZ2FzLTEu
Y2lzY28uY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC/GNVACdjQu41C
dQ1WkjKjSICdpLfK5eJSmNCQujGpzcuKsZPFXjF2UoiyeCYE8ylncWyw5E08rJ47
glxr42/sI9IRIb/8udU/cj9jSSfKK56koa7xWYAu8rDfz8jMCnIM4W1aY/q2q4Gb
x7RifdV06uFqFZEgs17/Elash9LxLwIDAQABo4ICEzCCAg8wJQYDVR0RAQH/BBsw
GYIRVmVnYXMtMS5jaXNjby5jb22HBKwWH6IwHQYDVR0OBBYEFKCLi+2sspWEfgrR
bhWmlVyo9jngMIHMBgNVHSMEgcQwgcGAFCco8kaDG6wjTEVNjskYUBoLFmxxoYGW
pIGTMIGQMSAwHgYJKoZIhvcNAQkBFhFhbWFuZGtlQGNpc2NvLmNvbTELMAkGA1UE
BhMCSU4xEjAQBgNVBAgTCUthcm5hdGFrYTESMBAGA1UEBxMJQmFuZ2Fsb3JlMQ4w
DAYDVQQKEwVDaXNjbzETMBEGA1UECxMKbmV0c3RvcmFnZTESMBAGA1UEAxMJQXBh
cm5hIENBghAFYNKJrLQZlE9JEiWMrRl6MGsGA1UdHwRkMGIwLqAsoCqGKGh0dHA6
Ly9zc2UtMDgvQ2VydEVucm9sbC9BcGFybmElMjBDQS5jcmwwMKAuoCyGKmZpbGU6
Ly9cXHNzZS0wOFxDZXJ0RW5yb2xsXEFwYXJuYSUyMENBLmNybDCBigYIKwYBBQUH
AQEEfjB8MDsGCCsGAQUFBzAChi9odHRwOi8vc3NlLTA4L0NlcnRFbnJvbGwvc3Nl
LTA4X0FwYXJuYSUyMENBLmNydDA9BggrBgEFBQcwAoYxZmlsZTovL1xcc3NlLTA4
XENlcnRFbnJvbGxcc3NlLTA4X0FwYXJuYSUyMENBLmNydDANBgkqhkiG9w0BAQUF
AANBADbGBGsbe7GNLh9xeOTWBNb4U69ZSuDDcOcUZUUTgrpnTqVpPyejtsyflw
E36cIZu4WsExREqxbTk8ycx7V5o=
-----END CERTIFICATE-----
```

```
Device-1(config)# exit
Device-1#
```

Step 11 Verify the certificate configuration.

Step 12 Save the certificate configuration to the startup configuration.

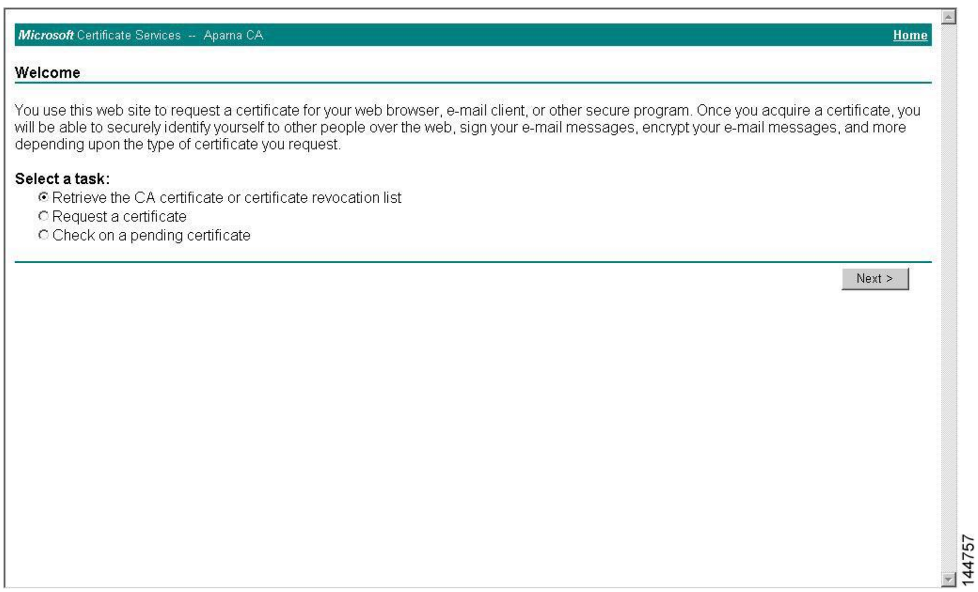
Related Topics

Downloading a CA Certificate.
Requesting an Identity Certificate.

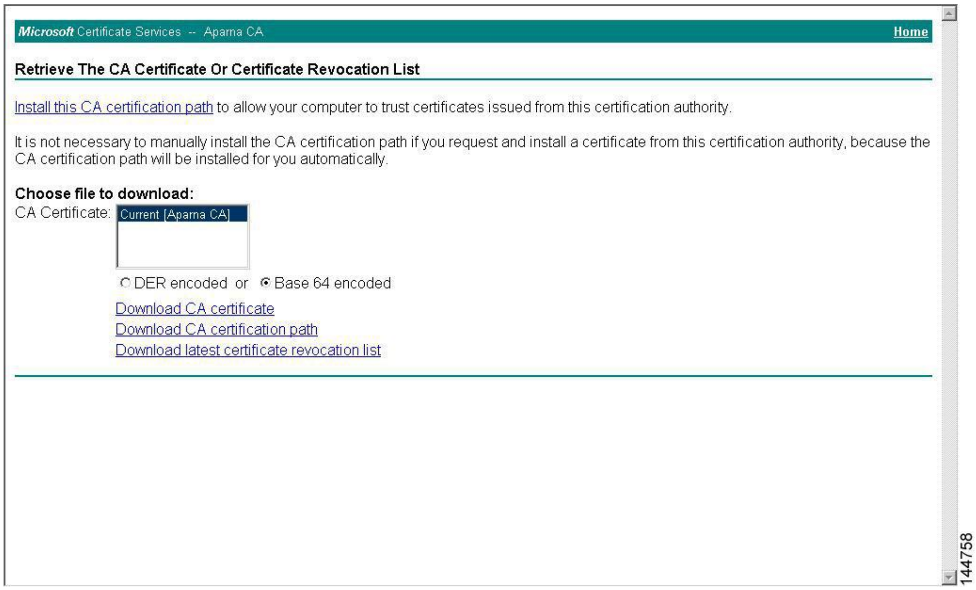
7.9.2 Downloading a CA Certificate

To download a CA certificate from the Microsoft Certificate Services web interface, follow these steps:

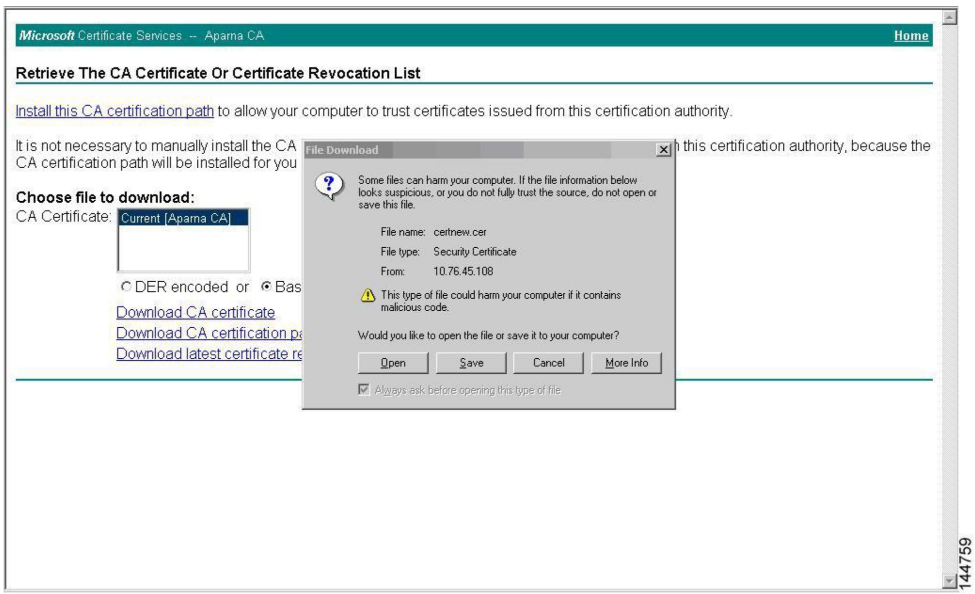
Step 1 From the Microsoft Certificate Services web interface, click **Retrieve the CA certificate or certificate revocation task** and click **Next**.



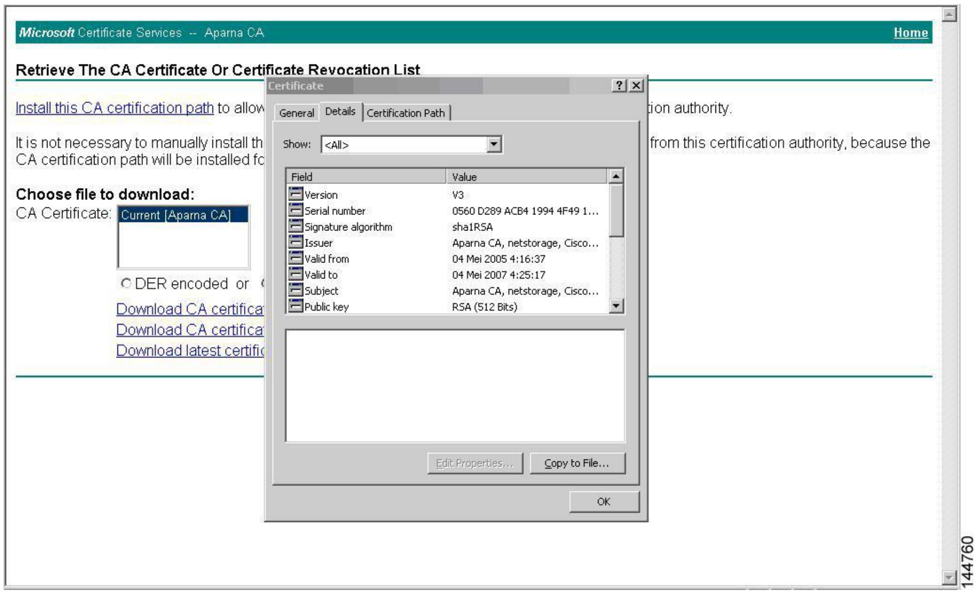
Step 2 From the display list, choose the CA certificate file to download from the displayed list. Then click **Base 64 encoded** and click **Download CA certificate**.



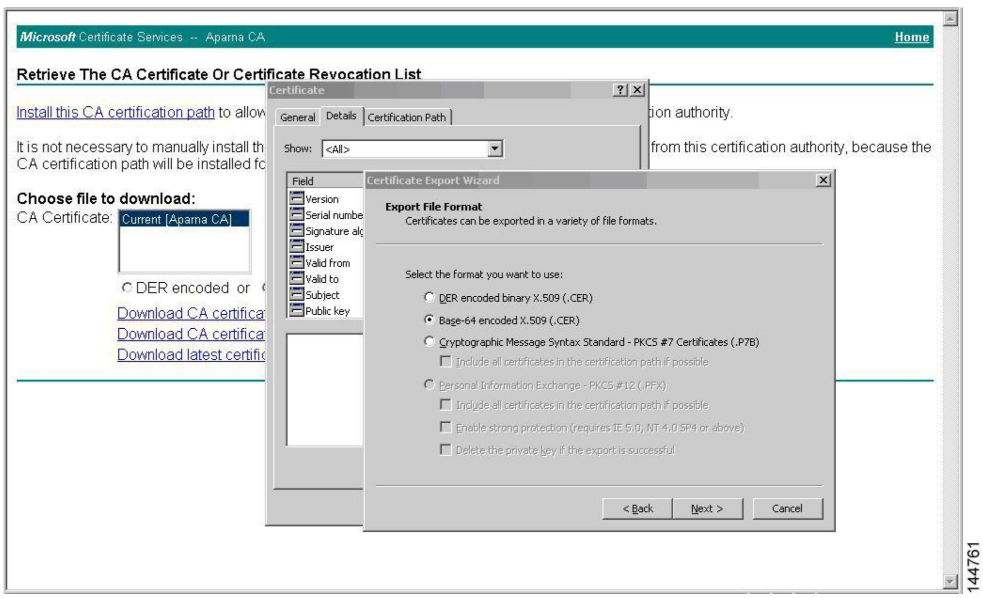
Step 3 Click **Open** in the File Download dialog box.



Step 4 In the Certificate dialog box, click **Copy to File** and click **OK**.



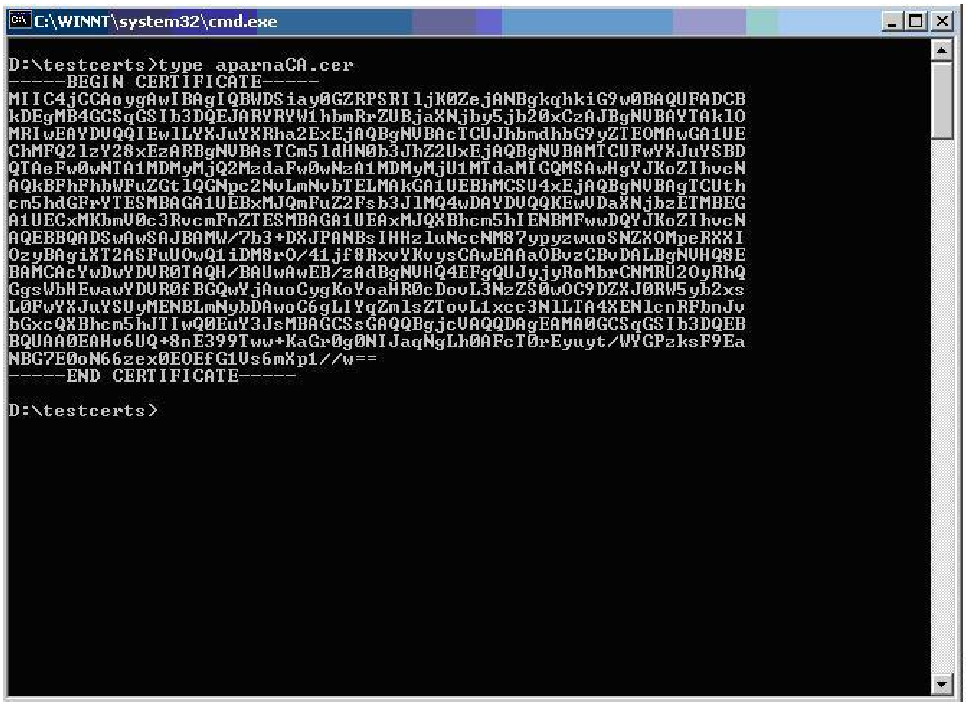
Step 5 From the Certificate Export Wizard dialog box, choose the **Base-64 encoded X.509 (CER)** and click **Next**.



Step 6 In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name and click **Next**.

Step 7 In the Certificate Export Wizard dialog box, click **Finish**.

Step 8 Enter the Microsoft Windows **type** command to display the CA certificate stored in Base-64 (PEM) format.

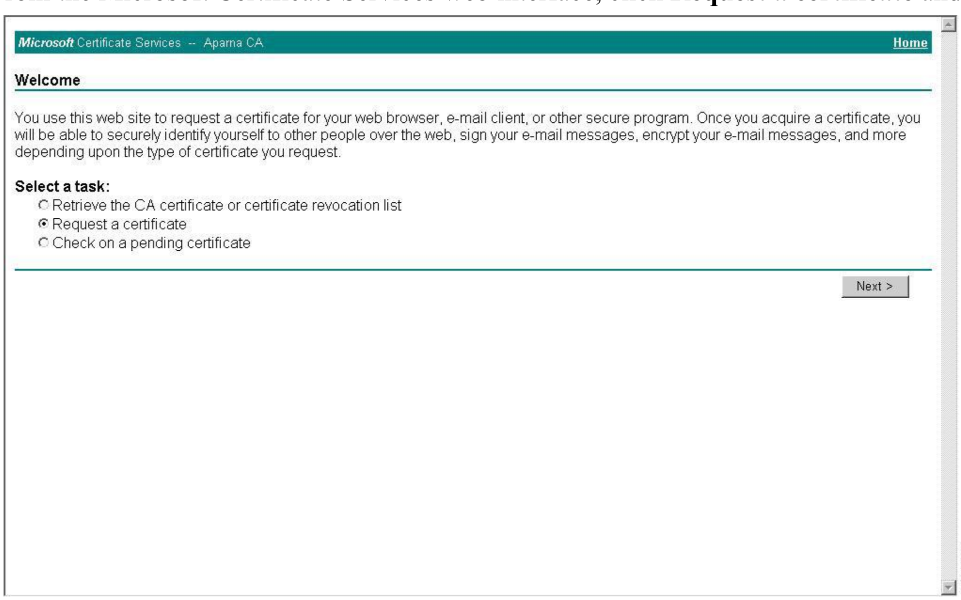


144764

7.9.3 Requesting an Identity Certificate

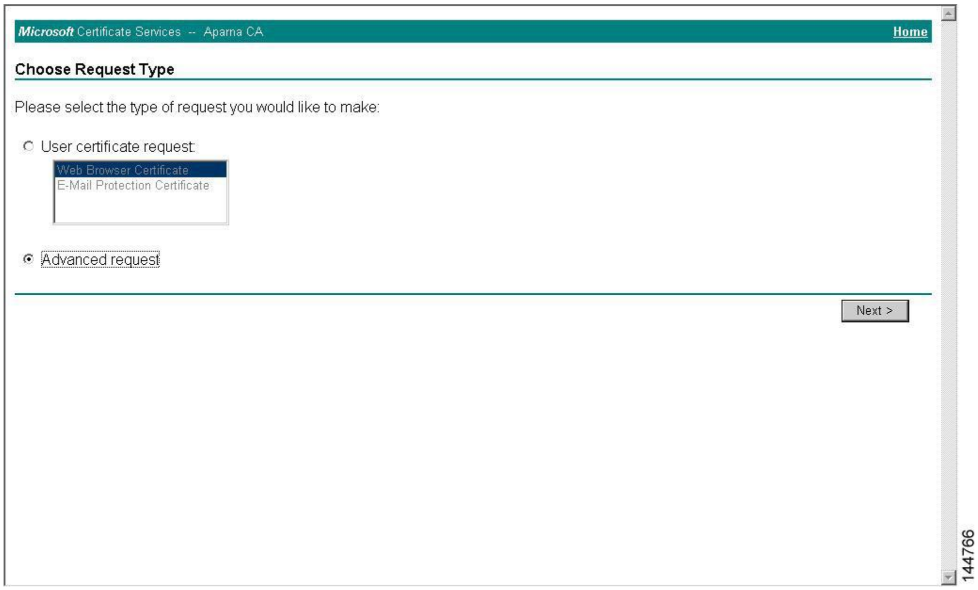
To request an identify certificate from a Microsoft Certificate server using a PKCS#12 certificate signing request (CRS), follow these steps:

Step 1 From the Microsoft Certificate Services web interface, click **Request a certificate** and click **Next**.

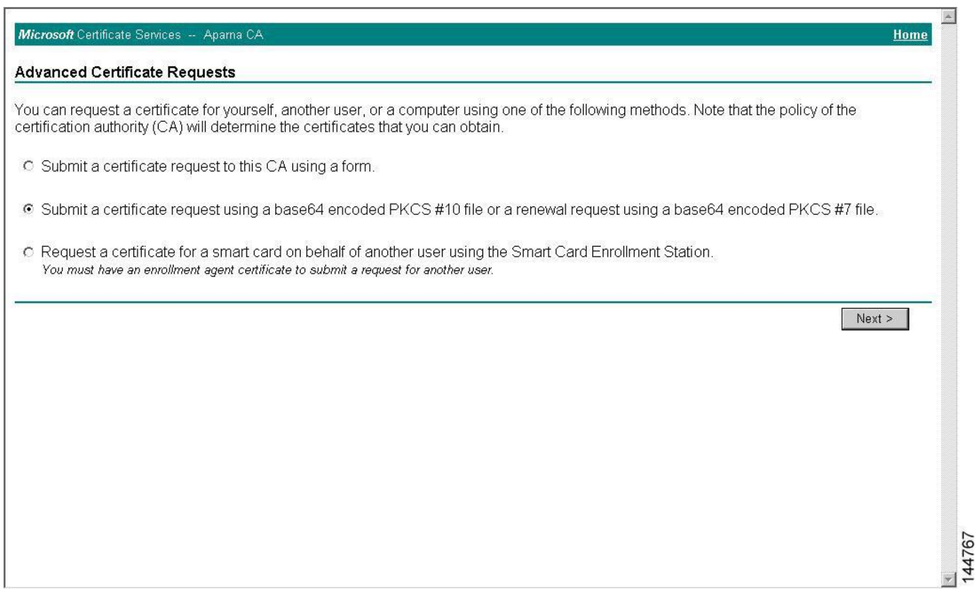


144765

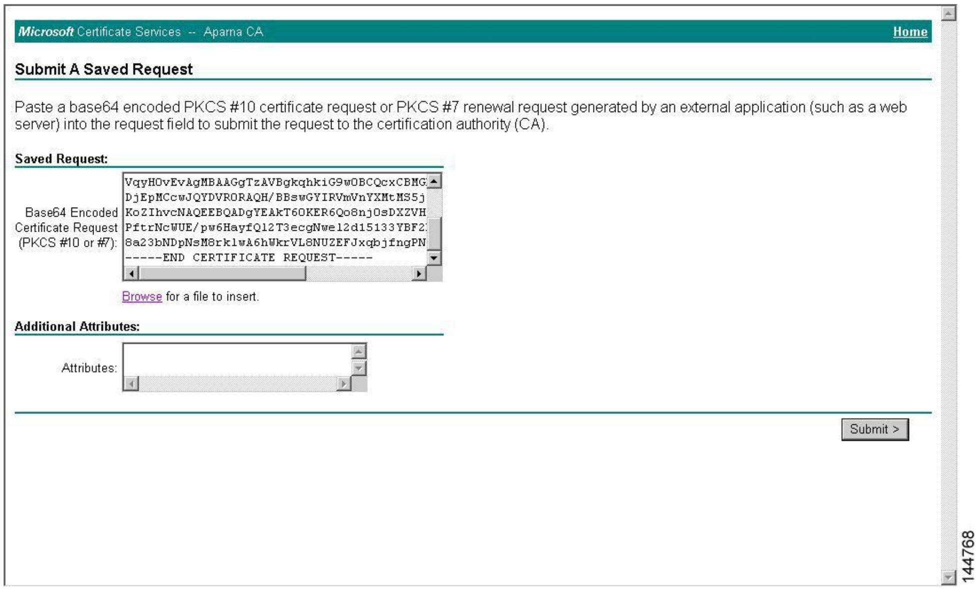
Step 2 Click **Advanced request** and click **Next**.



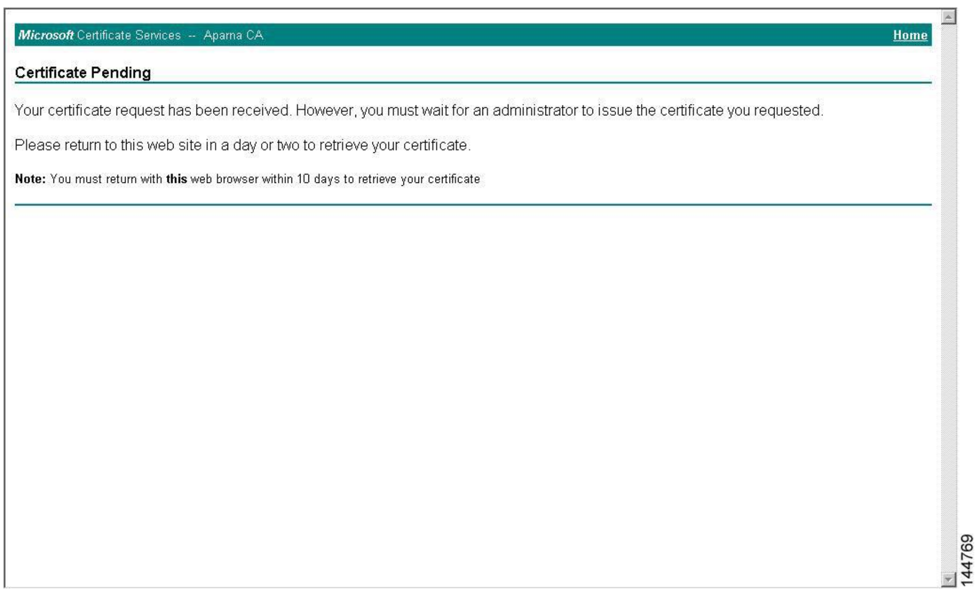
Step 3 Click Submit a certificate request using a base64 encoded PKCS#10 file or a renewal request using a base64 encoded PKCS#7 file and click Next.



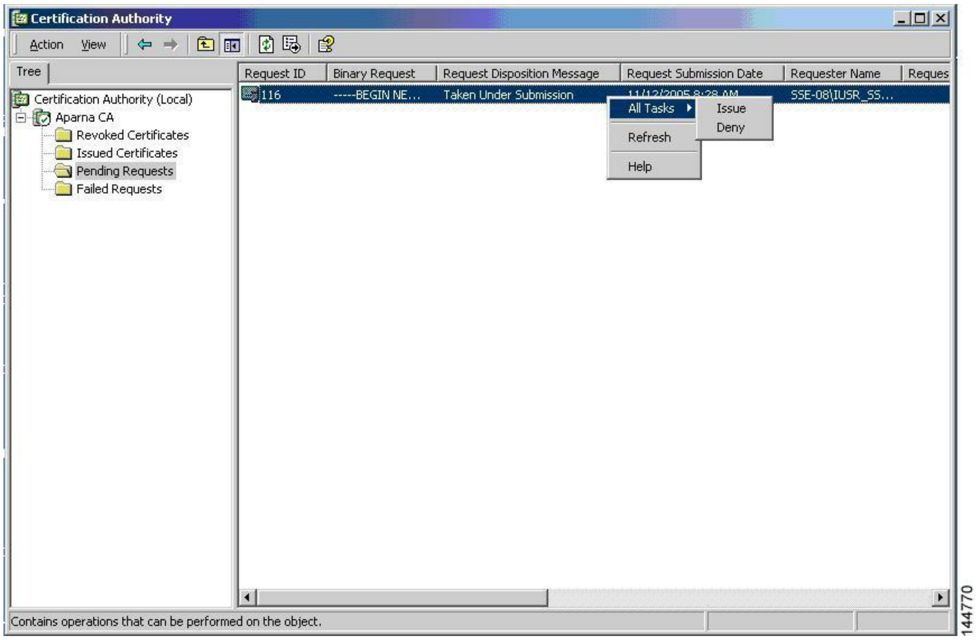
Step 4 In the Saved Request text box, paste the base64 PKCS#10 certificate request and click **Next**. The certificate request is copied from the Inspur INOS device console.



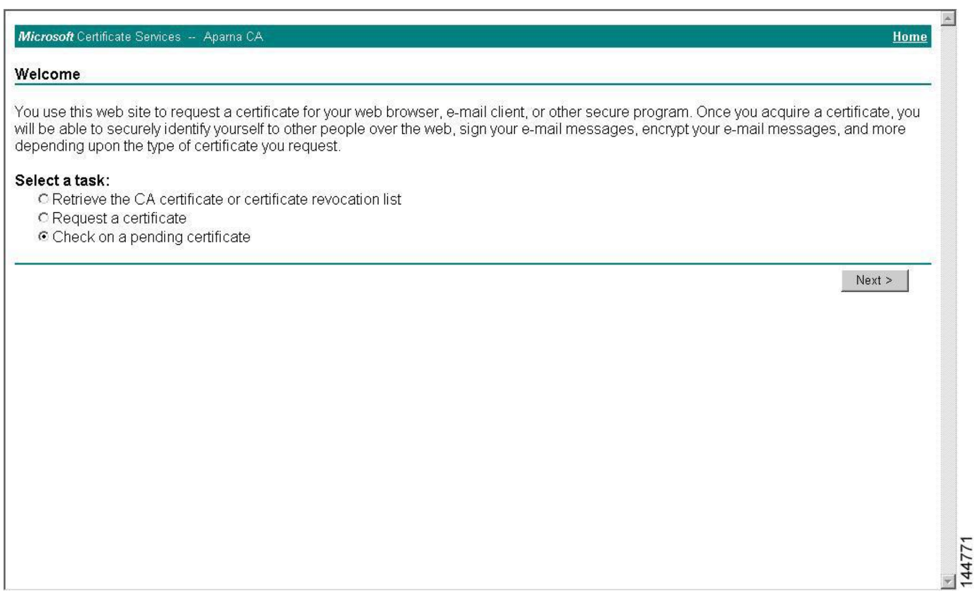
Step 5 Wait one or two days until the certificate is issued by the CA administrator.



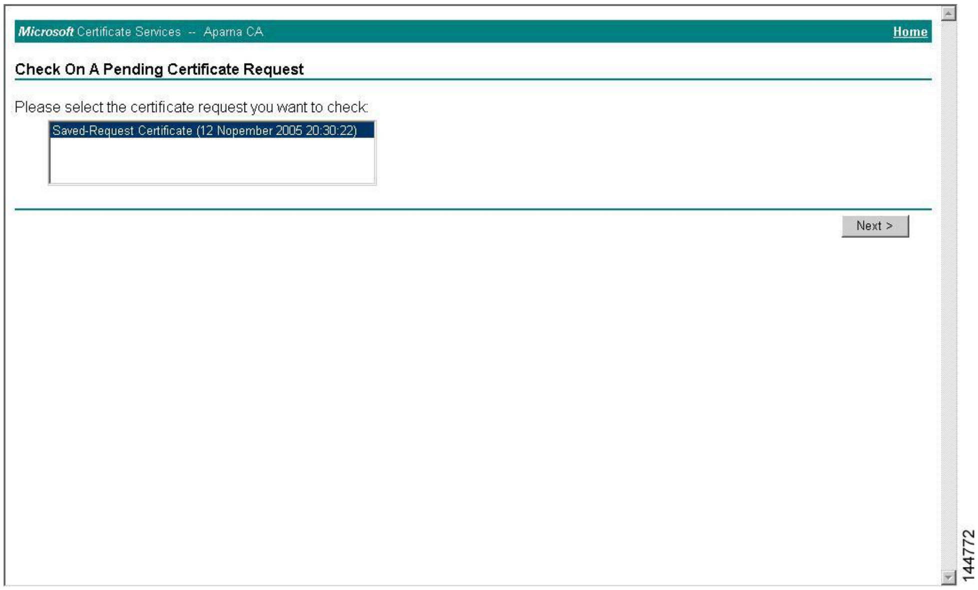
Step 6 Note that the CA administrator approves the certificate request.



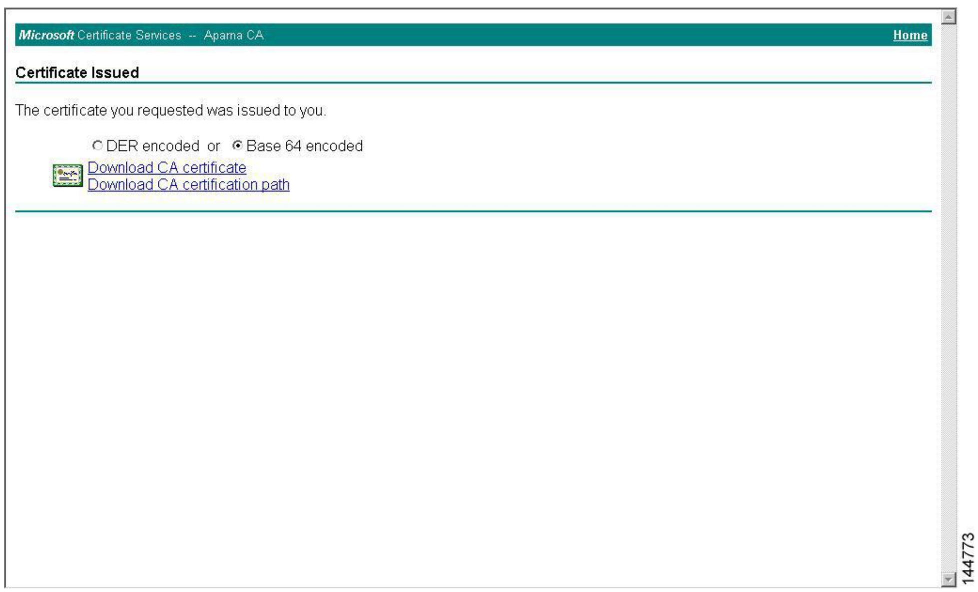
Step 7 From the Microsoft Certificate Services web interface, click **Check on a pending certificate** and click **Next**.



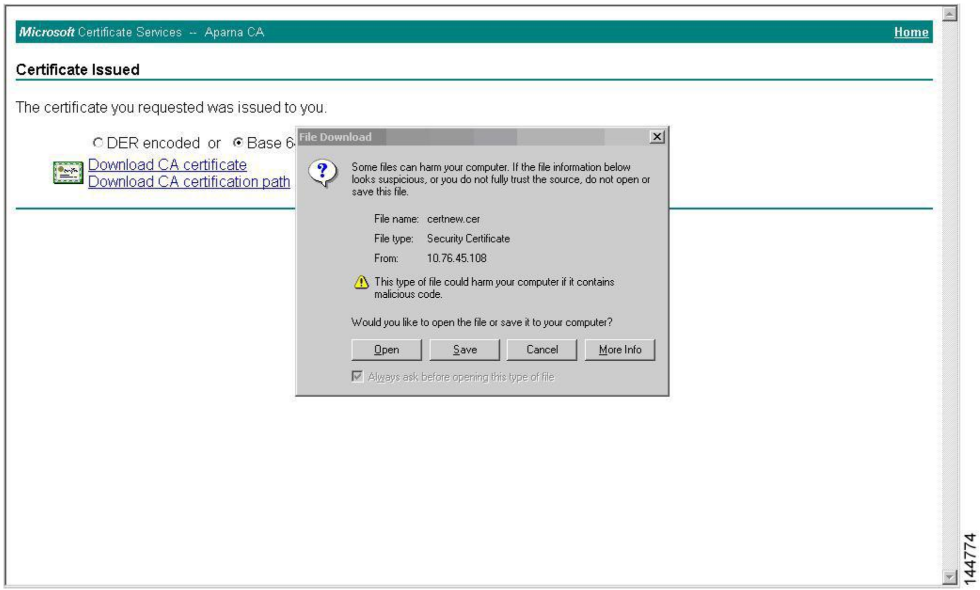
Step 8 Choose the certificate request that you want to check and click **Next**.



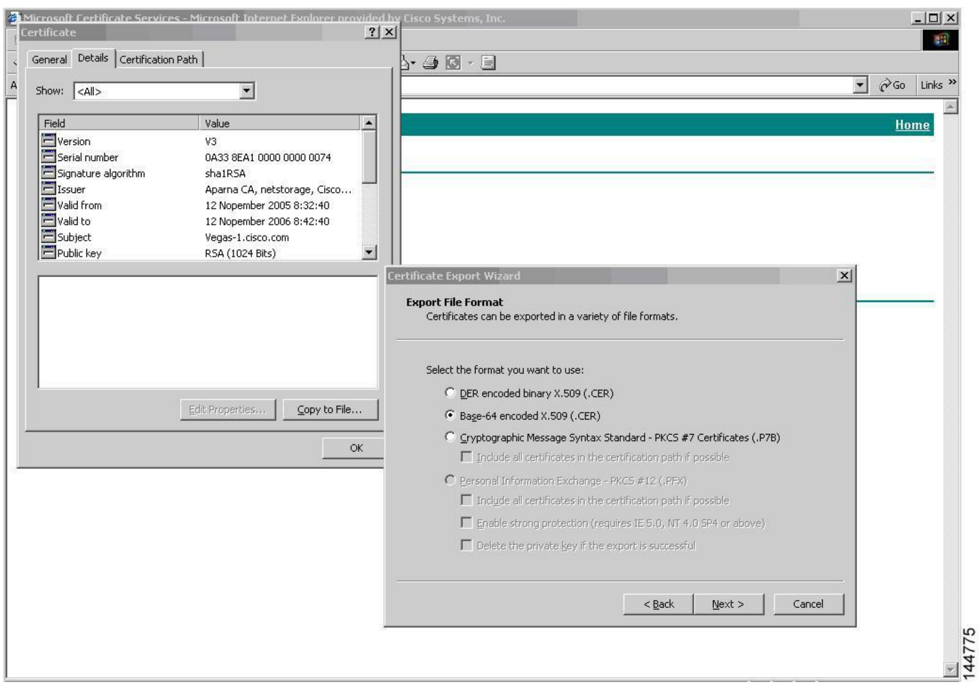
Step 9 Click **Base 64 encoded** and click **Download CA certificate**.



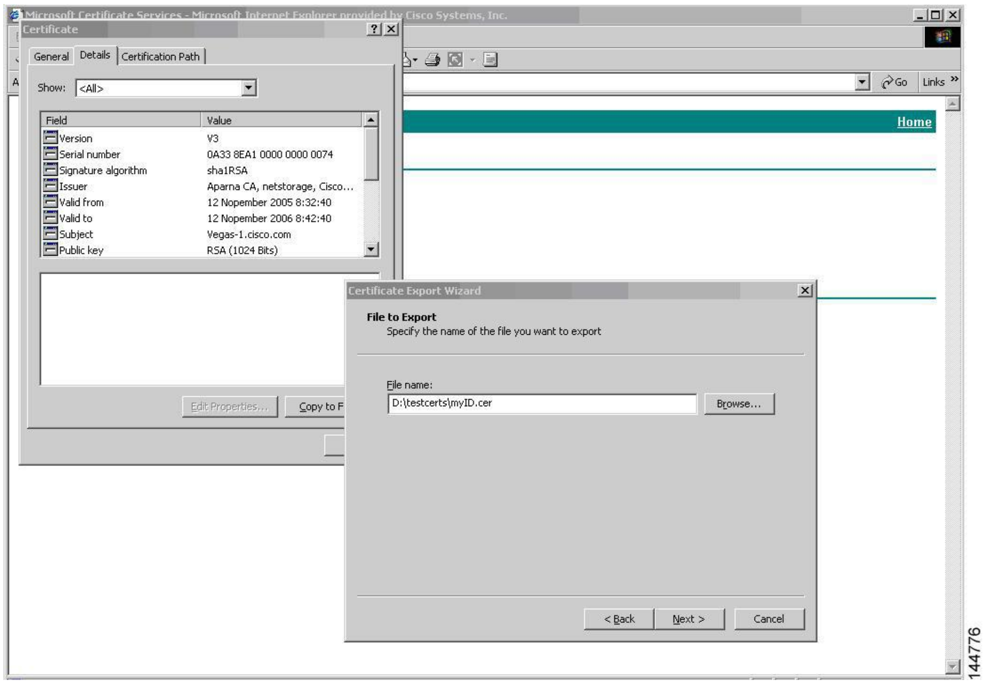
Step 10 In the File Download dialog box, click **Open**.



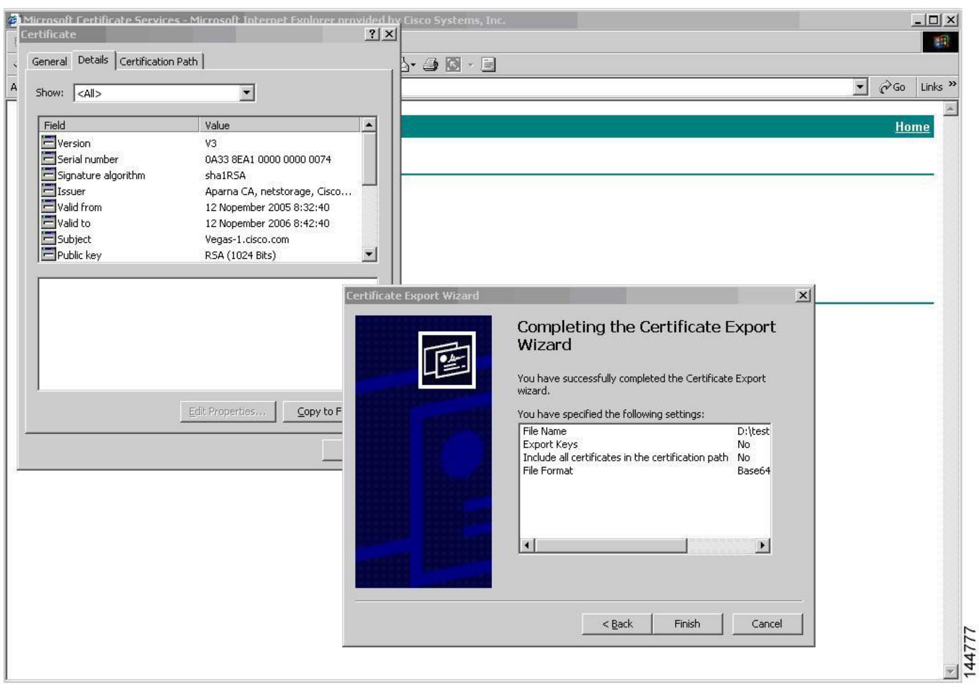
Step 11 In the Certificate box, click **Details** tab and click **Copy to File...**. In the Certificate Export Dialog box, click **Base-64 encoded X.509 (.CER)**, and click **Next**.



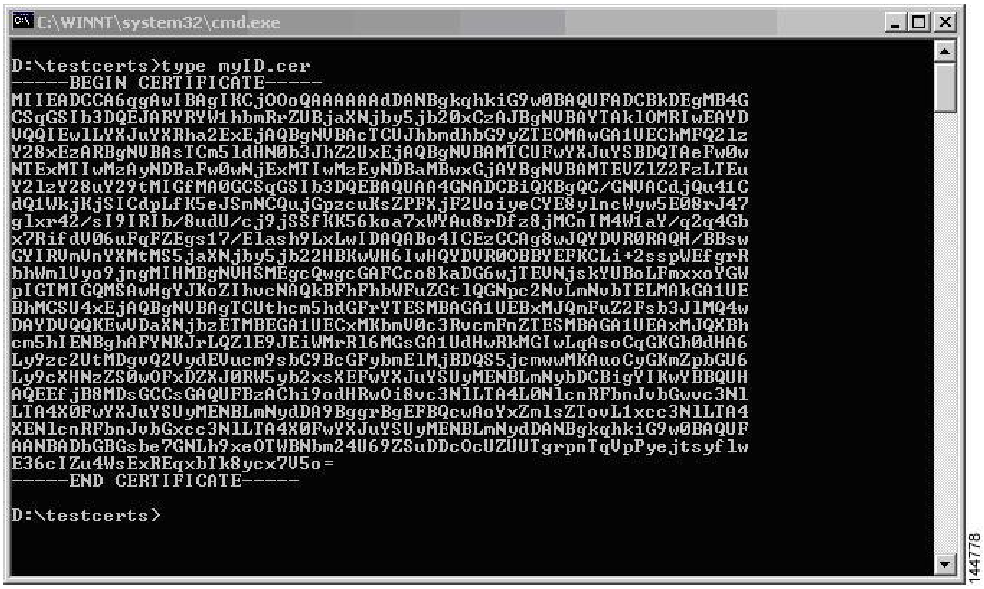
Step 12 In the File name: text box on the Certificate Export Wizard dialog box, enter the destination file name and click **Next**.



Step 13 Click **Finish**.



Step 14 Enter the Microsoft Windows **type** command to display the identity certificate in base64-encoded format.



Related Topics

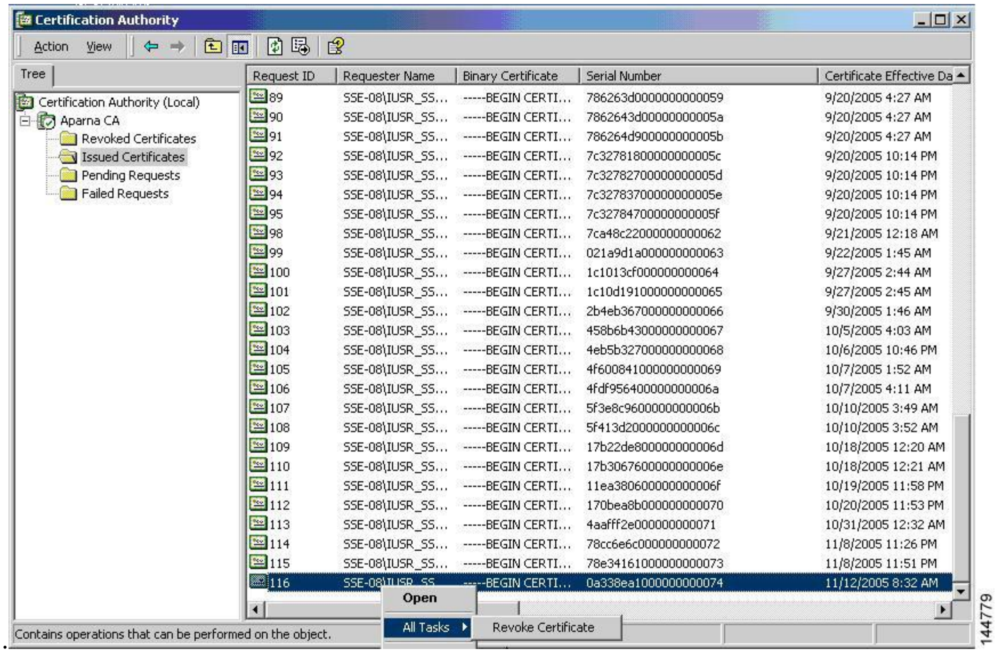
Generating Certificate Requests.
Configuring Certificates on a Inspur INOS Device.

7.9.4 Revoking a Certificate

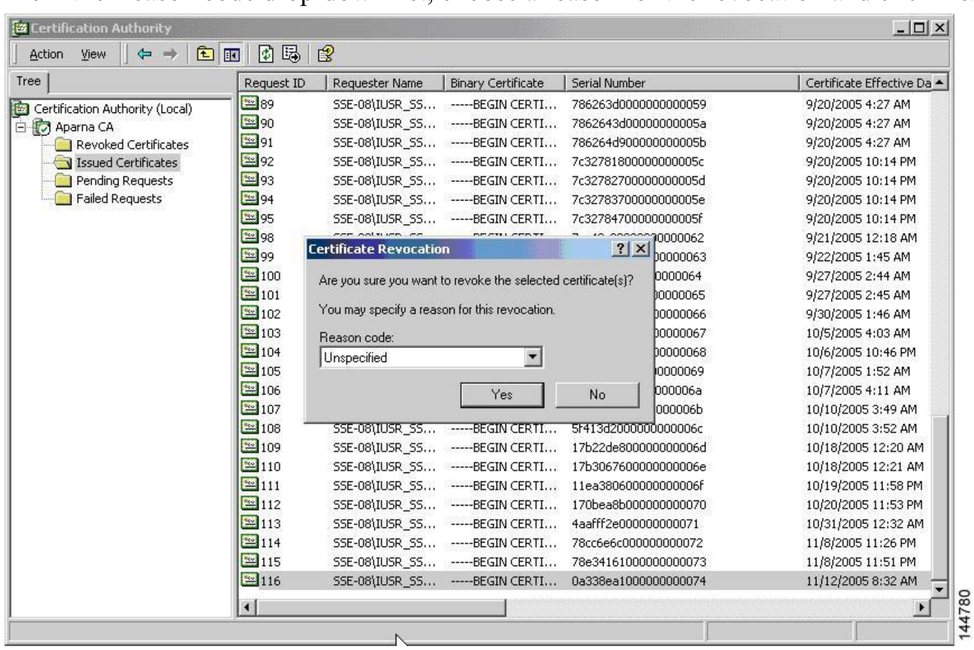
To revoke a certificate using the Microsoft CA administrator program, follow these steps:

Step 1 From the Certification Authority tree, click **Issued Certificates** folder. From the list, right-click the certificate that you want to revoke.

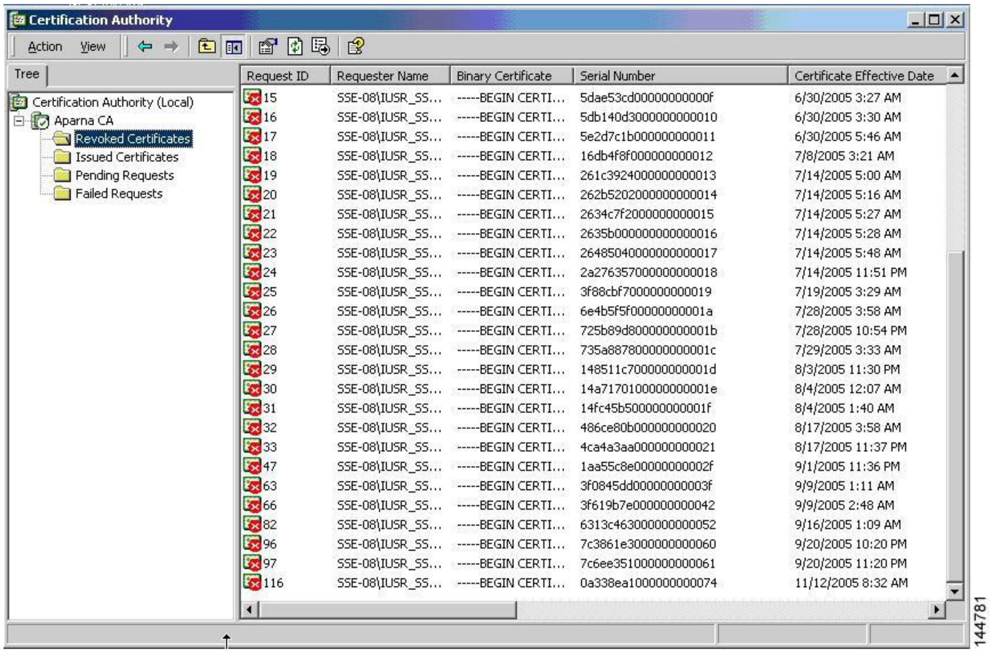
Step 2 Choose **All Tasks > Revoke Certificate**



Step 3 From the Reason code drop-down list, choose a reason for the revocation and click **Yes**.



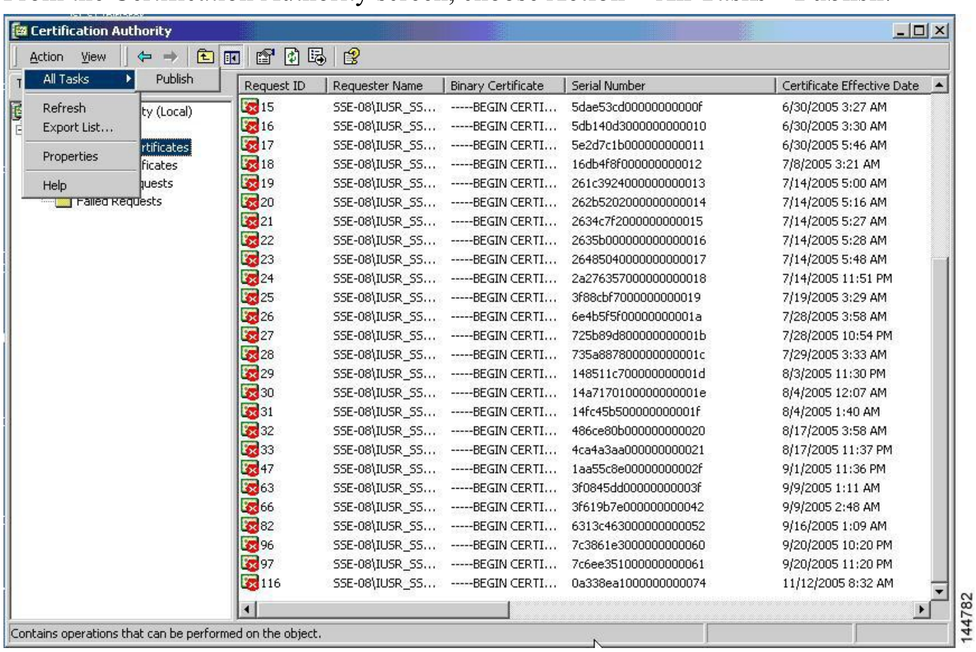
Step 4 Click the **Revoked Certificates** folder to list and verify the certificate revocation.



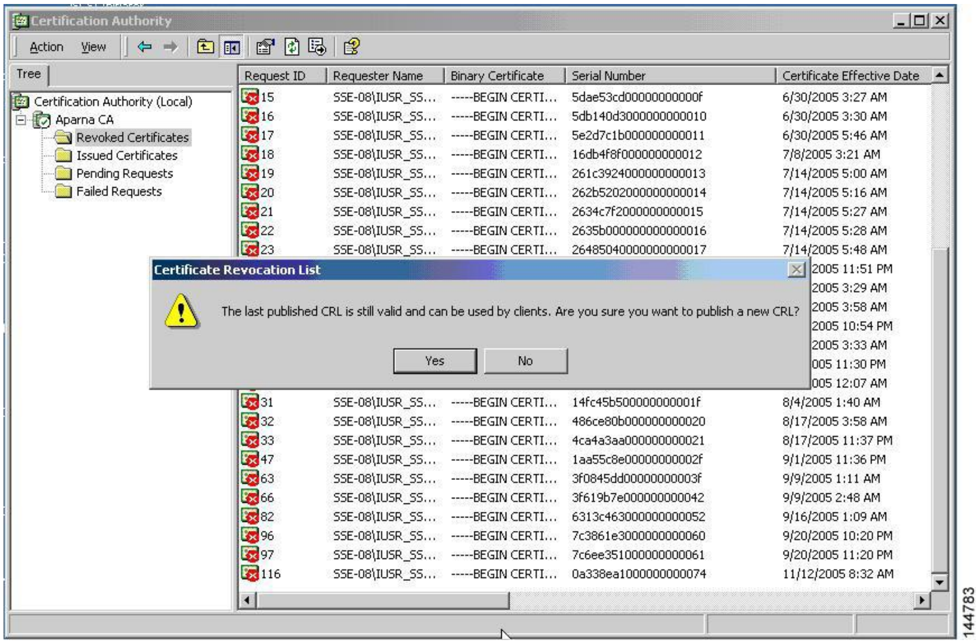
7.9.5 Generating and Publishing the CRL

To generate and publish the CRL using the Microsoft CA administrator program, follow these steps:

Step 1 From the Certification Authority screen, choose **Action > All Tasks > Publish**.



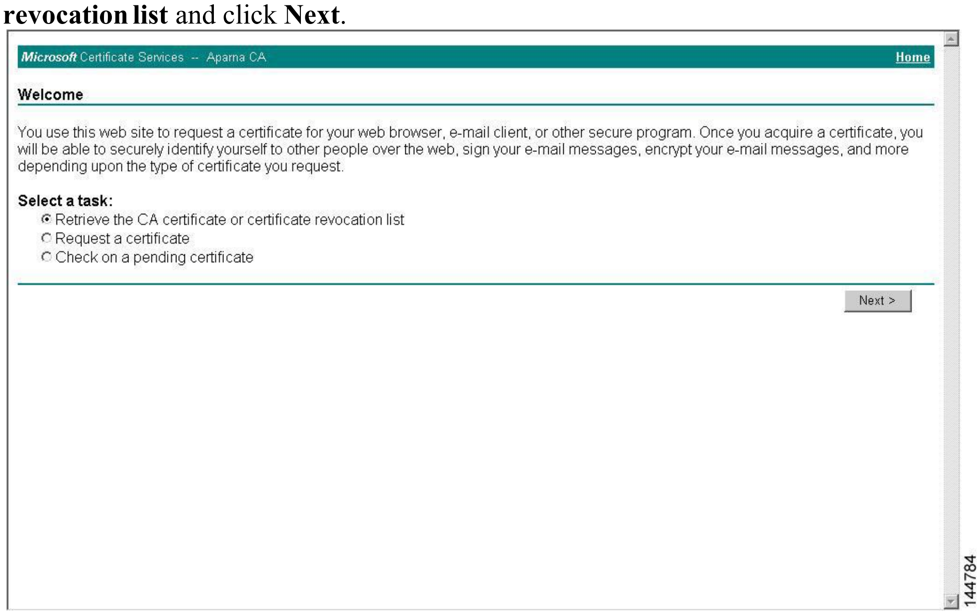
Step 2 In the Certificate Revocation List dialog box, click **Yes** to publish the latest CRL.



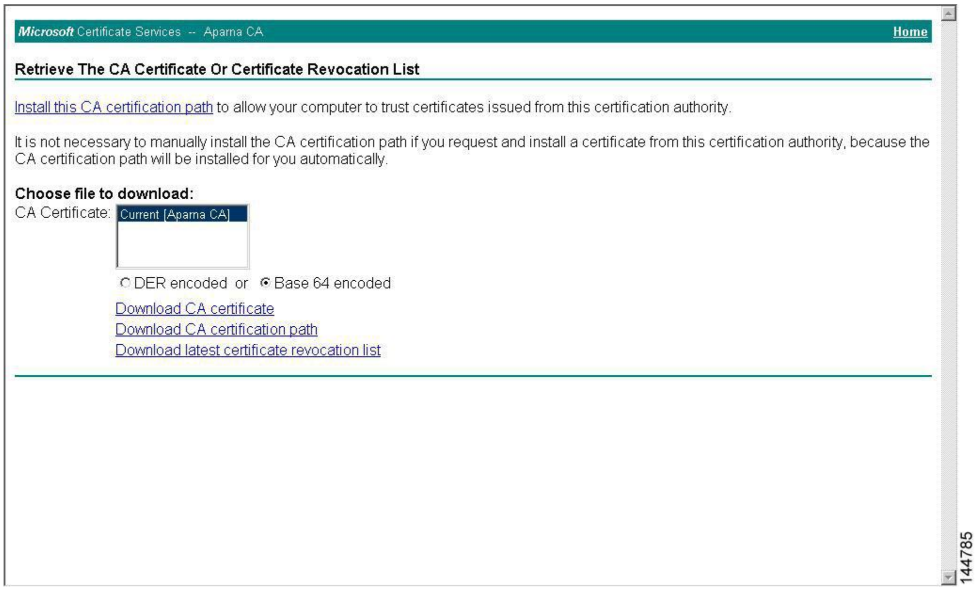
7.9.6 Downloading the CRL

To download the CRL from the Microsoft CA website, follow these steps:

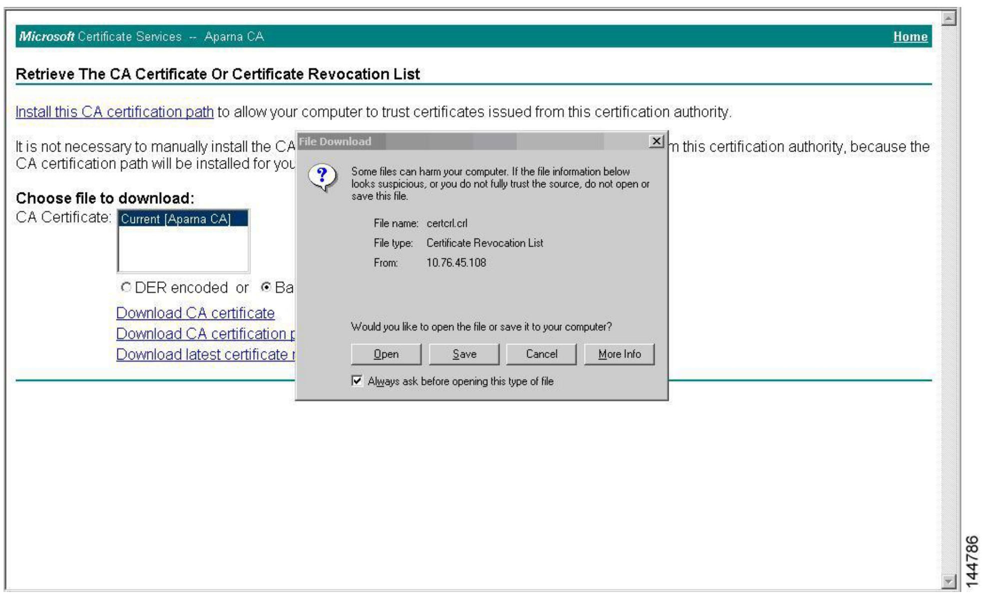
Step 1 From the Microsoft Certificate Services web interface, click **Retrieve the CA certificate or certificate revocation list** and click **Next**.



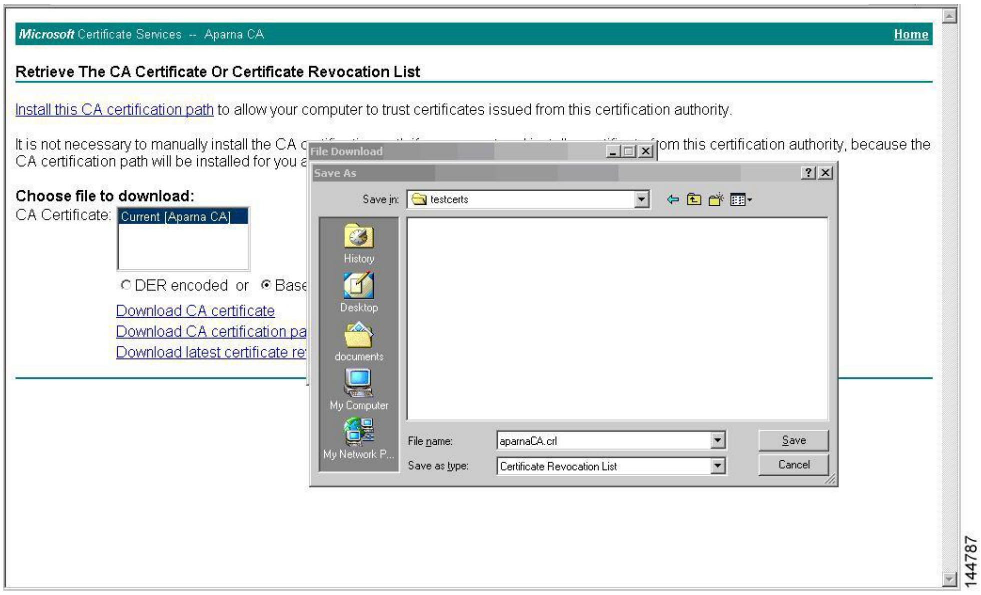
Step 2 Click **Download latest certificate revocation list**.



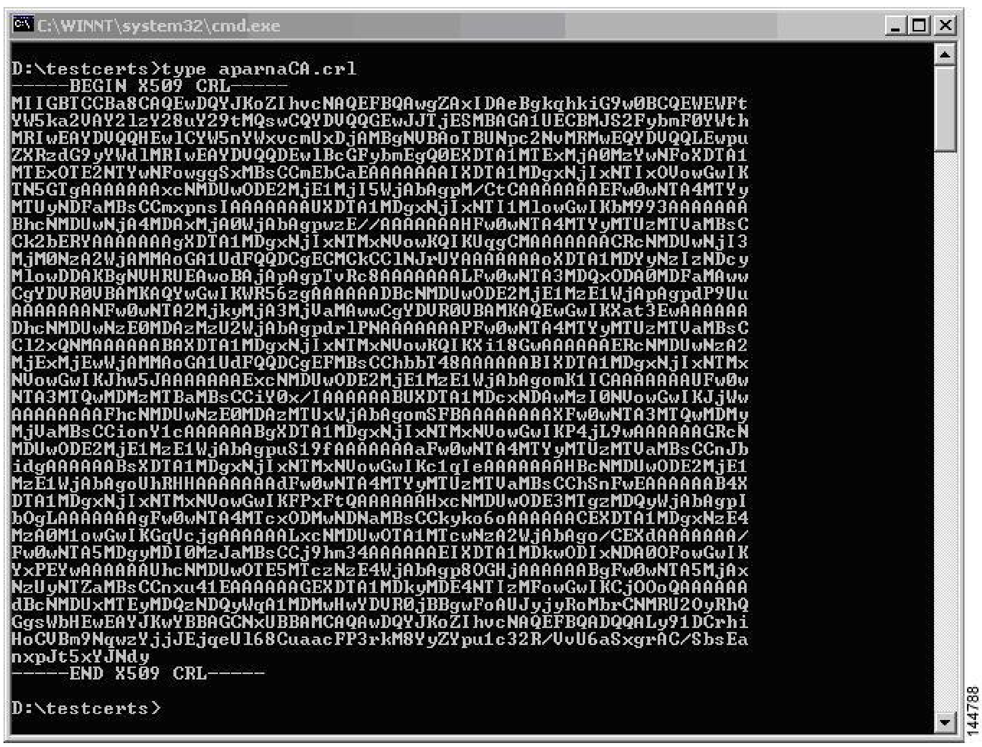
Step 3 In the File Download dialog box, click **Save**.



Step 4 In the Save As dialog box, enter the destination file name and click **Save**.



Step 5 Enter the Microsoft Windows **type** command to display the CRL.



Related Topics

Configuring Certificate Revocation Checking Methods.

7.9.7 Importing the CRL

To import the CRL to the trust point corresponding to the CA, follow these steps:

Step 1 Copy the CRL file to the Inspur INOS device bootflash.

```
Device-1# copy tftp:apranaCA.crl bootflash:aparnaCA.crl
```

Step 2 Configure the CRL.

```
Device-1# configure terminal
Device-1(config)# crypto ca crl request myCA bootflash:aparnaCA.crl
Device-1(config)#
```

Step 3 Display the contents of the CRL.

```
Device-1(config)# show crypto ca crl myCA
Trustpoint: myCA
CRL:
Certificate Revocation List (CRL):
        Version 2 (0x1)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: /emailAddress=admin@yourcompany.com/C=IN/ST=Karnatak
Yourcompany/OU=netstorage/CN=Aparna CA
        Last Update: Nov 12 04:36:04 2005 GMT
        Next Update: Nov 19 16:56:04 2005 GMT
        CRL extensions:
            X509v3       Authority        Key        Identifier:
            keyid:27:28:F2:46:83:1B:AC:23:4C:45:4D:8E:C9:18:50:1
            1.3.6.1.4.1.311.21.1:
                ...
Revoked Certificates:
    Serial      Number:      611B09A1000000000002
        Revocation Date: Aug 16 21:52:19 2005 GMT
Serial Number: 4CDE464E000000000003
        Revocation Date: Aug 16 21:52:29 2005 GMT
    Serial Number: 4CFC2B42000000000004
        Revocation Date: Aug 16 21:52:41 2005 GMT
    Serial Number: 6C699EC2000000000005
        Revocation Date: Aug 16 21:52:52 2005 GMT
    Serial Number: 6CCF7DDC000000000006
        Revocation  Date:  Jun  8  00:12:04  2005  GMT
    Serial Number: 70CC4FFF000000000007
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 4D9B1116000000000008
        Revocation Date: Aug 16 21:53:15 2005 GMT
    Serial Number: 52A80230000000000009
        Revocation Date: Jun 27 23:47:06 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            CA Compromise
Serial Number: 5349AD4600000000000A
        Revocation Date: Jun 27 23:47:22 2005 GMT
        CRL entry extensions:
            X509v3 CRL Reason Code:
            CA Compromise
Serial Number: 53BD173C00000000000B
        Revocation Date: Jul  4 18:04:01 2005 GMT
        CRL    entry    extensions:
            X509v3 CRL Reason Code:
            Certificate Hold
Serial Number: 591E7ACE00000000000C
        Revocation Date: Aug 16 21:53:15 2005 GMT
```

```
              Serial Number: 5D3FD52E000000000000D
                  Revocation Date: Jun 29 22:07:25 2005 GMT
                  CRL entry extensions:
                      X509v3 CRL Reason Code:
                      Key Compromise
          Serial Number: 5DAB7713000000000000E
                  Revocation Date: Jul 14 00:33:56 2005 GMT
              Serial Number: 5DAE53CD00000000000F
                  Revocation Date: Aug 16 21:53:15 2005 GMT
              Serial Number: 5DB140D3000000000010
                  Revocation Date: Aug 16 21:53:15 2005 GMT
              Serial Number: 5E2D7C1B000000000011
                  Revocation Date: Jul 6 21:12:10 2005 GMT
                  CRL entry extensions:
                      X509v3 CRL Reason Code:
                      Cessation Of Operation
          Serial Number: 16DB4F8F000000000012
                  Revocation Date: Aug 16 21:53:15 2005 GMT
              Serial Number: 261C3924000000000013
                  Revocation Date: Aug 16 21:53:15 2005 GMT
              Serial Number: 262B5202000000000014
                  Revocation Date: Jul 14 00:33:10 2005 GMT
              Serial Number: 2634C7F2000000000015
                  Revocation Date: Jul 14 00:32:45 2005 GMT
              Serial Number: 2635B000000000000016
                  Revocation Date: Jul 14 00:31:51 2005 GMT
              Serial Number: 26485040000000000017
                  Revocation Date: Jul 14 00:32:25 2005 GMT
              Serial Number: 2A276357000000000018
          Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 3F88CBF7000000000019
    Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 6E4B5F5F00000000001A
    Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 725B89D800000000001B
    Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 735A887800000000001C
    Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 148511C700000000001D
    Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 14A7170100000000001E
    Revocation Date: Aug 16 21:53:15 2005 GMT
Serial Number: 14FC45B500000000001F
    Revocation Date: Aug 17 18:30:42 2005 GMT
Serial Number: 486CE80B000000000020
    Revocation Date: Aug 17 18:30:43 2005 GMT
Serial Number: 4CA4A3AA000000000021
    Revocation Date: Aug 17 18:30:43 2005 GMT
Serial Number: 1AA55C8E00000000002F
    Revocation Date: Sep  5 17:07:06 2005 GMT
Serial Number: 3F0845DD00000000003F
    Revocation Date: Sep  8 20:24:32 2005 GMT
Serial Number: 3F619B7E000000000042
    Revocation Date: Sep  8 21:40:48 2005 GMT
Serial Number: 6313C463000000000052
    Revocation Date: Sep 19 17:37:18 2005 GMT
Serial Number: 7C3861E3000000000060
    Revocation Date: Sep 20 17:52:56 2005 GMT
Serial Number: 7C6EE351000000000061

                  Revocation Date: Sep 20 18:52:30 2005 GMT
              Serial Number: 0A338EA1000000000074   <-- Revoked identity certificate
                  Revocation Date: Nov 12 04:34:42 2005 GMT
              Signature Algorithm: sha1WithRSAEncryption
```

```
0b:cb:dd:43:0a:b8:62:1e:80:95:06:6f:4d:ab:0c:d8:8e:32:
44:8e:a7:94:97:af:02:b9:a6:9c:14:fd:eb:90:cf:18:c9:96:
29:bb:57:37:d9:1f:d5:bd:4e:9a:4b:18:2b:00:2f:d2:6e:c1:
1a:9f:1a:49:b7:9c:58:24:d7:72
```

Note The identity certificate for the device that was revoked (serial number 0A338EA1000000000074) is listed at the end.

7.10 Additional References forPKI

This section includes additional information related to implementing PKI.

7.10.1 Related Documents for PKI

Related Topic	Document Title
Inspur INOS Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*
VRF configuration	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*

7.10.2 Standards for PKI

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

CHAPTER 8 Configuring User Accounts and RBAC

This chapter describes how to configure user accounts and role-based access control (RBAC) on Inspur INOS devices.

This chapter includes the following sections:

- Finding Feature Information.
- Information About User Accounts and RBAC.
- Virtualization Support for RBAC.
- Licensing Requirements for User Accounts and RBAC.
- Guidelines and Limitations for User Accounts and RBAC.
- Default Settings for User Accounts and RBAC.
- Enabling Password-Strength Checking.
- Configuring User Accounts.
- Configuring Roles.
- Verifying User Accounts and RBAC Configuration.
- Configuration Examples for User Accounts and RBAC.
- Additional References for User Accounts and RBAC.
- Feature History for User Accounts and RBAC.

8.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

8.2 Information About User Accounts and RBAC

You can create and manage users accounts and assign roles that limit access to operations on the Inspur INOS device. RBAC allows you to define the rules for an assign role that restrict the authorization that the user has to access management operations.

8.2.1 User Accounts

You can configure up to a maximum of 256 user accounts. By default, the user account does not expire unless you explicitly configure it to expire. The expire option determines the date when the user account is disabled.

Users can have user accounts on multiple VDCs. These users can move between VDCs after an initial connection to a VDC.

The following words are reserved and cannot be used to configure users: bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator, games, gopher, ftp, nobody, nscd, mailnull, root, rpc, rpcuser, xfs, gdm, mtsuser, ftpuser, man, and sys.

Caution	Usernames must begin with an alphanumeric character in Inspur INOS Releases 8.2(3) and earlier releases. Usernames can contain only these special characters: (+ = . _ \ -). The # and ! symbols are not supported. If the username contains characters that are not allowed, the specified user is unable to log in.

8.2.2 Characteristics of Strong Passwords

A strong password has the following characteristics:

- Is at least eight characters long
- Does not contain many consecutive characters (such as abcd)
- Does not contain many repeating characters (such as aaabbb)
- Does not contain dictionary words
- Does not contain proper names
- Contains both uppercase and lowercase characters
- Contains numbers

The following are examples of strong passwords:
- 2004AsdfLkj30
- Cb1955S21

If a password is trivial (such as a short, easy-to-decipher password), the Inspur INOS software will reject your password configuration if password-strength checking is enabled. Be sure to configure a strong password as shown in the sample configuration. Passwords are case sensitive.

Related Topics

Enabling Password-Strength Checking.

8.2.3 User Roles

User roles contain rules that define the operations allowed for the user who is assigned the role. Each user role can contain multiple rules and each user can have multiple roles. For example, if role1 allows access only to configuration operations, and role2 allows access only to debug operations, then users who belong to both role1 and role2 can access configuration and debug operations. You can also limit access to specific VLANs, virtual routing and forwarding instances (VRFs), and interfaces.

The Inspur INOS software provides four default user roles:
- network-admin—Complete read-and-write access to the entire Inspur INOS device (only available in the default VDC)
- network-operator—Complete read access to the entire Inspur INOS device (only available in the default VDC)
- vdc-admin—Read-and-write access limited to a VDC
- vdc-operator—Read access limited to a VDC

You can create custom roles within a VDC. By default, the user accounts without administrator roles can access only the **show**, **exit**, **end**, and **configure terminal** commands. You can add rules to allow users to configure features.

The VDCs on the same physical device do not share user roles. Each VDC maintains an independent user role database. Within a VDC, roles are configured by rule and attribute assignment.

8.2.4 User Role Rules

The rule is the basic element of a role. A rule defines what operations the role allows the user to perform. You can apply rules for the following parameters:

Command

A command or group of commands defined in a regular expression.

Feature

A command or group of commands defined in a regular expression.

Feature group

Default or user-defined group of features.

OID

An SNMP object identifier (OID).

The command, feature, and feature group parameters create a hierarchical relationship. The most basic control parameter is the command. The next control parameter is the feature, which represents all commands associated with the feature. The last control parameter is the feature group. The feature group combines related features and allows you to easily manage the rules. The Inspur INOS software also supports the predefined feature group L3 that you can use.

You can configure up to 256 rules for each role. The user-specified rule number determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.

8.2.5 User Role Configuration Distribution

Inspur Fabric Services (CFS) allows the Inspur INOS device to distribute the user role configuration to other Inspur INOS devices in the network. When you enable CFS distribution for a feature on your device, the device belongs to a CFS region containing other devices in the network that you have also enabled for CFS distribution for the feature. CFS distribution for the user role feature is disabled by default.

After you enable CFS distribution for user roles on your Inspur INOS device, the first user role configuration command that you enter causes the Inspur INOS software to take the following actions:

- Creates a CFS session on your Inspur INOS device.
- Locks the user role configuration on all Inspur INOS devices in the CFS region with CFS enabled for the user role feature.
- Saves the user role configuration changes in a temporary buffer on the Inspur INOS device.

The changes stay in the temporary buffer on the Inspur INOS device until you explicitly commit them to be distributed to the devices in the CFS region. When you commit the changes, the Inspur INOS software takes the following actions:

- Applies the changes to the running configuration on your Inspur INOS device.
- Distributes the updated user role configuration to the other Inspur INOS devices in the CFS region.
- Unlocks the user role configuration in the devices in the CFS region.
- Terminates the CFS session.

For detailed information on CFS, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.

8.3 Virtualization Support for RBAC

The users with the network-admin and network-operator roles can operate in all virtual device contexts (VDCs) when logged in from the default VDC and use the **switchto vdc** command to access other VDCs. All other user roles are local to the VDC. Roles are not shared between VDCs. Each VDC maintains an independent user role database.

The following guidelines and limitations apply to the **switchto vdc** command:

- Only users with the network-admin or network-operator role can use the **switchto vdc** command. No other users are permitted to use it.
- No user can grant permission to another role to use the **switchto vdc** command.
- After a network-admin uses the **switchto vdc** command, this user becomes a vdc-admin for the new VDC. Similarly, after a network-operator uses the **switchto vdc** command, this user becomes a vdc-operator for the new VDC. Any other roles associated with the user are not valid after the **switchto vdc** command is entered.
- After a network-admin or network-operator uses the switchto vdc command, this user cannot use this command to switch to another VDC. The only option is to use the **switchback** command to return to the original VDC.

8.4 Licensing Requirements for User Accounts and RBAC

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS	User accounts and RBAC require no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

8.5 Guidelines and Limitations for User Accounts and RBAC

User accounts and RBAC have the following configuration guidelines and limitations:

- You can create up to 64 user-defined roles in a VDC in addition to the four default user roles in the default VDC and the two default user roles in the nondefault VDCs.
- You can add up to 256 rules to a user role.
- You can add up to 64 user-defined feature groups to a VDC in addition to the default feature group, L3.
- You can configure up to 256 users in a VDC.
- You can assign a maximum of 64 user roles to a user account.
- If you have a user account configured on the local Inspur INOS device that has the same name as a remote user account on an AAA server, the Inspur INOS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.
- You cannot delete the default admin and SNMP user accounts.
- You cannot remove the default user roles from the default admin user accounts.
- The network-operator and vdc-operator roles cannot run the **show running-config** and **show startup-config** commands.
- The following guidelines are applicable for the **rule** command:
- When you use the **rule** *rule-id* **permit command** *command-string* command, the *command-string* argument should be complete or it should contain an asterisk (*) after the command name, for example, **show *** or **show running-config ***.
- If you are adding more than one command in the command-string argument, the commands should be separated by a command separator (;) and a whitespace should be added.
- When you are specifying interfaces, it is recommended to specify the entire media type keyword such as Ethernet or loopback. However, if you are using the short form of the media type keyword, it should be followed by an asterisk (*).

For example, rule 22 permit command show run int Ethernet4/1, rule 22 permit command show run int loopback1, or rule 22 permit command show run int eth*.

Rules that do not follow this guideline are not accepted. For example, **rule 22 permit command show run int Eth1/4** and **rule 22 permit command show run int loop1**.

8.6 Default Settings for User Accounts and RBAC

This table lists the default settings for user accounts and RBAC parameters.

Table 15 : Default User Accounts and RBAC Parameters

Parameters	Default
User account password	Undefined.
User account expiry date	None.

User account role in the default VDC	Network-operator if the creating user has the network-admin role, or vdc-operator if the creating user has the vdc-admin role.
User account role in the non-VDCs	Vdc-operator if the creating user has the vdc-admin role.
Default user roles in the default VDC	Network-operator.
Default user roles in the non-default VDCs	Vdc-operator.
Interface policy	All interfaces are accessible.
VLAN policy	All VLANs are accessible.
VRF policy	All VRFs are accessible.
Feature group	L3.

8.7 Enabling Password-Strength Checking

You can enable password-strength checking which prevents you from creating weak passwords for user accounts.

SUMMARY STEPS

1. **configure terminal**
2. **password strength-check**
3. **exit**
4. (Optional) **show password strength-check**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**password strength-check** **Example:** `switch(config)# password strength-check`	Enables password-strength checking. The default is enabled. You can disable password-strength checking by using the **no** form of this command.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show password strength-check** **Example:** `switch# show password strength-check`	Displays the password-strength check configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Characteristics of Strong Passwords.

8.8 Configuring User Accounts

You can create a maximum of 256 user accounts on a Inspur INOS device. User accounts have the following attributes:

- Username
- Password
- Expiry date
- User roles

You can enter the password in clear text format or encrypted format. The Inspur INOS password encrypts clear text passwords before saving them to the running configuration. Encrypted format passwords are saved to the running configuration without further encryption. SHA256 is the hashing algorithm used for password encryption. As a part of the encryption, a 5000 iteration of 64-bit SALT is added to the password.

User accounts can have a maximum of 64 user roles. The user can determine what commands are available by using the command-line interface (CLI) context sensitive help utility.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **show role**
3. **username** *user-id* [**password** [**0** | **5**] *password*] [**expire** *date*] [**role** *role-name*]
4. **exit**
5. (Optional) **show user-account**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **show role** **Example:** `switch(config)# show role`	Displays the user roles available. You can configure other user roles, if necessary.
Step 3	**username** *user-id* [**password** [**0** | **5**] *password*] [**expire** *date*] [**role** *role-name*] **Example:** `switch(config)# username NewUser password 4Ty18Rnt`	Configures a user account. The *user-id* argument is a case-sensitive, alphanumeric character string with a maximum length of 28 characters. Valid characters are uppercase letters A through Z, lowercase letters a through z, numbers 0 through 9, hypen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The default password is undefined. The **0** option indicates that the password is clear text, and the **5** option indicates that the password is encrypted. The

	Command or Action	Purpose
		default is **0** (clear text). **Note** If you do not specify a password, the user might not be able to log in to the Inspur INOS device. **Note** If you create a user account with the encrypted password option, the corresponding SNMP user will not be created. The **expire** *date* option format is YYYY-MM-DD. The default is no expiry date. User accounts can have a maximum of 64 user roles.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 5	(Optional) **show user-account** **Example:** `switch# show user-account`	Displays the role configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring Roles.
Creating User Roles and Rules.

8.9 Configuring Roles

This section describes how to configure user roles.

8.9.1 Enabling User Role Configuration Distribution

To distribute the user roles configuration to other Inspur INOS devices in the network, you must first enable CFS distribution for user roles.

SUMMARY STEPS

1. **configure terminal**
2. **role distribute**
3. **exit**
4. (Optional) **show role session status**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**role distribute** **Example:** `switch(config)# role distribute`	Enables user role configuration distribution. The default is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show role session status** **Example:** `switch# show role session status`	Displays the user role distribution status information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating User Roles and Rules

You can configure up to 64 user roles in a VDC. Each user role can have up to 256 rules. You can assign a user role to more than one user account.

The rule number that you specify determines the order in which the rules are applied. Rules are applied in descending order. For example, if a role has three rules, rule 3 is applied before rule 2, which is applied before rule 1.

Before you begin

If you want to distribute the user role configuration, enable user role configuration distribution on all Inspur INOS devices to which you want the configuration distributed.

SUMMARY STEPS

1. **configure terminal**
2. **role name** *role-name*
3. **rule** *number* {**deny** | **permit**} **command** *command-string*
4. **rule** *number* {**deny** | **permit**} {**read** | **read-write**}
5. **rule** *number* {**deny** | **permit**} {**read** | **read-write**} **feature** *feature-name*
6. **rule** *number* {**deny** | **permit**} {**read** | **read-write**} **feature-group** *group-name*
7. **rule** *number* {**deny** | **permit**} {**read** | **read-write**} **oid** *snmp_oid_name*
8. (Optional) **description** *text*
9. **exit**
10. (Optional) **show role**
11. (Optional) **show role** {**pending** | **pending-diff**}
12. (Optional) **role commit**
13. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**role name** *role-name*
	Example:
	`switch(config)# role name UserA`
	`switch(config-role)#`
		Specifies a user role and enters role configuration mode. The *role-name* argument is a case-sensitive, alphanumeric character string with a maximum length of 16 characters.

Step 3	**rule** *number* {**deny** | **permit**} **command** *command-string* **Example:** `switch(config-role)# rule 1 deny command clear users`	Configures a command rule. The *command-string* argument can contain spaces and regular expressions. For example, interface ethernet includes all Ethernet interfaces. Repeat this command for as many rules as needed. For more information about guidelines for this command, see Guidelines and Limitations for User Accounts and RBAC,
Step 4	**rule** *number* {**deny** | **permit**} {**read** | **read-write**} **Example:** `switch(config-role)# rule 2 deny read-write`	Configures a read-only or read-and-write rule for all operations.
Step 5	**rule** *number* {**deny** | **permit**} {**read** | **read-write**} **feature** *feature-name* **Example:** `switch(config-role)# rule 3 permit read feature router-bgp`	Configures a read-only or read-and-write rule for a feature. Use the **show role feature** command to display a list of features. Repeat this command for as many rules as needed.
Step 6	**rule** *number* {**deny** | **permit**} {**read** | **read-write**} **feature-group** *group-name* **Example:** `switch(config-role)# rule 4 deny read-write feature-group L3`	Configures a read-only or read-and-write rule for a feature group. Use the **show role feature-group** command to display a list of feature groups. Repeat this command for as many rules as needed.
Step 7	**rule** *number* {**deny** | **permit**} {**read** | **read-write**} **oid** *snmp_oid_name* **Example:** `switch(config-role)# rule 5 deny read-write oid 1.3.6.1.2.1.1.9`	Configures a read-only or read-and-write rule for an SNMP object identifier (OID). You can enter up to 32 elements for the OID. This command can be used to allow SNMP-based performance monitoring tools to poll devices but restrict their access to system-intensive branches such as the IP routing table, ARP cache, MAC address tables, specific MIBs, and so on. **Note** The deepest OID can be at the scalar level or at the table root level.

	Command or Action	Purpose
		Repeat this command for as many rules as needed.
Step 8	(Optional) **description** *text* **Example:** `switch(config-role)# description This role does not allow users to use clear commands`	Configures the role description. You can include spaces in the description.

Step 9	**exit**
	Example:
	`switch(config-role)# exit`
	`switch(config)#`
		Exits role configuration mode.

Step 10	(Optional) **show role** **Example:** `switch(config)# show role`	Displays the user role configuration.
Step 11	(Optional) **show role** {**pending** | **pending-diff**} **Example:** `switch(config)# show role pending`	Displays the user role configuration pending for distribution.
Step 12	(Optional) **role commit** **Example:** `switch(config)# role commit`	Applies the user role configuration changes in the temporary database to the running configuration and distributes user role configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.
Step 13	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Committing the User Role Configuration to Distribution.

8.9.2 Creating Feature Groups

You can create custom feature groups to add to the default list of features provided by the Inspur INOS software. These groups contain one or more of the features. You can create up to 64 feature groups in a VDC.

Before you begin

If you want to distribute the user role configuration, enable user role configuration distribution on all Inspur INOS devices to which you want the configuration distributed.

SUMMARY STEPS

1. **configure terminal**
2. **role feature-group name** *group-name*
3. **feature** *feature-name*
4. **exit**
5. (Optional) **show role feature-group**
6. (Optional) **show role** {**pending** | **pending-diff**}

7. (Optional) **role commit**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**role feature-group name** *group-name*
	Example:
	`switch(config)# role feature-group name GroupA`
	`switch(config-role-featuregrp)#`
		Specifies a user role feature group and enters role feature group configuration mode.
		The *group-name* argument is a case-sensitive, alphanumeric character string with a maximum length of 32 characters.

Step 3	**feature** *feature-name* **Example:** `switch(config-role-featuregrp)# feature vdc`	Specifies a feature for the feature group. Repeat this command for as many features as needed. **Note** Use the **show role component** command to display a list of features.

Step 4	**exit**
	Example:
	`switch(config-role-featuregrp)# exit`
	`switch(config)#`
		Exits role feature group configuration mode.

Step 5	(Optional) **show role feature-group** **Example:** `switch(config)# show role feature-group`	Displays the role feature group configuration.
Step 6	(Optional) **show role** {**pending** | **pending-diff**} **Example:** `switch(config)# show role pending`	Displays the user role configuration pending for distribution.
Step 7	(Optional) **role commit** **Example:** `switch(config)# role commit`	Applies the user role configuration changes in the temporary database to the running configuration and distributes user role configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Committing the User Role Configuration to Distribution.

8.9.3 Changing User Role Interface Policies

You can change a user role interface policy to limit the interfaces that the user can access. By default, a user role allows access to all interfaces in the VDC.

Before you begin

Create one or more user roles.

If you want to distribute the user role configuration, enable user role configuration distribution on all Inspur INOS devices to which you want the configuration distributed.

SUMMARY STEPS

1. **configure terminal**
2. **role name** *role-name*
3. **interface policy deny**
4. **permit interface** *interface-list*
5. **exit**
6. (Optional) **show role**
7. (Optional) **show role** {**pending** | **pending-diff**}
8. (Optional) **role commit**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**role name** *role-name*
	Example:
	`switch(config)# role name UserA`
	`switch(config-role)#`
		Specifies a user role and enters role configuration mode.

Step 3	**interface policy deny**
	Example:
	`switch(config-role)# interface policy deny`
	`switch(config-role-interface)#`
		Enters role interface policy configuration mode.

Step 4	**permit interface** *interface-list* **Example:** `switch(config-role-interface)# permit interface ethernet 2/1-4`	Specifies a list of interfaces that the role can access. Repeat this command for as many interfaces as needed.

Step 5	**exit**
	Example:
	`switch(config-role-interface)# exit`
	`switch(config-role)#`
		Exits role interface policy configuration mode.

Step 6	(Optional) **show role** **Example:** `switch(config-role)# show role`	Displays the role configuration.
Step 7	(Optional) **show role** {**pending** | **pending-diff**}	Displays the user role configuration pending for

	Command or Action	Purpose
	Example: `switch(config-role)# show role pending`	distribution.
Step 8	(Optional) **role commit** **Example:** `switch(config-role)# role commit`	Applies the user role configuration changes in the temporary database to the running configuration and distributes user role configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-role)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Creating User Roles and Rules.
Committing the User Role Configuration to Distribution.

8.9.4 Changing User Role VLAN Policies

You can change a user role VLAN policy to limit the VLANs that the user can access. By default, a user role allows access to all VLANs in the VDC.

Before you begin

Create one or more user roles.

If you want to distribute the user role configuration, enable user role configuration distribution on all Inspur INOS devices to which you want the configuration distributed.

SUMMARY STEPS

1. **configure terminal**
2. **role name** *role-name*
3. **vlan policy deny**
4. **permit vlan** *vlan-list*
5. **exit**
6. (Optional) **show role**
7. (Optional) **show role** {**pending** | **pending-diff**}
8. (Optional) **role commit**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**role name** *role-name* **Example:**	Specifies a user role and enters role configuration mode.

	Command or Action	Purpose
	`switch(config)# role name UserA` `switch(config-role)#`	

Step 3	**vlan policy deny**
	Example:
	`switch(config-role)# vlan policy deny`
	`switch(config-role-vlan)#`
		Enters role VLAN policy configuration mode.

Step 4	**permit vlan** *vlan-list* **Example:** `switch(config-role-vlan)# permit vlan 1-4`	Specifies a range of VLANs that the role can access. Repeat this command for as many VLANs as needed.

Step 5	**exit**
	Example:
	`switch(config-role-vlan)# exit`
	`switch(config-role)#`
		Exits role VLAN policy configuration mode.

Step 6	(Optional) **show role** **Example:** `switch(config)# show role`	Displays the role configuration.
Step 7	(Optional) **show role** {**pending** | **pending-diff**} **Example:** `switch(config-role)# show role pending`	Displays the user role configuration pending for distribution.
Step 8	(Optional) **role commit** **Example:** `switch(config-role)# role commit`	Applies the user role configuration changes in the temporary database to the running configuration and distributes user role configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-role)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Creating User Roles and Rules.
Committing the User Role Configuration to Distribution.

8.9.5 Changing User Role VRF Policies

You can change a user role VRF policy to limit the VRFs that the user can access. By default, a user role allows access to all VRFs in the VDC.

Before you begin

Create one or more user roles.

If you want to distribute the user role configuration, enable user role configuration distribution on all Inspur INOS devices to which you want the configuration distributed.

SUMMARY STEPS

1. **configure terminal**
2. **role name** *role-name*
3. **vrf policy deny**
4. **permit vrf** *vrf-name*
5. **exit**
6. (Optional) **show role**
7. (Optional) **show role** {**pending** | **pending-diff**}
8. (Optional) **role commit**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**role name** *role-name*
	Example:
	`switch(config)# role name`
	`UserA switch(config-role)#`
		Specifies a user role and enters role configuration mode.

Step 3	**vrf policy deny**
	Example:
	`switch(config-role)# vrf policy`
	`deny switch(config-role-vrf)#`
		Enters role VRF policy configuration mode.

Step 4	**permit vrf** *vrf-name* **Example:** `switch(config-role-vrf)# permit vrf vrf1`	Specifies the VRF that the role can access. Repeat this command for as many VRFs as needed.

Step 5	**exit**
	Example:
	`switch(config-role-vrf)# exit`
	`switch(config-role)#`
		Exits role VRF policy configuration mode.

Step 6	(Optional) **show role** **Example:** `switch(config-role)# show role`	Displays the role configuration.
Step 7	(Optional) **show role** {**pending** | **pending-diff**} **Example:** `switch(config-role)# show role pending`	Displays the user role configuration pending for distribution.
Step 8	(Optional) **role commit** **Example:** `switch(config-role)# role commit`	Applies the user role configuration changes in the temporary database to the running configuration and distributes user role configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

	Command or Action	Purpose
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config-role)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Creating User Roles and Rules.
Committing the User Role Configuration to Distribution.

8.9.6 Committing the User Role Configuration to Distribution

You can apply the user role global and/or server configuration stored in the temporary buffer to the running configuration across all switches in the fabric (including the originating switch).

Before you begin

You have enabled user role configuration distribution on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **show role** {**pending** | **pending-diff**}
3. (Optional) **role commit**
4. **exit**
5. (Optional) **show role session status**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **show role** {**pending** | **pending-diff**} **Example:** `switch(config)# show role pending`	Displays the user role configuration pending for distribution.
Step 3	(Optional) **role commit** **Example:** `switch(config)# role commit`	Applies the user role configuration changes in the temporary database to the running configuration and distributes user role configuration to other Inspur INOS devices if you have enabled CFS configuration distribution for the user role feature.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show role session status**	Displays the user role CFS session status.

	Command or Action	Purpose
	Example: `switch# show role session status`	
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Applies the running configuration to the startup configuration on all Inspur INOS devices in the network that have CFS enabled.

Related Topics

User Role Configuration Distribution.

8.9.7 Discarding the User Role Distribution Session

You can discard the temporary database of user role changes and end the CFS distribution session.

Before you begin

You have enabled user role configuration distribution on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **show role** {**pending** | **pending-diff**}
3. **role abort**
4. **exit**
5. (Optional) **show role session status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	(Optional) **show role** {**pending** | **pending-diff**} **Example:** `switch(config)# show role pending`	Displays the user role configuration pending for distribution.
Step 3	**role abort** **Example:** `switch(config)# role abort`	Discards the user role configuration in the temporary storage and ends the session.
Step 4	**exit** **Example:** `switch(config)# exit` `switch#`	Exits configuration mode.
Step 5	(Optional) **show role session status** **Example:** `switch# show role session status`	Displays the user role CFS session status.

Related Topics

Committing the User Role Configuration to Distribution.
User Role Configuration Distribution.

8.9.8 Clearing the User Role Distribution Session

You can clear the ongoing Inspur Fabric Services distribution session (if any) and unlock the fabric for the user role feature.

You have enabled user role configuration distribution on the Inspur INOS device.

SUMMARY STEPS

1. **clear role session**
2. (Optional) **show role session status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear role session** **Example:** `switch# clear role session`	Clears the session and unlocks the fabric.
Step 2	(Optional) **show role session status** **Example:** `switch# show role session status`	Displays the user role CFS session status.

Related Topics

Committing the User Role Configuration to Distribution.
User Role Configuration Distribution.

8.10 Verifying User Accounts and RBAC Configuration

To display user account and RBAC configuration information, perform one of the following tasks:

Command	Purpose
show role	Displays the user role configuration.
show role feature	Displays the feature list.
show role feature-group	Displays the feature group configuration.
show startup-config security	Displays the user account configuration in the startup configuration.
show running-config security [**all**]	Displays the user account configuration in the running configuration. The **all** keyword displays the default values for the user accounts.

show user-account	Displays user account information.

For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

8.11 Configuration Examples for User Accounts and RBAC

The following example shows how to configure a user role:

```
role name User-role-A
  rule 3 permit read-write feature l2nac
  rule 2 permit read-write feature dot1x
  rule 1 deny command clear *
```

The following example shows how to create a user role that can configure an interface to enable and show HSRP and show GLBP:

```
role name iftest
   rule 1 permit command config t; interface *; hsrp *
   rule 2 permit read-write feature hsrp
   rule 3 permit read feature glbp
```

In the above example, rule 1 allows you to configure HSRP on an interface, rule 2 allows you to configure the **config hsrp** commands and enable the exec-level **show** and **debug** commands for HSRP, and rule 3 allows you to enable the exec-level **show** and **debug glbp** commands.

The following example shows how to configure a user role that can configure only a specific interface:

```
role name Int_Eth2-3_only
  rule 1 permit command configure terminal; interface *
  interface policy deny
    permit interface Ethernet2/3
```

The following example shows how to configure a user role feature group:

```
role feature-group name Security-features
  feature radius
  feature tacacs
  feature  dot1x
  feature    aaa
  feature  l2nac
  feature acl
  feature access-list
```

The following example shows how to configure a user account:

```
username user1 password A1s2D4f5 role User-role-A
```

The following example shows how to add an OID rule to restrict access to part of the OID subtree:

```
role name User1
  rule 1 permit read feature snmp
  rule 2 deny read oid 1.3.6.1.2.1.1.9
show role name User1

Role:            User1
```

```
  Description: new role
  Vlan   policy:   permit   (default)
  Interface policy: permit (default)
  Vrf policy: permit (default)
  -------------------------------------------------------------------
  Rule    Perm    Type        Scope               Entity
  -------------------------------------------------------------------
  2       deny    read        oid                 1.3.6.1.2.1.1.9
  1       permit  read        feature             snmp
```

The following example shows how to give write permission to a specified OID subtree:

```
role name User1
rule 3 permit read-write oid 1.3.6.1.2.1.1.5
show role name User1

Role:             User1
  Description: new role
  Vlan   policy:   permit   (default)
  Interface policy: permit (default)
  Vrf policy: permit (default)
  -------------------------------------------------------------------
  Rule    Perm    Type        Scope               Entity
  -------------------------------------------------------------------
  3       permit  read-write  oid                 1.3.6.1.2.1.1.5
  2       deny    read        oid                 1.3.6.1.2.1.1.9
  1       permit  read        feature             snmp
```

8.12 Additional References for User Accounts and RBAC

This section includes additional information related to implementing user accounts and RBAC.

Related Documents

Related Topic	Document Title
Inspur INOS Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*
VRF configuration	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
• INSPUR-COMMON-MGMT-MIB	-

8.12.1 Related Documents for User Accounts and RBAC

Related Topic	Document Title

Inspur INOS Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*
VRF configuration	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*

8.12.2 Standards for User Accounts and RBAC

Standards	**Title**
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

8.12.3 MIBs for User Accounts and RBAC

MIBs	**MIBs Link**
• INSPUR-COMMON-MGMT-MIB	-

8.13 Feature History for User Accounts and RBAC

This table lists the release history for this feature.

Table 16 : Feature History for User Accounts and RBAC

Feature Name	**Releases**	**Feature Information**
RBAC	8.2(3)	Added support for F3Series modules.
User accounts and RBAC	8.2(3)	Added the ability to configure a read-only or read-and-write rule for an SNMP OID.
User accounts and RBAC	8.2(3)	No change from Release 8.2(3).
User accounts and RBAC	8.2(3)	Added support for the Inspur CN3000 Series Switches.
User roles	8.2(3)	Added the ability to display the syntax of the commands that the network-admin and network-operator roles can use.
User accounts and RBAC	8.2(3)	No change from Release 8.2(3)
User accounts and RBAC	8.2(3)	Added the ability to support the at symbol (@) in remote usernames.
User accounts and RBAC	8.2(3)	No change from Release 8.2(3)
Usernames	8.2(3)	Valid characters in username are limited to lowercase a through z, uppercase A through Z, the numbers 0 through 9, plus sign (+), hyphen (-), equal sigh (=), underscore (_) and period (.).

CHAPTER 9 Configuring 802.1X

This chapter describes how to configure IEEE 802.1X port-based authentication on Inspur INOS devices. This chapter includes the following sections:

- Finding Feature Information.
- Information About 802.1X.
- Licensing Requirements for 802.1X.
- Prerequisites for 802.1X.
- 802.1X Guidelines and Limitations.
- Default Settings for 802.1X.
- Configuring 802.1X.
- Verifying the 802.1X Configuration.
- Monitoring 802.1X.
- Configuration Example for 802.1X.
- Additional References for 802.1X.
- Feature History for 802.1X.

9.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

9.2 Information About 802.1X

802.1 X defines a client-server-based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. The authentication server authenticates each client connected to a Inspur INOS device port.

Until the client is authenticated, 802.1X access control allows only Extensible Authentication Protocol over LAN (EAPOL) traffic through the port to which the client is connected. After authentication is successful, normal traffic can pass through the port.

9.2.1 Device Roles

With 802.1X port-based authentication, the devices in the network have specific roles.

The specific roles are as follows:

Supplicant

The client device that requests access to the LAN and Inspur INOS device services and responds to requests from the Inspur INOS device. The workstation must be running 802.1X-compliant client software such as that offered in the Microsoft Windows XP operating device.

Authentication server

The authentication server performs the actual authentication of the supplicant. The authentication server validates the identity of the supplicant and notifies the Inspur INOS device regarding whether the supplicant is authorized to access the LAN and Inspur INOS device services. Because the Inspur INOS device acts as the proxy, the authentication service is transparent to the supplicant. The Remote Authentication Dial-In User Service (RADIUS) security device with Extensible Authentication Protocol (EAP) extensions is the only supported authentication server; it is available in Inspur Secure Access Control Server, version 3.0. RADIUS uses a supplicant-server model in which secure authentication information is exchanged between the RADIUS server and one or more RADIUS clients.

Authenticator

The authenticator controls the physical access to the network based on the authentication status of the supplicant. The authenticator acts as an intermediary (proxy) between the supplicant and the authentication server, requesting identity information from the supplicant, verifying the requested identity information with the authentication server, and relaying a response to the supplicant. The authenticator includes the RADIUS client, which is responsible for encapsulating and decapsulating the EAP frames and interacting with the authentication server.

When the authenticator receives EAPOL frames and relays them to the authentication server, the authenticator strips off the Ethernet header and encapsulates the remaining EAP frame in the RADIUS format. This encapsulation process does not modify or examine the EAP frames, and the authentication server must support EAP within the native frame format. When the authenticator receives frames from the authentication server, the authenticator removes the server's frame header, leaving the EAP frame, which the authenticator then encapsulates for Ethernet and sends to the supplicant.

9.2.2 Authentication Initiation and Message Exchange

Either the authenticator (Inspur INOS device) or the supplicant (client) can initiate authentication. If you enable authentication on a port, the authenticator must initiate authentication when it determines that the port link state transitions from down to up. The authenticator then sends an EAP-request/identity frame to the supplicant to request its identity (typically, the authenticator sends an initial identity/request frame followed by one or more requests for authentication information). When the supplicant receives the frame, it responds with an EAP-response/identity frame.

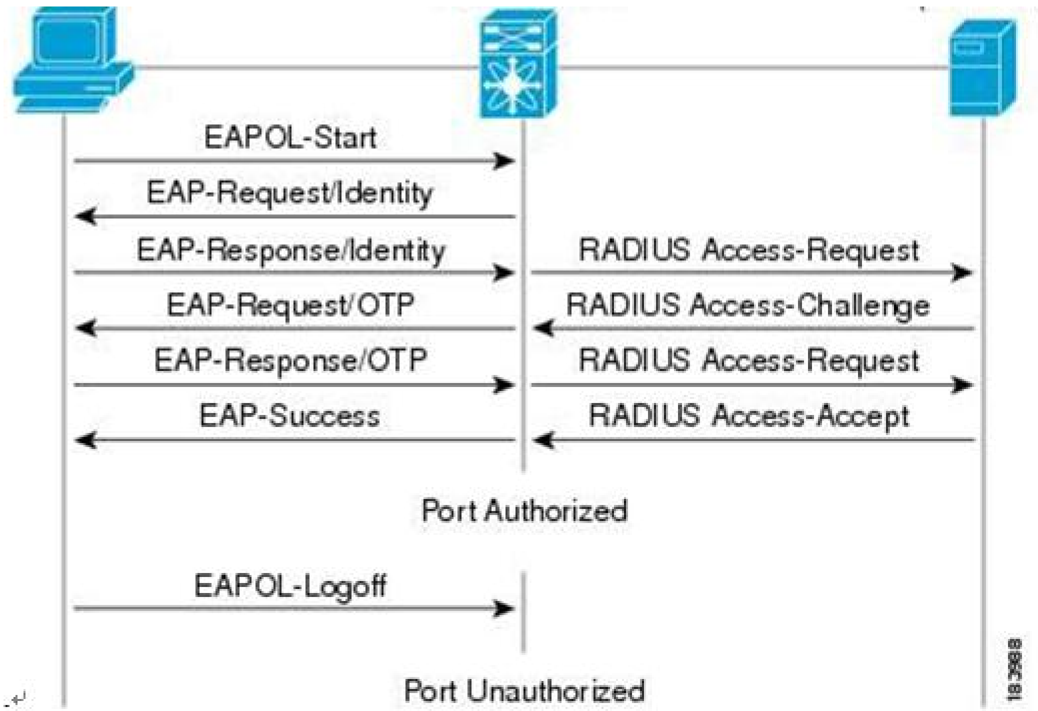
If the supplicant does not receive an EAP-request/identity frame from the authenticator during bootup, the supplicant can initiate authentication by sending an EAPOL-start frame, which prompts the authenticator to request the supplicant's identity.

When the supplicant supplies its identity, the authenticator begins its role as the intermediary, passing EAP frames between the supplicant and the authentication server until authentication succeeds or fails. If the authentication succeeds, the authenticator port becomes authorized.

The specific exchange of EAP frames depends on the authentication method being used.

Figure 3：Message Exchange

This figure shows a message exchange initiated by the supplicant using the One-Time-Password (OTP) authentication method with a RADIUS server. The OTP authentication device uses a secret pass-phrase to generate a sequence of one-time (single use) passwords.



The user's seret pass-phrase never crosses the network at any time such as during authentication or during pass-phrase changes.

Related Topics

Ports in Authorized and Unauthorized States.

9.2.3 Authenticator PAE Status for Interfaces

When you enable 802.1X on an interface, the Inspur INOS software creates an authenticator port access entity (PAE) instance. An authenticator PAE is a protocol entity that supports authentication on the interface. When you disable 802.1X on the interface, the Inspur INOS software does not automatically clear the authenticator PAE instances. You can explicitly remove the authenticator PAE from the interface and then reapply it, as needed.

9.2.4 Ports in Authorized and Unauthorized States

The authenticator port state determines if the supplicant is granted access to the network. The port starts in the unauthorized state. In this state, the port disallows all ingress and egress traffic except for 802.1X protocol packets. When a supplicant is successfully authenticated, the port transitions to the authorized state, allowing all traffic for the supplicant to flow normally.

If a client that does not support 802.1X is connected to an unauthorized 802.1X port, the authenticator requests the client's identity. In this situation, the client does not respond to the request, the port remains in the unauthorized state, and the client is not granted access to the network.

In contrast, when an 802.1X-enabled client connects to a port that is not running the 802.1X protocol, the client initiates the authentication process by sending the EAPOL-start frame. When no response is received, the client sends the request for a fixed number of times. Because no response is received, the client begins sending frames as if the port is in the authorized state.

Ports can have the following authorization states:

Force authorized

Disables 802.1X port-based authentication and transitions to the authorized state without requiring any authentication exchange. The port transmits and receives normal traffic without 802.1X-based authentication of the client. This authorization state is the default.

Force unauthorized

Causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The authenticator cannot provide authentication services to the client through the interface.

Auto

Enables 802.1X port-based authentication and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received through the port. The authentication process begins when the link state of the port transitions from down to up or when an EAPOL-start frame is received from the supplicant. The authenticator requests the identity of the client and begins relaying authentication messages between the client and the authentication server. Each supplicant that attempts to access the network is uniquely identified by the authenticator by using the supplicant's MAC address.

If the supplicant is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated supplicant are allowed through the port. If the authentication fails, the port remains in the unauthorized state, but authentication can be retried. If the authentication server cannot be reached, the authenticator can retransmit the request. If no response is received from the server after the specified number of attempts, authentication fails, and the supplicant is not granted network access.

When a supplicant logs off, it sends an EAPOL-logoff message, which causes the authenticator port to transition to the unauthorized state.

If the link state of a port transitions from up to down, or if an EAPOL-logoff frame is received, the port returns to the unauthorized state.

9.2.5 MAC Authentication Bypass

You can configure the Inspur INOS device to authorize a supplicant based on the supplicant MAC address by using the MAC authentication bypass feature. For example, you can enable this feature on interfaces configured for 802.1X that are connected to devices such as printers.

If 802.1X authentication times out while waiting for an EAPOL response from the supplicant, the Inspur INOS device tries to authorize the client by using MAC authentication bypass.

When you enable the MAC authentication bypass feature on an interface, the Inspur INOS device uses the MAC address as the supplicant identity. The authentication server has a database of supplicant MAC addresses that are allowed network access. After detecting a client on the interface, the Inspur INOS device waits for an Ethernet packet from the client. The Inspur INOS device sends the authentication server a RADIUS-access/request frame with a username and password based on the MAC address. If authorization succeeds, the Inspur INOS device grants the client access to the network. If authorization fails, the Inspur INOS device assigns the port to the guest VLAN if one is configured.

If an EAPOL packet is detected on the interface during the lifetime of the link, the Inspur INOS device determines that the device connected to that interface is an 802.1X-capable supplicant and uses 802.1X authentication (not MAC authentication bypass) to authorize the interface. EAPOL history is cleared if the interface link status goes down.

If the Inspur INOS device already authorized an interface by using MAC authentication bypass and detects an 802.1X supplicant, the Inspur INOS device does not unauthorize the client connected to the interface. When reauthentication occurs, the Inspur INOS device uses 802.1X authentication as the preferred reauthentication process if the previous session ended because the Termination-Action RADIUS attribute value is DEFAULT.

Clients that were authorized with MAC authentication bypass can be reauthenticated. The reauthentication process is the same as that for clients that were authenticated with 802.1X. During reauthentication, the port remains in the previously assigned VLAN. If reauthentication is successful, the switch keeps the port in the same VLAN. If reauthentication fails, the switch assigns the port to the guest VLAN, if one is configured.

If reauthentication is based on the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS attribute (Attribute [29]) and if the Termination-Action RADIUS attribute (Attribute [29]) action is Initialize (the attribute value is DEFAULT), the MAC authentication bypass session ends, and connectivity is lost during reauthentication. If MAC authentication bypass is enabled and the 802.1X authentication times out, the switch uses the MAC authentication bypass feature to initiate reauthorization. For more information about these AV pairs, see RFC 3580, *IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*.

MAC authentication bypass interacts with the following features:

- 802.1X authentication—You can enable MAC authentication bypass only if 802.1X authentication is enabled on the port.
- Port security— You can configure 802.1X authentication and port security on the same Layer 2 ports.
- Network admission control (NAC) Layer 2 IP validation—This feature takes effect after an 802.1X port is authenticated with MAC authentication bypass, including hosts in the exception list.

Related Topics

802.1X and Port Security.

9.2.6 802.1X and Port Security

You can configure port security and 802.1X on the same interfaces. Port security secures the MAC addresses that 802.1X authenticates. 802.1X processes packets before port security processes them, so when you enable both on an interface, 802.1X is already preventing inbound traffic on the interface from unknown MAC addresses.

When you enable 802.1X and port security on the same interface, port security continues to learn MAC addresses by the sticky or dynamic method, as configured. Additionally, depending on whether you enable 802.1X in single-host mode or multiple-host mode, one of the following occurs:

Single host mode

Port security learns the MAC address of the authenticated host.

Multiple host mode

Port security drops any MAC addresses learned for this interface by the dynamic method and learns the MAC address of the first host authenticated by 802.1X.

If a MAC address that 802.1X passes to port security would violate the applicable maximum number of secure MAC addresses, the device sends an authentication failure message to the host.

The device treats MAC addresses authenticated by 802.1X as though they were learned by the dynamic method, even if port security previously learned the address by the sticky or static methods. If you attempt to delete a secure MAC address that has been authenticated by 802.1X, the address remains secure.

If the MAC address of an authenticated host is secured by the sticky or static method, the device treats the address as if it were learned by the dynamic method, and you cannot delete the MAC address manually.

Port security integrates with 802.1X to reauthenticate hosts when the authenticated and secure MAC address of the host reaches its port security age limit. The device behaves differently depending upon the type of aging, as follows:

Absolute

Port security notifies 802.1X and the device attempts to reauthenticate the host. The result of reauthentication determines whether the address remains secure. If reauthentication succeeds, the device restarts the aging timer on the secure address; otherwise, the device drops the address from the list of secure addressees for the interface.

Inactivity

Port security drops the secure address from the list of secure addresses for the interface and notifies 802.1X. The device attempts to reauthenticate the host. If reauthentication succeeds, port security secures the address again.

9.2.7 Single Host and Multiple Hosts Support

The 802.1X feature can restrict traffic on a port to only one endpoint device (single-host mode) or allow traffic from multiple endpoint devices on a port (multi-host mode).

Single-host mode allows traffic from only one endpoint device on the 802.1X port. Once the endpoint device is authenticated, the Inspur INOS device puts the port in the authorized state. When the endpoint device leaves the port, the Inspur INOS device put the port back into the unauthorized state. A security violation in 802.1X is defined as a detection of frames sourced from any MAC address other than the single MAC address authorized as a result of successful authentication. In this case, the interface on which this security association violation is detected (EAPOL frame from the other MAC address) will be disabled. Single host mode is applicable only for host-to-switch topology and when a single host is connected to the Layer 2 (Ethernet access port) or Layer 3 port (routed port) of the Inspur INOS device.

Only the first host has to be authenticated on the 802.1X port configured with multiple host mode. The port is moved to the authorized state after the successful authorization of the first host. Subsequent hosts are not required to be authorized to gain network access once the port is in the authorized state. If the port becomes unauthorized when reauthentication fails or an EAPOL logoff message is received, all attached hosts are denied access to the network. The capability of the interface to shut down upon security association violation is disabled in multiple host mode. This mode is applicable for both switch-to-switch and host-to-switch topologies.

9.2.8 Supported Topologies

The 802.1X port-based authentication is supported in two topologies:

- Point-to-point
- Wireless LAN

In a point-to-point configuration, only one supplicant (client) can connect to the 802.1X-enabled authenticator (Inspur INOS device) port. The authenticator detects the supplicant when the port link state changes to the up state. If a supplicant leaves or is replaced with another supplicant, the authenticator changes the port link state to down, and the port returns to the unauthorized state.

When the port is authorized, all other hosts indirectly attached to the port are granted access to the network. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), the Inspur INOS device denies access to the network to all of the attached supplicants.

9.2.9 Virtualization Support for 802.1X

The 802.1X configuration and operation are local to the virtual device context (VDC). For more information on VDCs, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

9.3 Licensing Requirements for 802.1X

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS	802.1X requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

9.4 Prerequisites for 802.1X

802.1 X has the following prerequisites:

- One or more RADIUS servers are accessible in the network.

- 802.1X supplicants are attached to the ports, unless you enable MAC address authentication bypass.

Related Topics

Enabling MAC Authentication Bypass.

9.5 802.1X Guidelines and Limitations

802.1 X port-based authentication has the following configuration guidelines and limitations:

- The Inspur INOS software supports 802.1X authentication only on physical ports.
- The Inspur INOS software does not support 802.1X authentication on port channels or subinterfaces.
- When you enable 802.1X authentication, supplicants are authenticated before any other Layer 2 or Layer 3 features are enabled on an Ethernet interface.
- The Inspur INOS software supports 802.1X authentication only on Ethernet interfaces that are in a port channel, a trunk, or an access port.
- The Inspur INOS software does not support single host mode on trunk interfaces or member interfaces in a port channel.
- The Inspur INOS software does not support MAC address authentication bypass on trunk interfaces.
- The Inspur INOS software does not support MAC address authentication bypass on a port channel.
- The Inspur INOS software does not support Dot1X on vPC ports and MCT.
- The Inspur INOS software does not support the following 802.1X protocol enhancements:
- One-to-many logical VLAN name to ID mapping
- Web authorization
- Dynamic domain bridge assignment
- IP telephony
- The Inspur INOS software does not support dynamic VLAN assignment on the Inspur CN12700 series switches.

9.6 Default Settings for 802.1X

This table lists the default settings for 802.1X parameters.

Table 17：Default 802.1X Parameters

Parameters	Default
802.1X feature	Disabled
AAA 802.1X authentication method	Not configured
Per-interface 802.1X protocol enable state	Disabled (**force-authorized**) **Note** The port transmits and receives normal traffic without 802.1X-based authentication of the supplicant.
Periodic reauthentication	Disabled
Number of seconds between reauthentication attempts	3600 seconds
Quiet timeout period	60 seconds (number of seconds that the Inspur INOS device remains in the quiet state following a failed authentication exchange with the supplicant)
Retransmission timeout period	30 seconds (number of seconds that the Inspur INOS device should wait for a response to an EAP request/identity frame from the supplicant before retransmitting the request)

Parameters	Default
Maximum retransmission number	2 times (number of times that the Inspur INOS device will send an EAP-request/identity frame before restarting the authentication process)
Host mode	Single host
Supplicant timeout period	30 seconds (when relaying a request from the authentication server to the supplicant, the amount of time that the Inspur INOS device waits for a response before retransmitting the request to the supplicant)
Authentication server timeout period	30 seconds (when relaying a response from the supplicant to the authentication server, the amount of time that the Inspur INOS device waits for a reply before retransmitting the response to the server)

9.7 Configuring 802.1X

This section describes how to configure the 802.1X feature.

9.7.1 Process for Configuring 802.1X

This section describes the process for configuring 802.1X.

SUMMARY STEPS

1. Enable the 802.1X feature.
2. Configure the connection to the remote RADIUS server.
3. Enable 802.1X feature on the Ethernet interfaces.

DETAILED STEPS

Step 1 Enable the 802.1X feature.

Step 2 Configure the connection to the remote RADIUS server.

Step 3 Enable 802.1X feature on the Ethernet interfaces.

Related Topics

Enabling the 802.1X Feature.
Configuring AAA Authentication Methods for 802.1X.
Controlling 802.1X Authentication on an Interface.

9.7.2 Enabling the 802.1X Feature

You must enable the 802.1X feature on the Inspur INOS device before authenticating any supplicant devices.

SUMMARY STEPS

1. **configure terminal**
2. **feature dot1x**
3. **exit**
4. (Optional) **show dot1x**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature dot1x** **Example:** `switch(config)# feature dot1x`	Enables the 802.1X feature. The default is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X feature status.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

9.7.3 Configuring AAA Authentication Methods for 802.1X

You can use remote RADIUS servers for 802.1X authentication. You must configure RADIUS servers and RADIUS server groups and specify the default AAA authentication method before the Inspur INOS device can perform 802.1X authentication.

Before you begin

Obtain the names or addresses for the remote RADIUS server groups.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authentication dot1x default group** *group-list*
3. **exit**
4. (Optional) **show radius-server**
5. (Optional) **show radius-server group** [*group-name*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authentication dot1x default group** *group-list* **Example:**	Specifies the RADIUS server groups to use for 802.1X authentication.

	Command or Action	Purpose
	`switch(config)# aaa authentication dot1x default group rad2`	The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius**—Uses the global pool of RADIUS servers for authentication. • *named-group* —Uses the global pool of RADIUS servers for authentication.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.
Step 5	(Optional) **show radius-server group** [*group-name*] **Example:** `switch# show radius-server group rad2`	Displays the RADIUS server group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring AAA.
Configuring RADIUS.

9.7.4 Controlling 802.1X Authentication on an Interface

You can control the 802.1X authentication performed on an interface. An interface can have the following 802.1X authentication states:

Auto

Enables 802.1X authentication on the interface.

Force-authorized

Disables 802.1X authentication on the interface and allows all traffic on the interface without authentication. This state is the default.

Force-unauthorized

Disallows all traffic on the interface.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot* / *port*
3. **dot1x port-control {auto | force-authorized | forced-unauthorized}**
4. **exit**
5. (Optional) **show dot1x all**
6. (Optional) **show dot1x interface ethernet** *slot* / *port*
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot* / *port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x port-control** {**auto** | **force-authorized** | **forced-unauthorized**} **Example:** `switch(config-if)# dot1x port-control auto`	Changes the 802.1X authentication state on the interface. The default is force-authorized.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **show dot1x interface ethernet** *slot* / *port* **Example:** `switch# show dot1x interface ethernet 2/1`	Displays 802.1X feature status and configuration information for an interface.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the 802.1X Feature.

9.7.5 Creating or Removing an Authenticator PAE on an Interface

You can create or remove the 802.1X authenticator port access entity (PAE) instance on an interface.

Before you begin

Enable the 802.1X feature.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **show dot1x interface ethernet** *slot/port*
3. **interface ethernet** *slot/port*
4. **[no] dot1x pae authenticator**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **show dot1x interface ethernet** *slot/port* **Example:** `switch# show do1x interface ethernet 2/1`	Displays the 802.1X configuration on the interface.

Step 3	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 4	[**no**] **dot1x pae authenticator** **Example:** `switch(config-if)# dot1x pae authenticator`	Creates an authenticator PAE instance on the interface. Use the **no** form to remove the PAE instance from the interface. **Note** If an authenticator PAE already exists on the interface the **dot1x pae authentication** command does not change the configuration on the interface.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

9.7.6 Enabling Periodic Reauthentication for an Interface

You can enable periodic 802.1X reauthentication on an interface and specify how often it occurs. If you do not specify a time period before enabling reauthentication, the number of seconds between reauthentication defaults to the global value.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*

3. **dot1x re-authentication**
4. (Optional) **dot1x timeout re-authperiod** *seconds*
5. **exit**
6. (Optional) **show dot1x all**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x re-authentication** **Example:** `switch(config-if)# dot1x re-authentication`	Enables periodic reauthentication of the supplicants connected to the interface. By default, periodic authentication is disabled.
Step 4	(Optional) **dot1x timeout re-authperiod** *seconds* **Example:** `switch(config-if)# dot1x timeout re-authperiod 3300`	Sets the number of seconds between reauthentication attempts. The default is 3600 seconds. The range is from 1 to 65535. **Note** This command affects the behavior of the Inspur INOS device only if you enable periodic reauthentication on the interface.
Step 5	**exit** **Example:** `switch(config-if)# exit switch(config)#`	Exits configuration mode.
Step 6	(Optional) **show dot1x all** **Example:** `switch(config)# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the 802.1X Feature.
Manually Reauthenticating Supplicants .

9.7.7 Manually ReauthenticatingSupplicants

You can manually reauthenticate the supplicants for the entire Inspur INOS device or for an interface.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **dot1x re-authenticate** [**interface** *slot*/*port*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**dot1x re-authenticate** [**interface** *slot*/*port*] **Example:** `switch# dot1x re-authenticate interface 2/1`	Reauthenticates the supplicants on the Inspur INOS device or on an interface.

Related Topics

Enabling the 802.1X Feature.
Enabling Periodic Reauthentication for an Interface.

9.7.8 Manually Initializing 802.1X Authentication

You can manually initialize the authentication for all supplicants on a Inspur INOS device or for a specific interface.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **dot1x initialize** [**interface ethernet** *slot*/*port*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**dot1x initialize** [**interface ethernet** *slot*/*port*] **Example:** `switch# dot1x initialize interface ethernet 2/1`	Initializes 802.1X authentication on the Inspur INOS device or on a specified interface.

9.7.9 Changing 802.1X Authentication Timers for an Interface

You can change the following 802.1X authentication timers on the Inspur INOS device interfaces:

Quiet-period timer

When the Inspur INOS device cannot authenticate the supplicant, the switch remains idle for a set period of time and then tries again. The quiet-period timer value determines the idle period. An authentication failure might occur because the supplicant provided an invalid password. You can provide a faster response time to the user by entering a smaller number than the default. The default is the value of the global quiet period timer. The range is from 1 to 65535 seconds.

Rate-limit timer

The rate-limit period throttles EAPOL-Start packets from supplicants that are sending too many EAPOL-Start packets. The authenticator ignores EAPOL-Start packets from supplicants that have successfully authenticated for the rate-limit period duration. The default value is 0 seconds and the authenticator processes all EAPOL-Start packets. The range is from 1 to 65535 seconds.

Switch-to-authentication-server retransmission timer for Layer 4 packets

The authentication server notifies the switch each time that it receives a Layer 4 packet. If the switch does not receive a notification after sending a packet, the Inspur INOS device waits a set period of time and then retransmits the packet. The default is 30 seconds. The range is from 1 to 65535 seconds.

Switch-to-supplicant retransmission timer for EAP response frames

The supplicant responds to the EAP-request/identity frame from the Inspur INOS device with an EAP-response/identity frame. If the Inspur INOS device does not receive this response, it waits a set period of time (known as the retransmission time) and then retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.

Switch-to-supplicant retransmission timer for EAP request frames

The supplicant notifies the Inspur INOS device it that received the EAP request frame. If the authenticator does not receive this notification, it waits a set period of time and then retransmits the frame. The default is the value of the global retransmission period timer. The range is from 1 to 65535 seconds.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*
3. (Optional) **dot1x timeout quiet-period** *seconds*
4. (Optional) **dot1x timeout ratelimit-period** *seconds*
5. (Optional) **dot1x timeout server-timeout** *seconds*
6. (Optional) **dot1x timeout supp-timeout** *seconds*
7. (Optional) **dot1x timeout tx-period** *seconds*
8. **exit**
9. (Optional) **show dot1x all**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.
Step 2	**interface ethernet** *slot*/*port* **Example:** `switch(config)# interface ethernet 2/1 switch(config-if)`	Selects the interface to configure and enters interface configuration mode.
Step 3	(Optional) **dot1x timeout quiet-period** *seconds* **Example:** `switch(config-if)# dot1x timeout quiet-period 25`	Sets the number of seconds that the authenticator waits for a response to an EAP-request/identity frame from the supplicant before retransmitting the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.

Step 4	(Optional) **dot1x timeout ratelimit-period** *seconds* **Example:** `switch(config-if)# dot1x timeout ratelimit-period 10`	Sets the number of seconds that the authenticator ignores EAPOL-Start packets from supplicants that have successfully authenticated. The default value is 0 seconds. The range is from 1 to 65535 seconds.
Step 5	(Optional) **dot1x timeout server-timeout** *seconds* **Example:** `switch(config-if)# dot1x timeout server-timeout 60`	Sets the number of seconds that the Inspur INOS device waits before retransmitting a packet to the authentication server. The default is 30 seconds. The range is from 1 to 65535 seconds.
Step 6	(Optional) **dot1x timeout supp-timeout** *seconds* **Example:** `switch(config-if)# dot1x timeout supp-timeout 20`	Sets the number of seconds that the Inspur INOS device waits for the supplicant to respond to an EAP request frame before the Inspur INOS device retransmits the frame. The default is 30 seconds. The range is from 1 to 65535 seconds.
Step 7	(Optional) **dot1x timeout tx-period** *seconds* **Example:** `switch(config-if)# dot1x timeout tx-period 40`	Sets the number of seconds between the retransmission of EAP request frames when the supplicant does not send notification that it received the request. The default is the global number of seconds set for all interfaces. The range is from 1 to 65535 seconds.
Step 8	**exit** **Example:** `switch(config)# exit switch#`	Exits configuration mode.
Step 9	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays the 802.1X configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the 802.1X Feature.

9.7.10 Enabling Single Host or Multiple Hosts Mode

You can enable single host or multiple hosts mode on an interface.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **dot1x host-mode {multi-host | single-host}**
4. **exit**

5. (Optional) **show dot1x all**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x host-mode** {**multi-host** | **single-host**} **Example:** `switch(config-if)# dot1x host-mode multi-host`	Configures the host mode. The default is single-host. **Note** Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits configuration mode.

Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the 802.1X Feature.

9.7.11 Enabling MAC Authentication Bypass

You can enable MAC authentication bypass on an interface that has no supplicant connected.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*
3. **dot1x mac-auth-bypass [eap]**
4. **exit**
5. (Optional) **show dot1x all**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x mac-auth-bypass** [**eap**] **Example:** `switch(config-if)# dot1x mac-auth-bypass`	Enables MAC authentication bypass. The default is bypass disabled. Use the **eap** keyword to configure the Inspur INOS device to use EAP for authorization.
Step 4	**exit** **Example:** `switch(config-if)# exit switch(config)#`	Exits configuration mode.
Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the 802.1X Feature.

9.7.12 Disabling 802.1X Authentication on the Inspur INOS Device

You can disable 802.1X authentication on the Inspur INOS device. By default, the Inspur INOS software enables 802.1X authentication after you enable the 802.1X feature. However, when you disable the 802.1X feature, the configuration is removed from the Inspur INOS device. The Inspur INOS software allows you to disable 802.1X authentication without losing the 802.1X configuration.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **no dot1x system-auth-control**
3. **exit**
4. (Optional) **show dot1x**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no dot1x system-auth-control** **Example:** `switch(config)# no dot1x system-auth-control`	Disables 802.1X authentication on the Inspur INOS device. The default is enabled. **Note** Use the **dot1x system-auth-control** command to enable 802.1X authentication on the Inspur INOS device.
Step 3	**exit** **Example:** `switch(config)# exit switch#`	Exits configuration mode.
Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X feature status.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the 802.1X Feature.
Controlling 802.1X Authentication on an Interface.

9.7.13 Disabling the 802.1X Feature

You can disable the 802.1X feature on the Inspur INOS device.

When you disable 802.1X, all related configurations are automatically discarded. The Inspur INOS software creates an automatic checkpoint that you can use if you reenable 802.1X and want to recover the configuration. For more information, see the *Inspur INOS System Management Configuration Guide* for your platform.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **no feature dot1x**
3. **exit**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose

	Example:
	`switch# configure terminal`
	`switch(config)#`	

Step 2	**no feature dot1x** **Example:** `switch(config)# no feature dot1x`	Disables 802.1X. **Caution** Disabling the 802.1X feature removes all 802.1X configuration.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the 802.1X Feature.
Disabling 802.1X Authentication on the Inspur INOS Device.

9.7.14 Resetting the 802.1X Interface Configuration to the Default Values

You can reset the 802.1X configuration for an interface to the default values.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*
3. **dot1x default**
4. **exit**
5. (Optional) **show dot1x all**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x default**	Reverts to the 802.1X configuration default values

	Command or Action	Purpose
	Example: `switch(config-if)# dot1x default`	for the interface.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits configuration mode.

Step 5	(Optional) **show dot1x all** **Example:** `switch(config)# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the 802.1X Feature.

9.7.15 SettingtheMaximumAuthenticator-to-SupplicantFrameRetransmissionRetry Countforan Interface

You can set the maximum number of times that the Inspur INOS device retransmits authentication requests to the supplicant on an interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **dot1x max-req** *count*
4. **exit**
5. (Optional) **show dot1x all**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

	Command or Action	Purpose
Step 3	**dot1x max-req** *count* **Example:** `switch(config-if)# dot1x max-req 3`	Changes the maximum authorization request retry count. The default is 2 times and the range is from 1 to 10. **Note** Make sure that the **dot1x port-control** interface configuration command is set to **auto** for the specified interface.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits interface configuration mode.

Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the 802.1X Feature.

9.7.16 Enabling RADIUS Accounting for 802.1X Authentication

You can enable RADIUS accounting for the 802.1X authentication activity.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **dot1x radius-accounting**
3. **exit**
4. (Optional) **show dot1x**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**dot1x radius-accounting** **Example:** `switch(config)# dot1x radius-accounting`	Enables RADIUS accounting for 802.1X. The default is disabled.

	Command or Action	Purpose

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show dot1x** **Example:** `switch# show dot1x`	Displays the 802.1X configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the 802.1X Feature.

9.7.17 Configuring AAA Accounting Methods for 802.1X

You can enable AAA accounting methods for the 802.1X feature.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **aaa accounting dot1x default group** *group-list*
3. **exit**
4. (Optional) **show aaa accounting**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**aaa accounting dot1x default group** *group-list*	Configures AAA accounting for 802.1X. The default is disabled. The *group-list* argument consists of a space-delimited list of group names. The group names are the following: • **radius**—For all configured RADIUS servers. • *named-group*—Any configured RADIUS server group name.
Step 3	**exit**	Exits configuration mode.
Step 4	(Optional) **show aaa accounting**	Displays the AAA accounting configuration.
Step 5	(Optional) **copy running-config startup-config**	Copies the running configuration to the

	Command or Action	Purpose
		startup configuration.

Example

This example shows how to enable the 802.1x feature:

```
switch# configure terminal
switch(config)# aaa accounting dot1x default group radius
switch(config)# exit
switch# show aaa accounting
switch# copy running-config startup-config
```

Related Topics

Enabling the 802.1X Feature.

9.7.18 Setting the Maximum Reauthentication Retry Count on an Interface

You can set the maximum number of times that the Inspur INOS device retransmits reauthentication requests to the supplicant on an interface before the session times out. The default is 2 times and the range is from 1 to 10.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **dot1x max-reauth-req** *retry-count*
4. **exit**
5. (Optional) **show dot1x all**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the interface to configure and enters interface configuration mode.

Step 3	**dot1x max-reauth-req** *retry-count* **Example:** `switch(config-if)# dot1x max-reauth-req 3`	Changes the maximum reauthentication request retry count. The default is 2 times and the range is from 1 to 10.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits interface configuration mode.

	Command or Action	Purpose
Step 5	(Optional) **show dot1x all** **Example:** `switch# show dot1x all`	Displays all 802.1X feature status and configuration information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the 802.1X Feature.

9.8 Verifying the 802.1X Configuration

To display 802.1X information, perform one of the following tasks:

Command	Purpose
show dot1x	Displays the 802.1X feature status.
show dot1x all [**details** | **statistics** | **summary**]	Displays all 802.1X feature status and configuration information.
show dot1x interface ethernet *slot*/*port* [**details** | **statistics** | **summary**]	Displays the 802.1X feature status and configuration information for an Ethernet interface.
show running-config dot1x [**all**]	Displays the 802.1X feature configuration in the running configuration.
show startup-config dot1x	Displays the 802.1X feature configuration in the startup configuration.

For detailed information about the fields in the output from these commands, see the *Inspur INOS Security Command Reference* for your platform.

9.9 Monitoring 802.1X

You can display the statistics that the Inspur INOS device maintains for the 802.1X activity.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **show dot1x** {**all** | **interface ethernet** *slot*/*port*} **statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show dot1x** {**all** | **interface ethernet** *slot*/*port*} **statistics** **Example:**	Displays the 802.1X statistics.

	switch# show dot1x all statistics	

Related Topics

Enabling the 802.1X Feature.

9.10 Configuration Example for 802.1X

The following example shows how to configure 802.1X for an access port:

```
feature dot1x
aaa authentication dot1x default group rad2
interface Ethernet2/1
dot1x pae-authenticator
dot1x port-control auto
```

The following example shows how to configure 802.1X for a trunk port:

```
feature dot1x
aaa authentication dot1x default group rad2
interface Ethernet2/1
dot1x pae-authenticator dot1x
port-control auto dot1x host-
mode multi-host
```

9.11 Additional References for 802.1X

This section includes additional information related to implementing 802.1X.

Related Documents

Related Topic	Document Title
Inspur INOS Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*
VRF configuration	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*

Standards

Standards	Title
IEEE Std 802.1X- 2004 (Revision of IEEE Std 802.1X-2001)	*802.1X IEEE Standard for Local and Metropolitan Area Networks Port-Based Network Access Control*
RFC 2284	*PPP Extensible Authentication Protocol (EAP)*
RFC 3580	*IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines*

MIBs

MIBs	MIBs Link
• IEEE8021-PAE-MIB	-

9.12 Feature History for 802.1X

This table lists the release history for this feature:

Table 18 : Feature History for 802.1X

Feature Name	Releases	Feature Information
802.1X	8.2(3)	No change from Release 8.2(3).

CHAPTER 10 Configuring NAC

This chapter describes how to configure Network Admission Control (NAC) on Inspur INOS devices. This chapter includes the following sections:

- Finding Feature Information
- Information About NAC
- Virtualization Support for NAC
- Licensing Requirements for NAC
- Prerequisites for NAC
- NAC Guidelines and Limitations
- Default Settings for NAC
- Configuring NAC
- Verifying the NAC Configuration
- Configuration Example for NAC
- Additional References for NAC

10.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

10.2 Information About NAC

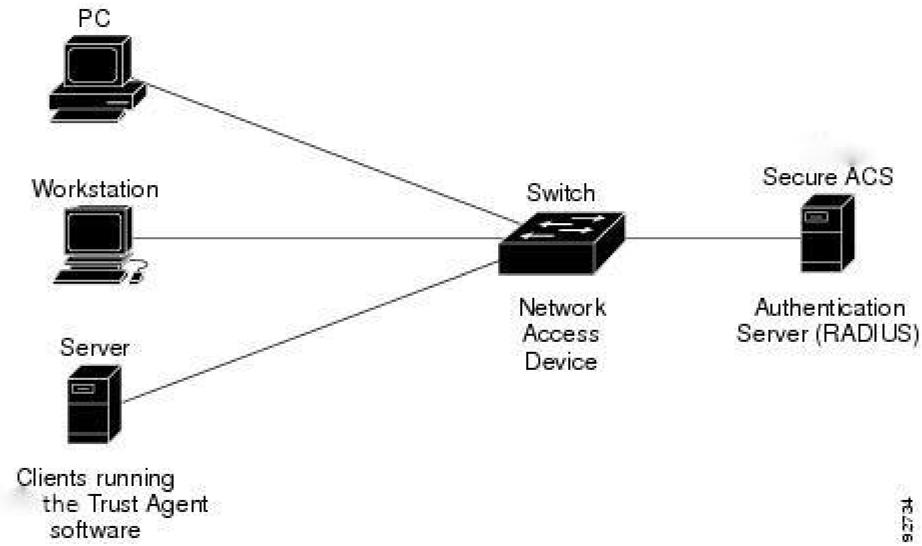
NAC allows you to check endpoint devices for security compliancy and vulnerability before these devices are allowed access to the network. This security compliancy check is referred to as *posture validation*. Posture validation allows you to prevent the spread of worms, viruses, and other rogue applications across the network.

NAC validates that the posture or state of endpoint devices complies with security policies before the devices can access protected areas of the network. For devices that comply with the security policies, NAC allows access to protected services in the network. For devices that do not comply with security policies, NAC allows access to the network only for remediation, when the posture of the device is checked again.

10.2.1 NAC Device Roles

NAC assigns roles to the devices in the network.

Figure 4 : Posture Validation Devices



This figure shows an example of a network with the NAC device roles.

NAC supports the following roles for network devices:

Endpoint device

Systems or clients on the network such as a PC, workstation, or server that is connected to a Inspur INOS device access port through a direct connection. The endpoint device, which is running the Inspur Trust Agent software, requests access to the LAN and switch services and responds to requests from the switch. Endpoint devices are potential sources of virus infections, and NAC must validate their antivirus statuses before granting network access.

Network access device (NAD)

Inspur INOS device that provides validation services and policy enforcement at the network edge and controls the physical access to the network based on the access policy of the client. The NAD relays Extensible Authentication Protocol (EAP) messages between the endpoints and the authentication server.

The NAD queries for posture credentials whenever it detects a new connection to the network. When the endpoint device has a posture agent (PA) installed, the NAD performs an in-band posture validation. The NAD acts as a relay agent between the endpoint device and AAA server for all messages in the posture validation exchange. If the NAD does not find a PA, the NAD performs an out-of-band posture validation through an audit server.

The NAD queries for posture credentials whenever it detects a new connection to the network. When the endpoint device has a posture agent (PA) installed, the NAD performs an in-band posture validation.

The NAD acts as a relay agent between the endpoint device and AAA server for all messages in the posture validation exchange. If the NAD does not find a PA, the NAD performs an out-of-band posture validation through an audit server.

The NAD controls which hosts have access to network destinations through that device based on a network access profile received from the AAA server once the posture validation exchange completes (whether in-band or out-of-band). The access profile can be one of the following forms:

- VLAN or private VLAN.
- Access control lists (ACLs) determine what type of traffic for which destinations are reachable for this host in addition to any default access that is provided to all hosts independent of the NAC process (for example, access to the Dynamic Host Configuration Protocol [DHCP] server, remediation server, audit server).

The NAD triggers the posture validation process at the following times:

- When a new session starts.
- When the revalidation timer expires.
- When you enter a system administrator command.
- When the posture agent indicates that the posture has changed (only for an endpoint device with a posture agent).

For Inspur INOS devices, the encapsulation information in the Extensible Authentication Protocol (EAP) messages is based on the User Datagram Protocol (UDP). When using UDP, the Inspur INOS device uses EAP over UDP (EAPoUDP or EoU) frames.

Authentication server

Server that performs the actual validation of the client. The authentication server validates the antivirus status of the client, determines the access policy, and notifies the NAD if the client is authorized to access the LAN and NAD services. Because the NAD acts as the proxy, the EAP message exchange between the NAD and authentication server is transparent to the NAD.

The Inspur INOS device supports the Inspur Secure Access Control Server (ACS) Version 4.0 or later with RADIUS, authentication, authorization, and accounting (AAA), and EAP extensions.

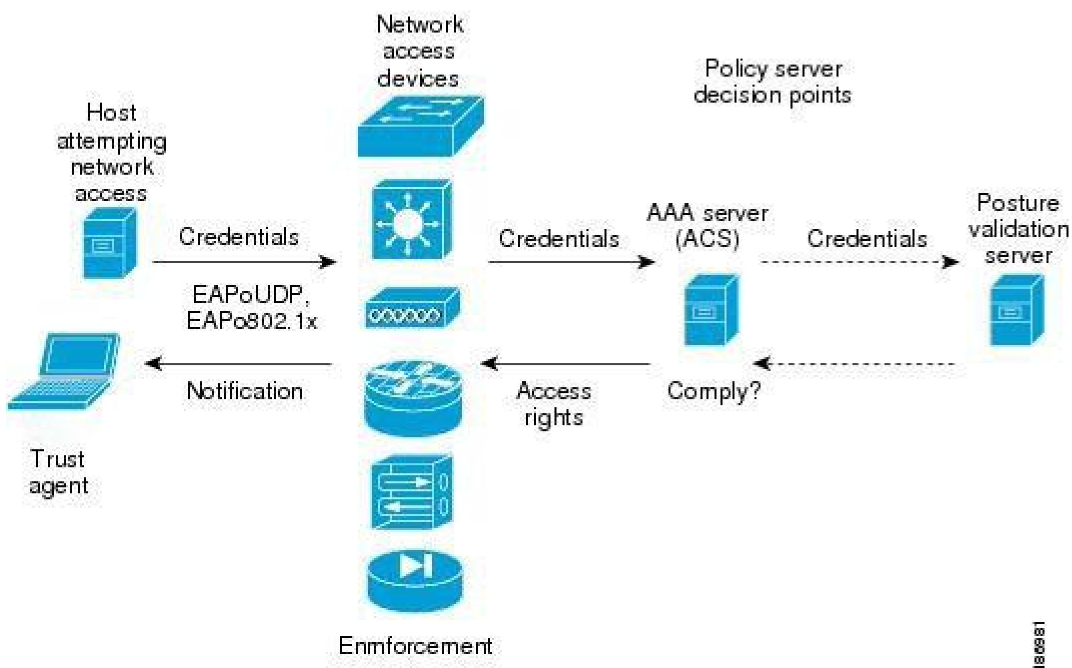
Posture validation server

Third-party server that acts as an application-specific policy decision point in NAC for authorizing a set of posture credentials against a set of policy rules. The posture validation server receives requests from an authentication server.

10.2.2 NAC Posture Validation

Posture validation occurs when a NAC-enabled NAD detects an endpoint device that is attempting to connect or use its network resources. When the NAD detects a new endpoint device, it requests the network access profile for the endpoint device from an AAA server (such as the Inspur Secure ACS).

Figure 5： NAC Endpoint Device Posture Validation



This figure shows the NAC endpoint device posture validation process.

The AAA server determines if the endpoint device has a posture agent installed. If the endpoint device has a posture agent (such as the Inspur Trust Agent), the AAA server requests the endpoint device for posture information via the NAD. The endpoint device responds to the AAA server with a set of posture credentials. The AAA server then validates the posture information locally or delegates the posture validation decisions to one or more external posture validation servers.

If the endpoint device does not have a posture agent, the AAA server may request an audit server to collect posture information from the device through other means (for example, fingerprinting and port scanning). The AAA server also asks the audit server to validate that information and return a posture validation decision.

The AAA server aggregates the posture validation results from these sources and makes an authorization decision that is based on whether the endpoint device complies with the network policy. The AAA server determines the network access profile for the endpoint device and sends the profile to the NAD for enforcement of the endpoint device authorization.

The examination of endpoint device credentials by the AAA server can result in one or more application posture tokens (APTs). An APT represents a compliance check for a given vendor's application. The AAA server aggregates all APTs from the posture validation servers into a single system posture token (SPT) that represents the overall compliance of the endpoint device. The value SPT is based on the worst APT from the set of APTs. Both APTs and SPTs are represented using the following predefined tokens:

Healthy

The endpoint device complies with the posture policy so no restrictions are placed on this device.

Checkup

The endpoint device is within policy but does not have the latest software; an update is recommended.

Transition

The endpoint device is in the process of having its posture checked and is given interim access pending a result from a complete posture validation. A transition result may occur when a host is booting and complete posture information is not available, or when complete audit results are not available.

Quarantine

The endpoint device is out of compliance and must be restricted to a quarantine network for remediation. This device is not actively placing a threat on other endpoint devices but is vulnerable to attack or infection and must be updated as soon as possible.

Infected

The endpoint device is an active threat to other endpoint devices; network access must be severely restricted and the endpoint device must be placed into remediation or denied all network access to the endpoint device.

Unknown

The AAA server cannot determine the posture credentials of the endpoint device. You need to determine the integrity of the endpoint device so that proper posture credentials can be attained and assessed for network access authorization.

10.2.3 IP Device Tracking

The IP device tracking allows endpoint devices to remain connected to the network if the AAA server is not available. Typical deployments of NAC use Inspur Secure ACS to validate the client posture and to pass policies back to the NAD.

IP device tracking provides the following benefits:

- While AAA is unavailable, the endpoint device still has connectivity to the network, although it may be restricted.

- When the AAA server is available again, a user can be revalidated and the user's policies can be downloaded from the ACS.

10.2.4 NAC LPIP

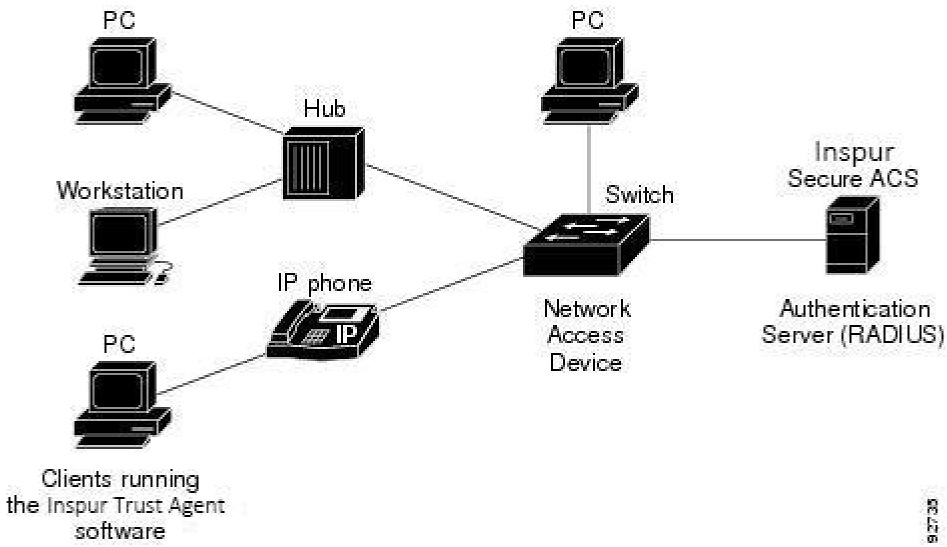
NAC LAN port IP (LPIP) validation uses the Layer 3 transport EAPoUDP to carry posture validation information. LPIP validation has the following characteristics:

- Operates only on Layer 2 ports and cannot operate on Layer 3 ports.
- Subjects all hosts sending IP traffic on the port to posture validation.

LPIP validation triggers admission control by snooping on DHCP messages or Address Resolution Protocol (ARP) messages rather than intercepting IP packets on the data path. LPIP validation performs policy enforcement using access control lists (ACLs).

Figure 6：Network Using LPIP Validation

This figure shows the LPIP validation process for a single host connected to a NAD port or multiple hosts on the same NAD port.



When you enable LPIP validation, EAPoUDP only supports IPv4 traffic. The NAD checks the antivirus status of the endpoint devices or clients and enforces access control policies.

Posture Validation

When you enable LPIP validation on a port connected to one or more endpoint devices, the Inspur INOS device uses DHCP snooping and ARP snooping to identify connected hosts. The Inspur INOS device initiates posture validation after receiving an ARP packet or creating a DHCP snooping binding entry. ARP snooping is the default method to detect connected hosts. If you want the NAD to detect hosts when a DHCP snooping binding entry is created, you must enable DHCP snooping.

Admission Triggers

ARP snooping allows LPIP validation to detect hosts with either dynamically acquired or statically configured IP addresses. When the NAD receives an ARP packet from an unknown host, it triggers posture validation. If you have enabled DHCP snooping on the interface, the creation of a DHCP binding entry on the NAD triggers posture validation. DHCP snooping provides a slightly faster response time because DHCP packets are exchanged prior to sending ARP requests. Both ARP snooping and DHCP snooping can trigger posture validation on the same host. In this case, the trigger initiated by the creation of a DHCP snooping binding takes precedence over ARP snooping.

Posture Validation Methods

After posture validation is triggered for a host, you can use one of two possible methods to determine the policy to be applied for the host:

- Exception lists
- EAPoUDP

Exception Lists

An exception list contains local profile and policy configurations. Use the identity profile to statically authorize or validate devices based on the IP address and MAC address. You can associate an identity profile with a local policy that specifies the access control attributes.

Using an exception list, you can bypass posture validation for specific endpoint devices and apply a statically configured policy. After posture validation is triggered, the NAD checks for the host information in the exception list. If a match is found in the exception list, the NAD applies the configured policy for the endpoint device.

EAPoUDP

If an endpoint device does not match the exception list, the NAD sends an EAPoUDP packet to initiate posture validation. While posture validation occurs, the NAD enforces the default access policy. After the NAD sends an EAPoUDP message to the host and the host responds to the antivirus condition request, the NAD forwards the EAPoUDP response to the Inspur Secure ACS. If the NAD does not receive a response from the host after the specified number of attempts, the NAD classifies the host as nonresponsive. After the ACS validates the credentials, the authentication server returns an Access-Accept or Access-Reject message to the NAD. The NAD updates the EAPoUDP session table and enforces the access limitations, which segments and quarantines the poorly postured endpoint device or denies network access.

For an Access-Accept message, the NAD applies the enforcement policy that contains the policy-based ACL (PACL) name and starts the EAP revalidation and status query timers.

For an Access-Reject message, the NAD removes any enforcement policy for the host and puts the endpoint device into the Held state for a configured period of time (Hold timer). After the Hold timer expires, the NAD revalidates the endpoint device.

Policy Enforcement Using ACLs

LPIP validation uses PACLs for policy enforcement.

The NAD applies the PACL when the posture validation fails (the AAA server sends an Access-Reject message). The default policy is to use the active MAC ACL applied to the port (also called a port ACL [PACL]). The active MAC ACL could either be a statically configured PACL or an AAA server-specified PACL based on 802.1X authentication.

The PACL defines a group that expands to a list of endpoint device IP addresses. The PACLs usually contain the endpoint device IP addresses. Once the NAD classifies an endpoint device using a particular group, the

NAD adds the IP address that corresponds to the endpoint device to the appropriate group. The result is that the policy is applied to the endpoint device.

When you configure LPIP validation for an NAD port, you must also configure a default PACL on that NAD port. In addition, you should apply the default ACL to the IP traffic for hosts that have not completed posture validation.

If you configure the default ACL on the NAD and the Inspur Secure ACS sends a host access policy to the NAD, the NAD applies the policy to that traffic from the host that is connected to a NAD port. If the policy applies to the traffic, the NAD forwards the traffic. If the policy does not apply, the NAD applies the default ACL. However, if the NAD gets an endpoint device access policy from the Inspur Secure ACS but the default ACL is not configured, the LPIP validation configuration does not take effect.

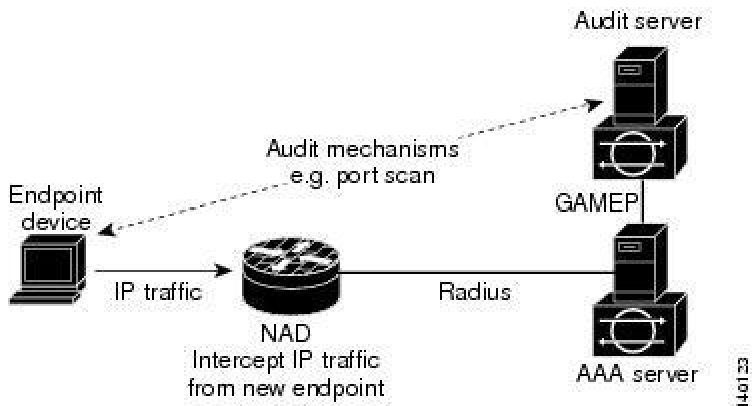
Audit Servers and Nonresponsive Hosts

Endpoint devices that do not run a posture agent (Inspur Trust Agent) cannot provide credentials when challenged by

NADs. These devices are described as *agentless* or *nonresponsive*.

The NAC architecture supports audit servers to validate agentless endpoint devices. An audit server is a third-party server that can probe, scan, and determine security compliance of a host without needing a posture again on the endpoint device. The result of the audit server examination can influence the access servers to make network access policy decisions specific to the endpoint device instead of enforcing a common restrictive policy for all nonresponsive endpoint devices. You can build more robust host audit and examination functionality by integrating any third-party audit operations into the NAC architecture.

Figure 7：NAC Device Roles



This figure shows how audit servers fit into the typical topology.

NAC assumes that the audit server can be reached so that the endpoint device can communicate with it. When an endpoint device makes network access through the NAD configured for posture validation, the network access device eventually requests the AAA server (Inspur Secure ACS) for an access policy to be enforced for the host. The AAA server can be configured to trigger a scan of the host with an external audit server. The audit server scan occurs asynchronously and takes several seconds to complete. During the scan, the AAA server conveys a minimal restrictive security policy to NAD for enforcement along with a short poll timer(session-timeout). The NAD polls the AAA sever at the specified timer interval until the result is available from the audit server. After the AAA server receives the audit result, it computes an access policy based on the audit result and sends it to the NAD for enforcement on its next request.

NAC Timers

This section describes the NAC timers.

Hold Timer

The hold timer prevents a new EAPoUDP session from immediately starting after the previous attempt to validate that the session fails. NAC uses this time only when the Inspur Secure ACS sends an Accept-Reject message to the NAD. The default value of the hold timer is 180 seconds (3 minutes).

An EAPoUDP session might not be validated when the posture validation of the host fails, a session timer expires, or the NAD or Inspur Secure ACS receives invalid messages. If the NAD or authentication server continuously receives invalid messages, a malicious user might be trying to cause a denial-of-service attack.

AAA Timer

The AAA timer controls the amount of time that the NAD waits for a response from the AAA server before resending a request during posture validation. The default value of the retransmission timer is 60 seconds.

Retransmit Timer

Revalidation Timer

The retransmit timer controls the amount of time that the NAD waits for a response from the client before resending a request during posture validation. The default value of the retransmission timer is 3 seconds.

The revalidation timer controls the amount of time that the NAD applies a NAC policy to an endpoint device that used EAPoUDP messages during posture validation. The timer starts after the initial posture validation completes. The timer resets when the host is revalidated. The default value of the revalidation timer is 36000 seconds (10 hours).

The Inspur INOS software bases the revalidation timer operation on the Session-Timeout RADIUS attribute (Attribute[27]) and the Termination-Action RADIUS-REQUEST attribute (Attribute[29]) in the Access-Accept message from the AAA server (Inspur Secure ACS). If the NAD receives the Session-Timeout value, this value overrides the revalidation timer value on the NAD.

If the revalidation timer expires, the NAD action depends on one of these values of the Termination-Action attribute:

- If the value of the Termination-Action RADIUS attribute is the default, the session ends.
- If the NAD receives a value for the Termination-Action attribute other than the default, the EAPoUDP session and the current access policy remain in effect during posture revalidation.
- If the value of the Termination-Action attribute is RADIUS, the NAD revalidates the client.
- If the packet from the server does not include the Termination-Action attribute, the EAPoUDP session ends.

Status-Query Timer

The status-query timer controls the amount of time that the NAD waits before verifying that the previously validated client is present and that its posture has not changed. Only clients that were authenticated with EAPoUDP messages use this timer, which starts after the client is initially validated. The default value of the status-query timer is 300 seconds (5 minutes).

The timer resets when the host is reauthenticated. When the timer expires, the NAD checks the host posture validation by sending a Status-Query message to the host. If the host sends a message to the NAD that the posture has changed, the NAD revalidates the posture of the host.

NAC Posture Validation and Redundant Supervisor Modules

When a switchover occurs, the Inspur INOS device maintains information about the endpoint devices and the current PACL application but loses the current state of each EAPoUDP session. The Inspur INOS device removes the current PACL application and restarts posture validation.

10.2.5 LPIP Validation and Other Security Features

This section describes how LPIP validation interacts with other security features on the Inspur INOS device.

802.1X

If you configure both 802.1X and LPIP on a port, the traffic that does not pass the 802.1X-authenticated source MAC check does not trigger posture validation. When you configure 802.1X on a port, the port cannot transmit or receive traffic (other than EAP over LAN [EAPOL] frames) until the attached host is authenticated via 802.1X. This mechanism ensures that the IP traffic from the host does not trigger posture validation before it is authenticated.

Port Security

The NAD checks the source MAC against the port security MACs and drops the endpoint device if the check fails. The NAD allows posture validation only on port security-validated MAC addresses. If a port security violation occurs and results in a port shutdown, the Inspur INOS software removes the LPIP state of the port.

DHCP Snooping

Posture validation does not occur until after a DHCP creates a binding entry. When you enable DHCP snooping and LPIP, the Inspur INOS software triggers posture validation for a host when DHCP creates a binding entry for the host using DHCP to acquire IP address.

Dynamic ARP Inspection

If you enable LPIP validation on the interface, posture validation is triggered only if the packet passes the dynamic ARP inspection (DAI) check. If you do not enable DAI, then all ARP packets (with valid MAC/IP pairs) will trigger posture validation.

IP Source Guard

IP Source Guard is a per-interface traffic filter that permits IP traffic only when the IP address and MAC address of each packet matches one of two sources of IP and MAC address bindings:

- Entries in the DHCP snooping binding table.
- Static IP source entries that you configure.

Filtering on trusted IP and MAC address bindings helps prevent attacks that rely on spoofing the IP address of a valid host. To circumvent IP Source Guard, an attacker would have to spoof both the IP address and the MAC address of a valid host.

Related Topics

Configuring IP Source Guard.

Posture Host-Specific ACEs

The Inspur INOS software drops the packet if the packet matches the deny condition and skips the active PACL if a packet matches a permit condition. If no implicit deny exists at the end of the ACEs and no match occurs, the Inspur INOS software checks the packet against the active PACL.

Active PACLs

The active PACL is either a statically configured PACL or an AAA server-specified PACL that is based on 802.1X authentication. The packet is dropped if it matches any deny condition and moves to the next step if it matches a permit condition.

VACLs

The Inspur INOS software drops any packet that matches a deny condition.

10.3 Virtualization Support for NAC

NAC configuration and operation are local to the virtual device context (VDC).

10.4 Licensing Requirements for NAC

The following table shows the licensing requirements for this feature:

Product	**License Requirement**
Inspur INOS	NAC requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

10.5 Prerequisites for NAC

NAC has the following prerequisites:
- Ensure that a Layer 3 route exists between the NAD and each endpoint device.

10.6 NAC Guidelines andLimitations

NAC has the following guidelines and limitations:
- EAPoUDP bypass and AAA down policy are not supported.
- NAC uses only RADIUS for authentication.

10.6.1 LPIP Limitations

LPIP validation has the following limitations:
- LPIP validation is allowed only on access ports.
- You cannot enable LPIP validation on trunk ports or port channels.
- LPIP validation is not allowed on ports that are SPAN destinations.
- LPIP validation is not allowed on ports that are part of a private VLAN.
- LPIP validation does not support IPv6.
- LPIP validation is allowed only for endpoint devices directly connected to the NAD.
- You cannot use LPIP validation unless you have a Layer 3 route between the NAD and the endpoint device.

10.7 Default Settings for NAC

This table lists the default settings for NAC parameters.

Table 19： Default NAC Parameter Settings

Parameters	Default
EAPoUDP	Disabled.
EAP UDP port number	21862 (0x5566).
Clientless hosts allowed	Disabled.
Automatic periodic revalidation	Enabled.
Revalidation timeout interval	36000 seconds (10 hours).
Retransmit timeout interval	3 seconds.
Status query timeout interval	300 seconds (5 minutes).
Hold timeout interval	180 seconds (3 minutes).
AAA timeout interval	60 seconds (1 minute).
Maximum retries	3.
EAPoUDP rate limit maximum	20 simultaneous sessions.
EAPoUDP logging	Disabled.
IP device tracking	Enabled.

10.8 Configuring NAC

This section describes how to configure NAC.

10.8.1 Process for Configuring NAC

Follow these steps to configure NAC:

SUMMARY STEPS

1. Enable EAPoUDP.
2. Configure the connection to the AAA server.
3. Apply PACLs to the interfaces connected to endpoint devices.
4. Enable NAC on the interfaces connected to the endpoint devices.

DETAILED STEPS

Step 1 Enable EAPoUDP.

Step 2 Configure the connection to the AAA server.

Step 3 Apply PACLs to the interfaces connected to endpoint devices.

Step 4 Enable NAC on the interfaces connected to the endpoint devices.

Related Topics

Enabling EAPoUDP.
Enabling the Default AAA Authenication Method for EAPoUDP
Applying PACLs to Interfaces.
Enabling NAC on an Interface.

10.8.2 Enabling EAPoUDP

The Inspur INOS device relays Extensible Authentication Protocol (EAP) messages between the endpoints and the authentication server. You must enable EAP over UDP (EAPoUDP) before configuring NAC on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **feature eou**
3. **exit**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature eou** **Example:** `switch(config)# feature eou`	Enables EAPoUDP. The default is disabled.
Step 3	**exit** **Example:** `switch(config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
	`switch#`	
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

10.8.3 Enabling the Default AAA Authenication Method for EAPoUDP

You must enable the default AAA authentication method EAPoUDP.

Before you begin

Enable EAPoUDP.
Configure RADIUS or TACACS+ server groups, as needed.

SUMMARY STEPS

1. **configure terminal**
2. **aaa authentication eou default group** *group-list*
3. **exit**
4. (Optional) **show aaa authentication**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa authentication eou default group** *group-list* **Example:** `switch(config)# aaa authentication eou default group RadServer`	Configures a list of one or more RADIUS server groups as the default AAA authentication method for EAPoUDP. The *group-list* argument consists of a space-delimited list of groups. The group names are as follows: • **radius**—Uses the global pool of RADIUS servers for authentication. • *named-group*—Uses a named subset of RADIUS servers for authentication. The default setting is no method.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show aaa authentication** **Example:**	Displays the default AAA authentication methods.

	Command or Action	Purpose
	`switch# show aaa authentication`	
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP
Configuring AAA
Configuring RADIUS.

10.8.4 Applying PACLs to Interfaces

You must apply a PACL to the access interfaces on the NAD that perform LPIP posture validation if no PACL is available from the AAA server.

Before you begin

Create a MAC ACL.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **mac access-group** *access-list*
4. **exit**
5. (Optional) **show running-config interface**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Specifies the Ethernet interface and enters interface configuration mode.

Step 3	**mac access-group** *access-list* **Example:** `switch(config-if)# mac access-group acl-01`	Applies a PACL to the interface for traffic that flows in the direction specified. **Note** An interface can have only one PACL. To replace the PACL on the interface, enter this command again using the new PACL name.
Step 4	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`switch(config-if)# exit` `switch(config)#`	
Step 5	(Optional) **show running-config interface** **Example:** `switch(config)# show running-config interface`	Displays the interface PACL configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling NAC on an Interface

You must enable NAC on an interface for posture validation to occur.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **switchport**
4. **switchport mode access**
5. **nac enable**
6. **exit**
7. (Optional) **show running-config interface**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Specifies the Ethernet interface and enters interface configuration mode.

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Sets the interface as a Layer 2 switching interface. By default, all ports are Layer 3 ports.
Step 4	**switchport mode access** **Example:** `switch(config-if)# switchport mode access`	Configures the port mode as access.
Step 5	**nac enable**	Enables NAC on the interface.

	Command or Action	Purpose
	Example: `switch(config-if)# nac enable`	

Step 6	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits global configuration mode.

Step 7	(Optional) **show running-config interface** **Example:** `switch(config)# show running-config interface`	Displays the interface PACL configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.

10.8.5 Configuring Identity Policies and Identity Profile Entries

You can use the identity profile to configure exceptions to LPIP posture validation. The identity profile contains entries for the endpoint devices for which are not subject to LPIP validation. You can optionally configure an identity policy for each identity profile entry that specifies a PACL that the INOS device applies to the endpoint device. The default identity policy is the PACL for the interface.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. **configure terminal**
2. **identity policy** *policy-name*
3. **object-group** *access-list*
4. (Optional) **description "** *text* **"**
5. **exit**
6. (Optional) **show identity policy**
7. identity profile eapoudp
8. **device** {**authenticate** | **not-authenticate**} {**ip-address** *ipv4-address* [*ipv4-subnet-mask*] | **mac-address** *mac-address* [*mac-subnet-mask*]} **policy** *name*
9. **exit**
10. (Optional) **show identity profile eapoudp**
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose

	Example:
	`switch# configure terminal`
	`switch(config)#`	

Step 2	**identity policy** *policy-name*
	Example:
	`switch(config)# identity policy AccType1`
	`switch(config-id-policy)#`
		Specifies the identity policy name and enters identity policy configuration mode. You can create a maximum of 1024 identity policies. The maximum length of the name is 100 characters.

Step 3	**object-group** *access-list* **Example:** `switch(config-id-policy)# object-group maxaclx`	Specifies the IP ACL or MAC ACL for the policy.
Step 4	(Optional) **description " ** *text* **"** **Example:** `switch(config-id-policy)# description "This policy prevents endpoint device without a PA"`	Provides a description for the identity policy. The maximum length is 100 characters.

Step 5	**exit**
	Example:
	`switch(config-id-policy)# exit`
	`switch(config)#`
		Exits identity policy configuration mode.

Step 6	(Optional) **show identity policy** **Example:** `switch(config)# show identity policy`	Displays the identity policy configuration.

Step 7	**identity profile eapoudp**
	Example:
	`switch(config)# identity profile eapoudp`
	`switch(config-id-prof)#`
		Enters identity profile configuration mode for EAPoUDP.

Step 8	**device** {**authenticate** | **not-authenticate**} {**ip-address** *ipv4-address* [*ipv4-subnet-mask*] | **mac-address** *mac-address* [*mac-subnet-mask*]} **policy** *name* **Example:** `switch(config-id-prof)# device authenticate ip-address 10.10.2.2 policy AccType1`	Specifies an exception entry. The maximum number of entries is 5000.

Step 9	**exit**
	Example:
	`switch(config-id-prof)# exit`
	`switch(config)#`
		Exits identity profile configuration mode.

Step 10	(Optional) **show identity profile eapoudp** **Example:** `switch(config)# show identity profile eapoudp`	Displays the identity profile configuration.
Step 11	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	`switch# copy running-config startup-config`	

Related Topics

Enabling EAPoUDP.

10.8.6 Allowing Clientless Endpoint Devices

You can allow posture validation endpoint devices in your network that do not have a posture agent installed (clientless). The posture validation is performed by an audit server that has access to the endpoint devices.

Before you begin

Enable EAPoUDP.
Verify that the AAA server and clientless endpoint devices can access the audit server.

SUMMARY STEPS

1. **configure terminal**
2. **eou allow clientless**
3. **exit**
4. (Optional) **show eou**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**eou allow clientless** **Example:** `switch(config)# eou allow clientless`	Allows posture validation for clientless endpoint devices. The default is disabled.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show eou** **Example:** `switch# show eou`	Displays the EAPoUDP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.

10.8.7 Enabling Logging for EAPoUDP

You can enable logging for EAPoUDP event messages. EAPoUDP events include errors and status changes. The destination for these event messages is the configured syslog.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. **configure terminal**
2. **eou logging**
3. **exit**
4. (Optional) **show eou**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**eou logging** **Example:** `switch(config)# eou logging`	Enables EAPoUDP logging. The default is disabled.
Step 3	**exit** **Example:** `switch(config)# exit switch#`	Exits global configuration mode.
Step 4	(Optional) **show eou** **Example:** `switch)# show eou`	Displays the EAPoUDP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.

10.8.8 Changing the Global EAPoUDP Maximum Retry Value

You can change the global maximum number of EAPoUDP retries. The default value is three.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. **configure terminal**

2. **eou max-retry** *count*
3. **exit**
4. (Optional) **show eou**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**eou max-retry** *count* **Example:** `switch(config)# eou max-retry 2`	Changes the EAPoUDP maximum retry count. The default is 3. The range is from 1 to 3.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show eou** **Example:** `switch# show eou`	Displays the EAPoUDP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.
Changing the EAPoUDP Maximum Retry Value for an Interface.

10.8.9 Changing the EAPoUDP Maximum Retry Value for an Interface

You can change the maximum number of EAPoUDP retries for an interface. The default value is three.

Before you begin

Enable EAPoUDP.
Enable NAC on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*
3. **eou max-retry** *count*
4. **exit**
5. (Optional) **show eou**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Specifies the Ethernet interface and enters interface configuration mode.

Step 3	**eou max-retry** *count* **Example:** `switch(config-if)# eou max-retry 2`	Changes the EAPoUDP maximum retry count. The default is 3. The range is from 1 to 3.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 5	(Optional) **show eou** **Example:** `switch(config)# show eou`	Displays the EAPoUDP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.
Changing the Global EAPoUDP Maximum Retry ValueEnabling NAC on an Interface.

10.8.10 Changing the UDP Port for EAPoUDP

You can change the UDP port used by EAPoUDP. The default port is 21862.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. **configure terminal**
2. **eou port** *udp-port*
3. **exit**
4. (Optional) **show eou**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose

	Example:
	`switch# configure terminal`
	`switch(config)#`	

Step 2	**eou port** *udp-port* **Example:** `switch(config)# eou port 27180`	Changes the UDP port used by EAPoUDP. The default is 21862. The range is from 1 to 65535.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show eou** **Example:** `switch# show eou`	Displays the EAPoUDP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.

10.8.11 Configuring Rate Limiting of Simultaneous EAPoUDP Posture Validation Sessions

You can configure rate limiting to control the number of simultaneous EAPoUDP posture validations sessions. You can change the rate-limiting value that controls the maximum number of simultaneous EAPoUDP posture validation sessions. The default number is 20. Setting the number to zero (0) disables rate limiting.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. **configure terminal**
2. **eou ratelimit** *number-of-sessions*
3. **exit**
4. (Optional) **show eou**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**eou ratelimit** *number-of-sessions* **Example:** `switch(config)# eou ratelimit 15`	Configures the number of simultaneous EAPoUDP posture validation sessions. The default is 20. The range is from 0 to 200. **Note** A setting of zero (0) disables rate limiting.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show eou** **Example:** `switch# show eou`	Displays the EAPoUDP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.

10.8.12 Configuring Global Automatic Posture Revalidation

The Inspur INOS software automatically revalidates the posture of the endpoint devices for the Inspur INOS device at a configured interval. The default interval is 36,000 seconds (10 hours). You can disable revalidation or change the length of the revalidation interval.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **eou revalidate**
3. (Optional) **eou timeout revalidation** *seconds*
4. **exit**
5. (Optional) **show eou**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **eou revalidate** **Example:** `switch(config)# eou revalidate`	Enables the automatic posture validation. The default is enabled.

	Command or Action	Purpose
Step 3	(Optional) **eou timeout revalidation** *seconds* **Example:** `switch(config)# eou timeout revalidation 30000`	Changes the revalidation timer interval. The default is 36000. The range is from 5 to 86400 seconds. Use the **no eou revalidate** command to disable automatic posture validation.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 5	(Optional) **show eou** **Example:** `switch# show eou`	Displays the EAPoUDP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.
Configuring Automatic Posture Revalidation for an Interface.

10.8.13 Configuring Automatic Posture Revalidation for an Interface

The Inspur INOS software automatically revalidates the posture of the endpoint devices for the Inspur INOS device at a configured interval. The default interval is 36,000 seconds (10 hours). You can disable revalidation or change the length of the revalidation interval.

Before you begin

Enable EAPoUDP.
Enable NAC on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. (Optional) **eou revalidate**
4. (Optional) **eou timeout revalidation** *seconds*
5. **exit**
6. (Optional) **show eou**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
	`switch(config)#`	

Step 2	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Specifies the Ethernet interface and enters interface configuration mode.

Step 3	(Optional) **eou revalidate** **Example:** `switch(config-if)# eou revalidate`	Enables the automatic posture validation. The default is enabled. Use the **no eou revalidate** command to disable automatic posture validation.
Step 4	(Optional) **eou timeout revalidation** *seconds* **Example:** `switch(config-if)# eou timeout revalidation 30000`	Changes the revalidation timer interval. The default is 36000. The range is from 5 to 86400 seconds.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits global configuration mode.

Step 6	(Optional) **show eou** **Example:** `switch(config)# show eou`	Displays the EAPoUDP configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.
Configuring Global Automatic Posture Revalidation
Enabling NAC on an Interface.

10.8.14 Changing the Global EAPoUDP Timers

The Inspur INOS software supports the following global timers for EAPoUDP:

AAA

Controls the amount of time that the NAD waits for a response from the AAA server before resending a request during posture validation.

Hold period

Prevents a new EAPoUDP session from immediately starting after the previous attempt to validate that the session fails. NAC uses this time only when the Inspur Secure ACS sends an Accept-Reject message to the NAD.

Retransmit

Controls the amount of time that the NAD waits for a response from the client before resending a request during posture validation.

Revalidation

Controls the amount of time that the NAD applies a NAC policy to an endpoint device that used EAPoUDP messages during posture validation. The timer starts after the initial posture validation completes.

Status query

Controls the amount of time that the NAD waits before verifying that the previously validated client is present and that its posture has not changed. Only clients that were authenticated with EAPoUDP messages use this timer, which starts after the client is initially validated.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **eou timeout aaa** *seconds*
3. (Optional) **eou timeout hold-period** *seconds*
4. (Optional) **eou timeout retransmit** *seconds*
5. (Optional) **eou timeout revalidation** *seconds*
6. (Optional) **eou timeout status-query** *seconds*
7. **exit**
8. (Optional) **show eou**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	(Optional) **eou timeout aaa** *seconds* **Example:** `switch(config)# eou timeout aaa 30`	Changes the AAA timeout interval. The default is 60 seconds (1 minute). The range is from 0 to 60 seconds.
Step 3	(Optional) **eou timeout hold-period** *seconds* **Example:** `switch(config)# eou timeout hold-period 300`	Changes the hold period timeout interval. The default is 180 seconds (3 minutes). The range is from 60 to 86400 seconds.
Step 4	(Optional) **eou timeout retransmit** *seconds* **Example:** `switch(config)# eou timeout retransmit 10`	Changes the retransmit timeout interval. The default is 3 seconds. The range is from 1 to 60 seconds.
Step 5	(Optional) **eou timeout revalidation** *seconds* **Example:** `switch(config)# eou timeout revalidation 30000`	Changes the revalidation timer interval. The default is 36000. The range is from 5 to 86400 seconds.
Step 6	(Optional) **eou timeout status-query** *seconds* **Example:** `switch(config)# eou timeout status-query 360`	Changes the status query timeout interval. The default is 300 seconds (5 minutes). The range is from 10 to 1800 seconds.

	Command or Action	Purpose

Step 7	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 8	(Optional) **show eou** **Example:** `switch# show eou`	Displays the EAPoUDP configuration.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.
Changing the EAPoUDP Timers for an Interface
NAC Timers.

10.8.15 Changing the EAPoUDP Timers for an Interface

The Inspur INOS software supports the following timers for EAPoUDP for each interface enabled for NAC:

AAA

Controls the amount of time that the NAD waits for a response from the AAA server before resending a request during posture validation.

Hold period

Prevents a new EAPoUDP session from immediately starting after the previous attempt to validate that the session fails. NAC uses this time only when the Inspur Secure ACS sends an Accept-Reject message to the NAD.

Retransmit

Controls the amount of time that the NAD waits for a response from the client before resending a request during posture validation.

Revalidation

Controls the amount of time that the NAD applies a NAC policy to an endpoint device that used EAPoUDP messages during posture validation. The timer starts after the initial posture validation completes.

Status query

Controls the amount of time that the NAD waits before verifying that the previously validated client is present and that its posture has not changed. Only clients that were authenticated with EAPoUDP messages use this timer, which starts after the client is initially validated.

Before you begin

Enable EAPoUDP.
Enable NAC on the interface.

SUMMARY STEPS

1. **configure terminal**

2. **interface ethernet** *slot*/*port*
3. (Optional) **eou timeout aaa** *seconds*
4. (Optional) **eou timeout hold-period** *seconds*
5. (Optional) **eou timeout retransmit** *seconds*
6. (Optional) **eou timeout revalidation** *seconds*
7. (Optional) **eou timeout status-query** *seconds*
8. **exit**
9. (Optional) **show eou**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Specifies the Ethernet interface and enters interface configuration mode.

Step 3	(Optional) **eou timeout aaa** *seconds* **Example:** `switch(config-if)# eou timeout aaa 50`	Changes the AAA timeout interval. The default is 60 seconds (1 minute). The range is from 0 to 60 seconds.
Step 4	(Optional) **eou timeout hold-period** *seconds* **Example:** `switch(config-if)# eou timeout hold-period 300`	Changes the hold period timeout interval. The default is 180 seconds (3 minutes). The range is from 60 to 86400 seconds.
Step 5	(Optional) **eou timeout retransmit** *seconds* **Example:** `switch(config-if)# eou timeout retransmit 10`	Changes the retransmit timeout interval. The default is 3 seconds. The range is from 1 to 60 seconds.
Step 6	(Optional) **eou timeout revalidation** *seconds* **Example:** `switch(config-if)# eou timeout revalidation 30000`	Changes the revalidation timer interval. The default is 36000. The range is from 5 to 86400 seconds.
Step 7	(Optional) **eou timeout status-query** *seconds* **Example:** `switch(config-if)# eou timeout status-query 360`	Changes the status query timeout interval. The default is 300 seconds (5 minutes). The range is from 10 to 1800 seconds.

Step 8	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 9	(Optional) **show eou**	Displays the EAPoUDP configuration.

	Command or Action	Purpose
	Example: `switch(config)# show eou`	
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.
Changing the Global EAPoUDP Timers
NAC Timers.
Enabling NAC on an Interface.

10.8.16 Resetting the EAPoUDP Global Configuration to the Default Values

You can reset the EAPoUDP global configuration to the default values.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. **configure terminal**
2. **eou default**
3. **exit**
4. (Optional) **show eou**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal` `switch(config)#`	Enters global configuration mode.
Step 2	**eou default** **Example:** `switch(config)# eou default`	Resets the EAPoUDP configuration to the default values.
Step 3	**exit** **Example:** `switch(config)# exit` `switch#`	Exits global configuration mode.
Step 4	(Optional) **show eou** **Example:** `switch# show eou`	Displays the EAPoUDP configuration.
Step 5	(Optional) **copy running-config startup-config**	Copies the running configuration to the

	Command or Action	Purpose
	Example: `switch# copy running-config startup-config`	startup configuration.

Related Topics

Enabling EAPoUDP.
Resetting the EAPoUDP Interface Configuration to the Default Values.

10.8.17 Resetting the EAPoUDP Interface Configuration to the Default Values

You can reset the EAPoUDP configuration for an interface to the default values.

Before you begin

Enable EAPoUDP.
Enabled NAC on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*
3. **eou default**
4. **exit**
5. (Optional) **show eou interface ethernet** *slot*/*port*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Specifies the Ethernet interface and enters interface configuration mode.

Step 3	**eou default** **Example:** `switch(config-if)# eou default`	Resets the EAPoUDP configuration for the interface to the default values.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits interface configuration mode.

Step 5	(Optional) **show eou interface ethernet** *slot*/*port* **Example:** `switch(config)# show eou interface ethernet 2/1`	Displays the EAPoUDP configuration.

	Command or Action	Purpose
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.
Resetting the EAPoUDP Global Configuration to the Default Values
Enabling NAC on an Interface.

10.8.18 Configuring IP Device Tracking

You can configure IP device tracking. The process for the IP device tracking for AAA servers operates is as follows:

- The Inspur INOS device detects a new session.
- Before posture validation is triggered and if the AAA server is unreachable, the Inspur INOS device applies the IP device tracking policy and maintains the session state as AAA DOWN.
- When the AAA server is once again available, a revalidation occurs for the host.

SUMMARY STEPS

1. **configure terminal**
2. **ip device tracking enable**
3. (Optional) **ip device tracking probe** {**count** *count* | **interval** *seconds*}
4. (Optional) **radius-server host** {*hostname* | *ip-address*} **test** [**username** *username* [**password** *password*]] [**idle-time** *minutes*]
5. **exit**
6. (Optional) **show ip device tracking all**
7. (Optional) **show radius-server** {*hostname* | *ip-address*}
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**ip device tracking enable** **Example:** `switch(config)# ip device tracking enable`	Enables the IP device tracking. The default state is enabled.
Step 3	(Optional) **ip device tracking probe** {**count** *count* | **interval** *seconds*} **Example:** `switch(config)# ip device tracking probe count 4`	Configures these parameters for the IP device tracking table: **count** Sets the number of times that the Inspur INOS device sends the ARP probe. The range is from 1 to 5. The default is 3.

	Command or Action	Purpose
		interval Sets the number of seconds that the Inspur INOS device waits for a response before resending the ARP probe. The range is from 1 to 302300 seconds. The default is 30 seconds
Step 4	(Optional) **radius-server host** {*hostname*|*ip-address*} **test** [**username** *username* [**password** *password*]] [**idle-time** *minutes*] **Example:** `switch(config)# radius-server host 10.10.1.1 test username User2 password G1r2D37&k idle-time 5`	Configures RADIUS server test packet parameters. The default username is test and the default password is test. The **idle-time** parameter determines how often the server is tested to determine its operational status. If there is no traffic to the RADIUS server, the NAD sends dummy packets to the RADIUS server based on the idle timer value. The default value for the idle timer is 0 minutes (disabled). If you have multiple RADIUS servers, reenter this command.

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 6	(Optional) **show ip device tracking all** **Example:** `switch# show ip device tracking all`	Displays IP device tracking information.
Step 7	(Optional) **show radius-server** {*hostname* | *ip-address*} **Example:** `switch# show radius-server 10.10.1.1`	Displays RADIUS server information.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling EAPoUDP.

10.8.19 Clearing IP Device Tracking Information

You can clear IP device tracking information for AAA servers.

SUMMARY STEPS

1. (Optional) **clear ip device tracking all**
2. (Optional) **clear ip device tracking interface ethernet** *slot/port*
3. (Optional) **clear ip device tracking ip-address** *ipv4-address*
4. (Optional) **clear ip device tracking mac-address** *mac-address*
5. (Optional) **show ip device tracking all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **clear ip device tracking all** **Example:** `switch# clear ip device tracking all`	Clears all EAPoUDP sessions.
Step 2	(Optional) **clear ip device tracking interface ethernet** *slot/port* **Example:** `switch# clear ip device tracking interface ethernet 2/1`	Clears EAPoUDP sessions on a specified interface.
Step 3	(Optional) **clear ip device tracking ip-address** *ipv4-address* **Example:** `switch# clear ip device tracking ip-address 10.10.1.1`	Clears an EAPoUDP session for a specified IPv4 address in the format A.B.C.D.
Step 4	(Optional) **clear ip device tracking mac-address** *mac-address* **Example:** `switch# clear ip device tracking mac-address 000c.30da.86f4`	Clears an EAPoUDP session for a specified MAC address in the format XXXX.XXXX.XXXX.
Step 5	(Optional) **show ip device tracking all** **Example:** `switch# show ip device tracking all`	Displays IP device tracking information.

10.8.20 Manually Initializing EAPoUDP Sessions

You can manually initialize EAPoUDP sessions.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. (Optional) **eou initialize all**
2. (Optional) **eou initialize authentication** {**clientless** | **eap** | **static**}
3. (Optional) **eou initialize interface ethernet** *slot/port*
4. (Optional) **eou initialize ip-address** *ipv4-address*
5. (Optional) **eou initialize mac-address** *mac-address*
6. (Optional) **eou initialize posturetoken** *name*
7. (Optional) **show eou all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **eou initialize all**	Initializes all EAPoUDP sessions.

	Command or Action	Purpose
	Example: `switch# eou initialize all`	
Step 2	(Optional) **eou initialize authentication** {**clientless** | **eap** | **static**} **Example:** `switch# eou initialize authentication static`	Initializes EAPoUDP sessions with a specified authentication type.
Step 3	(Optional) **eou initialize interface ethernet** *slot/port* **Example:** `switch# eou initialize interface ethernet 2/1`	Initializes EAPoUDP sessions on a specified interface.
Step 4	(Optional) **eou initialize ip-address** *ipv4-address* **Example:** `switch# eou initialize ip-address 10.10.1.1`	Initializes an EAPoUDP session for a specified IPv4 address in the format A.B.C.D.
Step 5	(Optional) **eou initialize mac-address** *mac-address* **Example:** `switch# eou initialize mac-address 000c.30da.86f4`	Initializes an EAPoUDP session for a specified MAC address in the format XXXX.XXXX.XXXX.
Step 6	(Optional) **eou initialize posturetoken** *name* **Example:** `switch# eou initialize posturetoken Healthy`	Initializes an EAPoUDP session for a specific posture token name. **Note** Use the **show eou all** command to display the token names.
Step 7	(Optional) **show eou all** **Example:** `switch# show eou all`	Displays the EAPoUDP session configuration.

Related Topics

Enabling EAPoUDP.

10.8.21 Manually Revalidating EAPoUDP Sessions

You can manually revalidate EAPoUDP sessions.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. (Optional) **eou revalidate all**
2. (Optional) **eou revalidate authentication** {**clientless** | **eap** | **static**}
3. (Optional) **eou revalidate interface ethernet** *slot/port*
4. (Optional) **eou revalidate ip-address** *ipv4-address*
5. (Optional) **eou revalidate mac-address** *mac-address*

6. (Optional) **eou revalidate posturetoken** *name*
7. (Optional) **show eou all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **eou revalidate all** **Example:** `switch# eou revalidate all`	Revalidates all EAPoUDP sessions.
Step 2	(Optional) **eou revalidate authentication** {**clientless** | **eap** | **static**} **Example:** `switch# eou revalidate authentication static`	Revalidates EAPoUDP sessions with a specified authentication type.
Step 3	(Optional) **eou revalidate interface ethernet** *slot/port* **Example:** `switch# eou revalidate interface ethernet 2/1`	Revalidates EAPoUDP sessions on a specified interface.
Step 4	(Optional) **eou revalidate ip-address** *ipv4-address* **Example:** `switch# eou revalidate ip-address 10.10.1.1`	Revalidates an EAPoUDP session for a specified IPv4 address.
Step 5	(Optional) **eou revalidate mac-address** *mac-address* **Example:** `switch# eou revalidate mac-address 000c.30da.86f4`	Revalidates an EAPoUDP session for a specified MAC address.
Step 6	(Optional) **eou revalidate posturetoken** *name* **Example:** `switch# eou revalidate posturetoken Healthy`	Revalidates an EAPoUDP session for a specific posture token name. **Note** Use the **show eou all** command to display the token names.
Step 7	(Optional) **show eou all** **Example:** `switch# show eou all`	Displays the EAPoUDP session configuration.

Related Topics

Enabling EAPoUDP.

10.8.22 Clearing EAPoUDP Sessions

You can clear EAPoUDP sessions from the Inspur INOS device.

Before you begin

Enable EAPoUDP.

SUMMARY STEPS

1. (Optional) **clear eou all**

2. (Optional) **clear eou authentication** {**clientless** | **eap** | **static**}
3. (Optional) **clear eou interface ethernet** *slot/port*
4. (Optional) **clear eou ip-address** *ipv4-address*
5. (Optional) **clear eou mac-address** *mac-address*
6. (Optional) **clear eou posturetoken** *name*
7. (Optional) **show eou all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **clear eou all** **Example:** `switch# clear eou all`	Clears all EAPoUDP sessions.
Step 2	(Optional) **clear eou authentication** {**clientless** | **eap** | **static**} **Example:** `switch# clear eou authentication static`	Clears EAPoUDP sessions with a specified authentication type.
Step 3	(Optional) **clear eou interface ethernet** *slot/port* **Example:** `switch# clear eou interface ethernet 2/1`	Clears EAPoUDP sessions on a specified interface.
Step 4	(Optional) **clear eou ip-address** *ipv4-address* **Example:** `switch# clear eou ip-address 10.10.1.1`	Clears an EAPoUDP session for a specified IPv4 address.
Step 5	(Optional) **clear eou mac-address** *mac-address* **Example:** `switch# clear eou mac-address 000c.30da.86f4`	Clears an EAPoUDP session for a specified MAC address.
Step 6	(Optional) **clear eou posturetoken** *name* **Example:** `switch# clear eou posturetoken Healthy`	Clears an EAPoUDP session for a specific posture token name. **Note** Use the **show eou all** command to display the token names.
Step 7	(Optional) **show eou all** **Example:** `switch# show eou all`	Displays the EAPoUDP session configuration.

Related Topics

Enabling EAPoUDP.

10.8.23 Disabling the EAPoUDP Feature

You can disable the EAPoUDP feature on the Inspur INOS device.

Caution	Disabling EAPoUDP removes all EAPoUDP configuration from the Inspur INOS device.

Before you begin

Enable the 802.1X feature on the Inspur INOS device.

SUMMARY STEPS

1. **configure terminal**
2. **no feature eou**
3. **exit**
4. (Optional) **show feature**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no feature eou** **Example:** `switch(config)# no feature eou`	Disables EAPoUDP. **Caution** Disabling the EAPoUDP feature removes all EAPoUDP configuration.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits configuration mode.

Step 4	(Optional) **show feature** **Example:** `switch# show feature`	Displays the enabled or disabled status for the Inspur INOS features.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

10.9 Verifying the NAC Configuration

To display NAC configuration information, perform one of the following tasks:

Command	Purpose
show eou [**all** | **authentication** {**clientless** | **eap** | **static**} | **interface ethernet** *slot*/*port* | **ip-address** *ipv4-address* | **mac-address** *mac-address* | **posturetoken** *name*]	Displays the EAPoUDP configuration.
show ip device tracking [**all** | **interface ethernet** *slot*/*port* | **ip-address** *ipv4-address* | **mac-address** *mac-address*]	Displays IP device tracking information.
show running-config eou [**all**]	Displays the EAPoUDP configuration in the running configuration.
show startup-config eou	Displays the EAPoUDP configuration in the startup configuration.

For detailed information about the fields in the output from this command, see the *Inspur CN12700 Series INOS Security Command Reference*.

10.10 Configuration Example for NAC

The following example shows how to configure NAC:

```
feature eou
aaa authentication eou default group radius
mac access-list macacl-01
  10 permit any any 0x100
interface Ethernet8/1
  mac access-group macacl-01
```

10.11 Additional References forNAC

This section lists the additional references for NAC.

Related Documents

Related Topic	Document Title
Inspur INOS licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*

CHAPTER 11 Configuring Inspur TrustSec

This chapter describes how to configure Inspur TrustSec on Inspur INOS devices. This chapter includes the following sections:

- Finding Feature Information
- Information About Inspur TrustSec
- Virtualization Support
- Licensing Requirements for Inspur TrustSec
- Prerequisites for Inspur TrustSec
- Guidelines and Limitations for Inspur TrustSec
- Default Settings for Inspur TrustSec Parameters
- Configuring Inspur TrustSec
- Inspur TrustSec Support on Port-Channel Members
- Verifying the Inspur TrustSec Configuration
- Configuration Examples for Inspur TrustSec
- Troubleshooting Inspur TrustSec
- Additional References for Inspur TrustSec
- Feature History for Inspur TrustSec

11.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

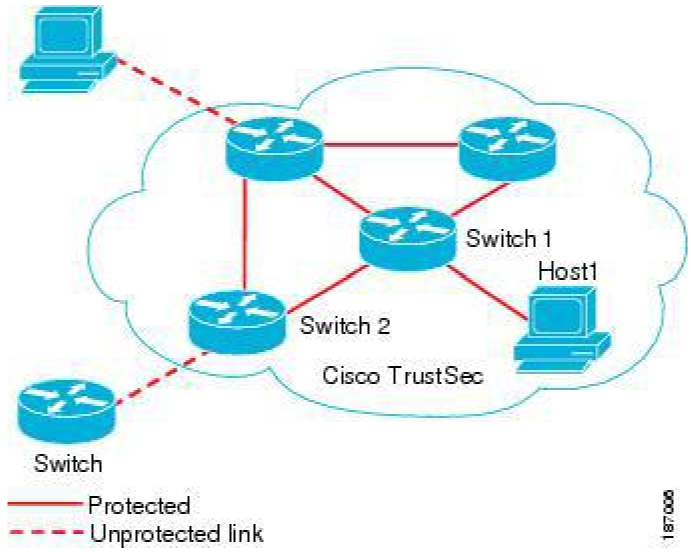
11.2 Information About Inspur TrustSec

This section provides information about Inspur TrustSec.

11.2.1 Inspur TrustSec Architecture

The Inspur TrustSec security architecture builds secure networks by establishing clouds of trusted network devices. Each device in a cloud is authenticated by its neighbors. Communication on the links between devices in the cloud is secured with a combination of encryption, message integrity checks, and data-path replay protection mechanisms. Inspur TrustSec uses the device and user identification information acquired during authentication for classifying, or coloring, the packets as they enter the network. This packet classification is maintained by tagging packets on ingress to the Inspur TrustSec network so that they can be properly identified for the purpose of applying security and other policy criteria along the data path. The tag, also called the security group tag (SGT), allows the network to enforce the access control policy by enabling the endpoint device to act upon the SGT to filter traffic.

Figure 8 : Inspur TrustSec Network Cloud Example



This figure shows an example of a Inspur TrustSec network cloud. In this example, several networking devices and an endpoint device are inside the cloud. One endpoint device and one networking device are outside the cloud because they are not Inspur TrustSec-capable devices or they have been refused access.

The Inspur TrustSec architecture consists of the following major components:

Authentication

Verifies the identity of each device before allowing it to join the Inspur TrustSec network

Authorization

Decides the level of access to the Inspur TrustSec network resources for a device based on its authenticated identity

Access Control

Applies access policies on a per-packet basis using the source tags on each packet

Secure communication

Provides encryption, integrity, and data-path replay protection for the packets that flow over each link in the Inspur TrustSec network

A Inspur TrustSec network has the following entities:

Supplicants

Devices that attempt to join a Inspur TrustSec network

Authenticators (AT)

Devices that are already part of a Inspur TrustSec network

Authorization Server

Servers that might provide authentication information, authorization information, or both

When the link between the supplicant and the AT comes up, the following sequence of events might occur:

Authentication (802.1X)

The authentication server authenticates the supplicant or the authentication is completed if you configure the devices to unconditionally authenticate each other.

Authorization

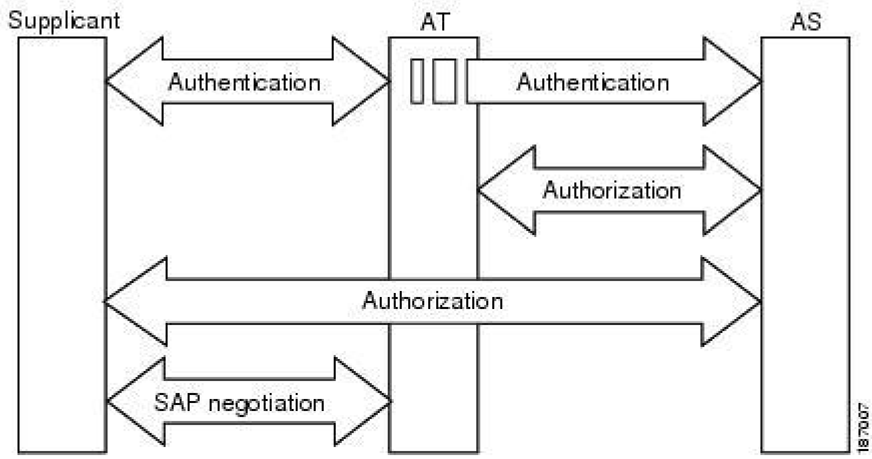
Each side of the link obtains policies, such as SGT and ACLs, that apply to the link. A supplicant might need to use the AT as a relay if it has no other Layer 3 route to the authentication server.

Security Association Protocol Negotiation

The EAPOL-Key exchange occurs between the supplicant and the AT to negotiate a cipher suite, exchange security parameter indexes (SPIs), and manage keys. Successful completion of all three tasks results in the establishment of a security association (SA).

The ports stay in the unauthorized state (blocking state) until the SA protocol negotiation is complete.

Figure 9：SA Protocol Negotiation



This figure shows the SA protocol negotiation, including how the ports stay in unauthorized state until the SA protocol negotiation is complete.

SA protocol negotiation can use any of the following modes of operation:

- Galois/Counter Mode (GCM) encryption
- GCM authentication (GMAC)
- No encapsulation (clear text)
- Encapsulation with no encryption or authentication

Based on the IEEE 802.1AE standard, Inspur TrustSec uses ESP-128 GCM and GMAC.

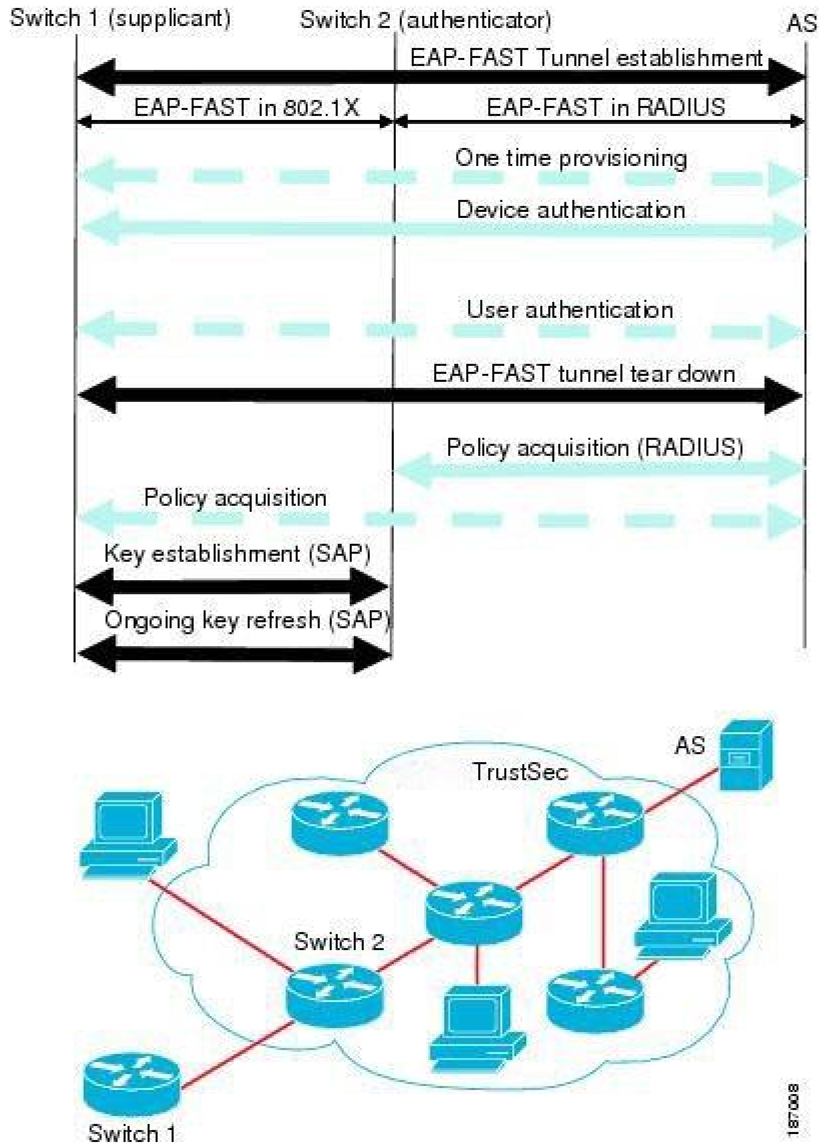
11.2.2 Authentication

Inspur TrustSec authenticates a device before allowing it to join the network. Inspur TrustSec uses 802.1X authentication with Extensible Authentication Protocol Flexible Authentication through Secure Tunnel (EAP-FAST) as the Extensible Authentication Protocol (EAP) method to perform the authentication.

Inspur TrustSec and Authentication

Inspur TrustSec uses EAP-FAST for authentication. EAP-FAST conversations allow other EAP method exchanges inside the EAP-FAST tunnel using chains, which allows administrators to use traditional user authentication methods, such as Microsoft Challenge Handshake Authentication Protocol Version 2 (MSCHAPv2), while still having security provided by the EAP-FAST tunnel.

Figure 10： Inspur TrustSec Authentication



This figure shows the EAP-FAST tunnel and inner methods used in Inspur TrustSec.

Inspur TrustSec Enhancements to EAP-FAST

The implementation of EAP-FAST for Inspur TrustSec has the following enhancements:

Authenticate the authenticator

Securely determines the identity of the AT by requiring the AT to use its protected access credential (PAC) to derive the shared secret between itself and the authentication server. This feature also prevents you from configuring RADIUS shared secrets on the authentication server for every possible IP address that can be used by the AT.

Notify each peer of the identity of its neighbor

By the end of the authentication exchange, the authentication server has identified the supplicant and the AT. The authentication server conveys the identity of the AT, and whether the AT is Inspur

TrustSec-capable, to the supplicant by using additional type-length-value parameters (TLVs) in the protected EAP-FAST termination. The authentication server also conveys the identity of the supplicant and whether the supplicant is Inspur TrustSec-capable to the AT by using RADIUS attributes in the Access-Accept message. Because each peer knows the identity of its neighbor, it can send additional RADIUS Access-Requests to the authentication server to acquire the policy to be applied on the link.

AT posture evaluation

The AT provides its posture information to the authentication server whenever it starts the authentication exchange with the authentication server on behalf of the supplicant.

802.1X Role Selection

In 802.1X, the AT must have IP connectivity with the authentication server because it has to relay the authentication exchange between the supplicant and the AT using RADIUS over UDP/IP. When an endpoint device, such as a PC, connects to a network, it is obvious that it should act as a supplicant. However, in the case of a Inspur TrustSec connection between two network devices, the 802.1X role of each network device might not be immediately apparent to the other network device.

Instead of requiring manual configuration of the AT and supplicant roles for the Inspur INOS devices, Inspur TrustSec runs a role-selection algorithm to automatically determine which Inspur INOS device acts as the AT and which device acts as the supplicant. The role-selection algorithm assigns the AT role to the device that has IP reachability to a RADIUS server. Both devices start both the AT and supplicant state machines. When a Inspur INOS device detects that its peer has access to a RADIUS server, it terminates its own AT state machine and assumes the role of the supplicant. If both Inspur INOS devices have access to a RADIUS server, the algorithm compares the MAC addresses used as the source for sending the EAP over LAN (EAPOL) packets. The Inspur INOS device that has the MAC address with the higher value becomes the AT and the other Inspur INOS device becomes the supplicant.

Inspur TrustSec Authentication Summary

By the end of the Inspur TrustSec authentication process, the authentication server has performed the following actions:

- Verified the identities of the supplicant and the AT
- Authenticated the user if the supplicant is an endpoint device At the end of the Inspur TrustSec authentication process, the AT and the supplicant have the following information:
- Device ID of the peer
- Inspur TrustSec capability information of the peer
- Key used for the SA protocol

Device Identities

Inspur TrustSec does not use IP addresses or MAC addresses as device identities. Instead, assign a name (device ID) to each Inspur TrustSec-capable Inspur INOS device to identify it uniquely in the Inspur TrustSec network. This device ID is used for the following:

- Looking up authorization policy
- Looking up passwords in the databases during authentication

Device Credentials

Inspur TrustSec supports password-based credentials. The authentication servers may use self-signed certificates instead. Inspur TrustSec authenticates the supplicants through passwords and uses MSCHAPv2 to provide mutual authentication even if the authentication server certificate is not verifiable.

The authentication server uses these credentials to mutually authenticate the supplicant during the EAP-FAST phase 0 (provisioning) exchange, where a PAC is provisioned in the supplicant. Inspur TrustSec does not perform the EAP-FAST phase 0 exchange again until the PAC expires and only performs EAP-FAST phase 1 and phase 2 exchanges for future link bringups. The EAP-FAST phase 1 exchange uses the PAC to mutually authenticate the authentication server and the supplicant. Inspur TrustSec uses the device credentials only during the PAC provisioning (or reprovisioning) steps.

The authentication server uses a temporarily configured password to authenticate the supplicant when the supplicant first joins the Inspur TrustSec network. When the supplicant first joins the Inspur TrustSec network, the authentication server authenticates the supplicant using a manufacturing certificate and then generates a strong password and pushes it to the supplicant with the PAC. The authentication server also keeps the new password in its database. The authentication server and the supplicant use this password for mutual authentication in all future EAP-FAST phase 0 exchanges.

User Credentials

Inspur TrustSec does not require a specific type of user credentials for endpoint devices. You can choose any type of authentication method for the user (for example, MSCHAPv2, LEAP, generic token card (GTC), or OTP) and use the corresponding credentials. Inspur TrustSec performs user authentication inside the EAP-FAST tunnel as part of the EAP-FAST phase 2 exchange.

11.2.3 Native VLAN Tagging on Trunk and FabricPath Ports

MACSec is supported over FabricPath through native VLAN tagging on trunk and FabricPath ports feature. Native VLAN tagging can be configured either globally or on an interface for control packets and data packets. Use the following commands to enable native VLAN tagging globally:

- **vlan dot1q tag native exclude control**
- **vlan dot1q tag native fabricpath**
- **vlan dot1q tag native fabricpath exclude control**

Use the following commands to enable native VLAN tagging on FabricPath ports:

- switchport trunk native vlan tag exclude control
- **switchport fabricpath native vlan tag**
- **switchport fabricpath native vlan tag exclude control**

Native VLAN tagging provides support for tagged and untagged modes when sending or receiving packets. The following table explains the mode for a packet on a global configuration or port configuration for the above commands.

Tagging Configuration	TX-Control	TX-Data (Native VLAN)	RX-Control	RX-Data
Global trunk port tagging	Untagged	Tagged	Untagged and tagged	Tagged
Global FabricPath tagging	Untagged	Untagged	Untagged and tagged	Untagged and tagged
Global FabricPath tagging for data packets	Untagged	Tagged	Untagged and tagged	Tagged
Port-level trunk port tagging	Untagged	Tagged	Untagged and tagged	Tagged
Port-level Fabricpath tagging	Untagged	Untagged	Untagged and tagged	Untagged and tagged
Port-level FabricPath tagging for data packets	Untagged	Tagged	Untagged and tagged	Tagged

11.2.4 SGACLs and SGTs

In security group access lists (SGACLs), you can control the operations that users can perform based on assigned security groups. The grouping of permissions into a role simplifies the management of the security policy. As you add users to a Inspur INOS device, you simply assign one or more security groups and they immediately receive the appropriate permissions. You can modify security groups to introduce new privileges or restrict current permissions.

Inspur TrustSec assigns a unique 16-bit tag, called the security group tag (SGT), to a security group. The number of SGTs in a Inspur INOS device is limited to the number of authenticated network entities. The SGT is a single label that indicates the privileges of the source within the entire enterprise. Its scope is global within a Inspur TrustSec network.

The management server derives the SGTs based on the security policy configuration. You do not have to configure them manually.

Once authenticated, Inspur TrustSec tags any packet that originates from a device with the SGT that represents the security group to which the device is assigned. The packet carries this SGT throughout the network within the Inspur TrustSec header. Because this tag represents the group of the source, the tag is referred to as the source SGT. At the egress edge of the network, Inspur TrustSec determines the group that is assigned to the packet destination device and applies the access control policy.

Inspur TrustSec defines access control policies between the security groups. By assigning devices within the network to security groups and applying access control between and within the security groups, Inspur TrustSec essentially achieves access control within the network.

Figure 11： GACL Policy Example

This figure shows an example of an SGACL policy.

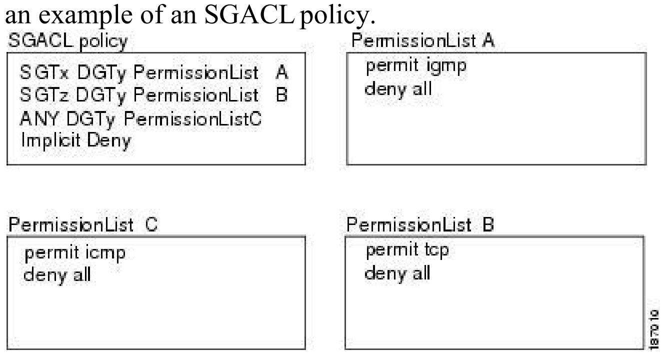
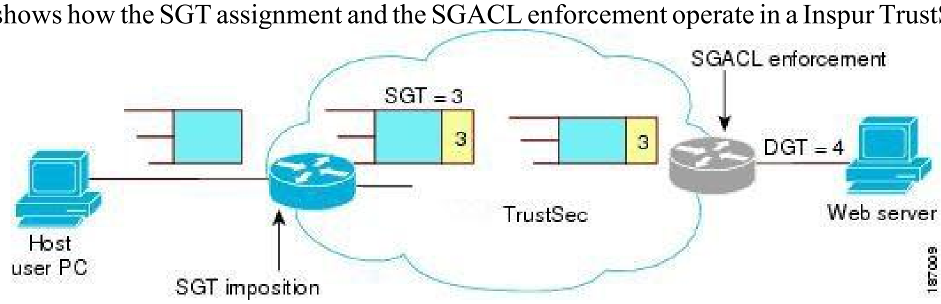


Figure 12： SGT and SGACL in Inspur TrustSec Network

This figure shows how the SGT assignment and the SGACL enforcement operate in a Inspur TrustSec network.



The Inspur INOS device defines the Inspur TrustSec access control policy for a group of devices as opposed to IP addresses in traditional ACLs. With such a decoupling, the network devices are free to move throughout the network and change IP addresses. Entire network topologies can change. As long as the roles and the permissions remain the same, changes to the network do not change the security policy. This feature greatly reduces the size of ACLs and simplifies their maintenance.

In traditional IP networks, the number of access control entries (ACEs) configured is determined as follows:

Number of ACEs = (number of sources specified) X (number of destinations specified) X (number of permissions specified)

Inspur TrustSec uses the following formula:

Number of ACEs = number of permissions specified

Determining the Source Security Group

A network device at the ingress of the Inspur TrustSec network cloud needs to determine the SGT of the packet entering the Inspur TrustSec network cloud so that it can tag the packet with that SGT when it forwards it into the Inspur TrustSec network cloud. The egress network device needs to determine the SGT of the packet so that it can apply the SGACLs.

The network device can determine the SGT for a packet using one of the following methods:

- Obtain the source SGT during policy acquisition—After the Inspur TrustSec authentication phase, a network device acquires a policy from an authentication server. The authentication server indicates whether the peer device is trusted or not. If a peer device is not trusted, the authentication server can also provide an SGT to apply to all packets coming from the peer device.Obtain the source SGT field from the Inspur TrustSec header—If a packet comes from a trusted peer device, the Inspur TrustSec header carries the correct SGT field if the network device is not the first network device in the Inspur TrustSec network cloud for the packet.
- Look up the source SGT based on the source IP address—In some cases, you can manually configure the policy to decide the SGT of a packet based on the source IP address. The SGT Exchange Protocol (SXP) can also populate the IP-address-to-SGT mapping table.

Determining the Destination Security Group

The egress network device in a Inspur TrustSec network cloud determines the destination group for applying the SGACL. In some cases, ingress devices or other nonegress devices might have destination group information available. In those cases, SGACLs might be applied in these devices rather than in egress devices.

Inspur TrustSec determines the destination group for the packet in the following ways:

- Destination SGT of the egress port obtained during the policy acquisition
- Destination SGT lookup based on the destination IP address Do not configure the destination SGT to enforce Inspur TrustSec on egress broadcast, multicast, and unknown unicast traffic on Fabric Extender (FEX) or vEthernet ports. Instead, set the DST to zero (unknown). The following is an example of the correct configuration:

```
cts role-based access-list acl-on-fex-egress
     deny udp
     deny ip
cts role-based sgt 9 dst 0 access-list acl-on-fex-egress
```

11.2.5 SGACL Detailed Logging

From Inspur INOS Release 8.2(3), you can use the SGACL detailed logging feature to observe the effects of SGACL policies after their enforcement at the egress point. You can check the following:

- Whether a flow is permitted or denied
- Whether a flow is monitored or enforced by the SGACL By default, the SGACL detailed logging feature is disabled.

From Inspur INOS Release8.2(3), the SGACL detailed logging feature is supported on the Inspur CN series modules. However, the SGACL detailed logging information for traffic arriving on interfaces of the Inspur series modules is supported when the following conditions are met:

- The source SGT for traffic is derived locally on the enforcement device.
- The interfaces of the Inspur series modules do not have any port-SGT configuration.

11.2.6 SGACL Monitor Mode

During the predeployment phase of Inspur TrustSec, an administrator will use the monitor mode to test the security policies without enforcing them to make sure that the policies are what were originally intended. If there is something wrong with the security policy, the monitor mode provides a convenient mechanism for identifying the same, along with an opportunity to correct the policy before enabling SGACL enforcement. This enables administrators to have an increased visibility to the outcome of the policy actions before they enforce it, and confirm that the subject policy meets the security requirements (access is denied to resources if users are not authorized).

The monitoring capability is provided at the SGT-DGT pair level. By default, the SGACL monitoring mode is disabled. When you enable the SGACL monitoring mode feature, the deny action is implemented as an ACL permit on the line cards. This allows the SGACL counters and logging to display how connections are handled by the SGACL policy. Since all the monitored traffic is now permitted, there is no disruption of service due to SGACLs while in the SGACL monitor mode.

From Inspur INOS Release 8.2(3), the SGACL monitor mode feature is supported on the Inspur CN series modules. However, the SGACL monitor mode feature is not supported on the Inspur CN F3modules.

11.2.7 Overview of SGACL Egress Policy Overwrite

In releases earlier than Inspur INOS Release 8.2(3), SGACLs from only one source was valid. Consider the following scenarios:

- SGACL is configured using CLI followed by SGACL downloaded from Integrated Services Engine (ISE). In this case, the SGACL downloaded from ISE is ignored.
- SGACL is downloaded from ISE followed by SGACL configured using CLI. In this case, the SGACL downloaded from ISE is overwritten.

From Inspur INOS Release 8.2(3), the SGACLs downloaded using ISE and SGACLs configured using CLI can coexist. You can prioritize whether to use SGACLs downloaded from ISE or SGACLs configured by using CLI. Use the **[no] cts role-based priority-static** command to choose the install priority between the SGACLs configured by using CLI or SGACLs downloaded by ISE. By default, the SGACLs configured by using CLI have higher priority in Inspur INOS.

SGACL Policy Enforcement With Inspur TrustSec SGT Caching

This section discusses about the special cases that needs to be considered when you enable SGT Caching feature with Inspur TrustSec SGACL policy enforcement. Specifically, the SGT Caching mode for **sgt=any**,**dgt=any**, and **sgt=0,dgt=0**.

The SGT Caching feature mandates the installation of two main SGACL policies, that is, <**sgt = any, dgt = any**> and <**sgt = 0, dgt = 0**> in the hardware. If these SGACL policies are not configured by using CLI, then CTS manager creates and installs the reserved SGACL policies: <**sgt = any, dgt= any, permit all log**> and <sgt = 0, dgt = 0, permit all>.

Prior to Inspur INOS Release 8.2(3), if the SGT Caching feature is enabled with Inspur TrustSec SGACL policy enforcement, the following changes are observed:

- The reserved SGACL created by SGT caching is considered as SGACL configured by CLI. The SGACL policy with values <**sgt =any, dgt = any, ise_user_rbacl**> downloaded from ISE is ignored, because SGACLs configured by using CLI are given higher priority. Therefore, the reserved SGACL with values <**sgt=any dgt=any, permit all log**> is installed in hardware, when SGACL with <**sgt =any, dgt = any**> is not configured by the user by using CLI and only available in ISE.
- SGACL traffic counters are not supported for the reserved SGACLs. Therefore, the SGACL traffic counters are not supported for the default Any-Any policy, when SGT-caching with enforcement is enabled and there is no SGACL with <**sgt=any , dgt=any**> configured by using CLI.
- If you` configure an SGACL with values <**sgt=any,dgt=any,user_rbacl** by using CLI, the **permit all log** is

appended with the **user_rbacl** ACE and installed in hardware. SGACL traffic counters are supported as usual for this user installed with Any-Any policy by using CLI.

Starting from Inspur INOS Release 8.2(3), the rules that apply to CLI installed Any-Any SGACLs with SGT-caching feature in prior releases, are also applicable to the ISE downloaded SGACLs. In case of coexistence of the Any-Any SGACL from both CLI and ISE, the policy that needs to be used is decided based on the priority selection. SGACL traffic counters for the default policy are supported as long as the Any-Any policy from either CLI or ISE is available.

SGACL Egress Policy Overwrite With Monitor Mode

The following table provides information about how SGACLs from different sources (CLI or ISE) are selected and installed based on the "install priority" and "monitor mode" configuration.

Priority Configured	Monitor Mode Status	CLI SGACL Only	CLI Monitored SGACL Only	ISE SGACL Only	ISE Monitored SGACL Only	CLI SGACL and ISE SGACL	CLI and ISE Monitored SGACL	CLI Monitored SGACL and ISE SGACL	CLI Monitored SGACL and ISE Monitored SGACL
no cts role-based priority-static	Disabled	Install CLI SGACL	Install CLI SGACL	Install ISE SGACL	No Install	Install ISE SGACL	Install CLI SGACL	Install ISE SGACL	No Install
cts role-based priority-static	Disabled	Install CLI SGACL	Install CLI SGACL	Install ISE SGACL	No Install	Install CLI SGACL	Install CLI SGACL	Install CLI SGACL	Install CLI SGACL
no cts role-based priority-static	Enabled	Install CLI SGACL	Install CLI Monitored SGACL	Install ISE SGACL	Install CLI Monitored SGACL	Install ISE SGACL	Install ISE Monitored SGACL	Install ISE SGACL	Install ISE Monitored SGACL
cts role-based priority-static	Enabled	Install CLI SGACL	Install CLI Monitored SGACL	Install ISE SGACL	Install CLI Monitored SGACL	Install CLI SGACL	Install CLI SGACL	Install CLI Monitored SGACL	Install CLI Monitored SGACL

11.2.8 Overview of SGACL Policy Enforcement Per Interface

From Inspur INOS Release 8.2(3), you can enable or disable SGACL policy enforcement on Layer 3 (L3) physical interfaces and port-channels.

Consider the following scenario with two Inspur CN12700 series switches. This scenario provides an overview about using the SGACL policy enforcement per interface.

Figure 13： SGACL Policy Enforcement Per Interface Enabled

The following table provides information about the SGACL policies.

SGT Information	SGT10	SGT20	SGT Unknown
SGT10	Permit	Deny	Deny
SGT20	Deny	Permit	Deny

When SGACLs are applied on this setup, hosts with SGT10 cannot communicate with SGT20 and Unknown SGT hosts, because SGACL policy drops the packets. However, when you disable the SGACL policy enforcement on the port E4/1:

- The host H1 cannot communicate with the host H2 because this network traffic is subjected to the SGT 10 DGT 20 Deny policy.
- The host H1 can communicate with host H3 even if this network traffic is subjected to the SGT 10 DGT unknown Deny policy. This communication is possible because the packet is exiting through the port E4/1 on which the SGACL policy enforcement is disabled.

The following figure shows the packet routes between different hosts after the SGACL policy enforcement is disabled on the port E4/1.

Figure 14： SGACL Policy Enforcement Per Interface Disabled

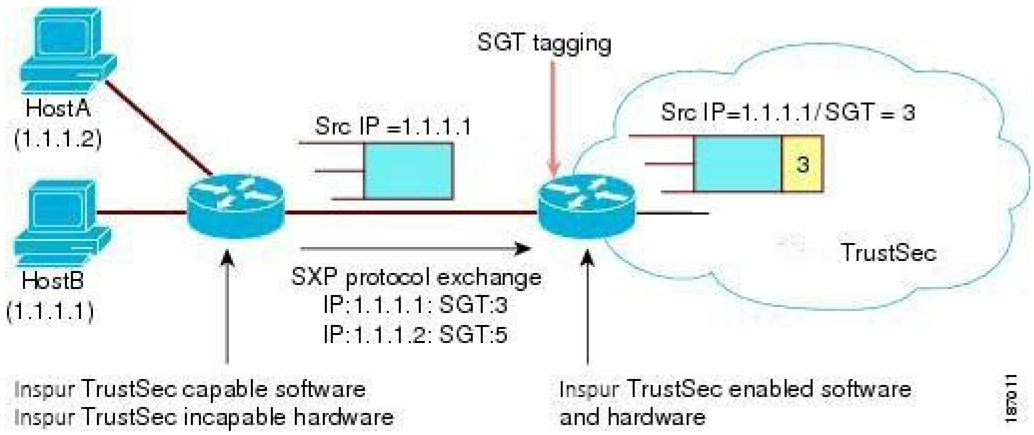
11.2.9 SXP for SGT Propagation Across Legacy Access Networks

The Inspur INOS device hardware in the access layer supports Inspur TrustSec. Without the Inspur TrustSec hardware, the Inspur TrustSec software cannot tag the packets with SGTs. You can use SXP to propagate the SGTs across network devices that do not have hardware support for Inspur TrustSec.

SXP operates between access layer devices and distribution layer devices. The access layer devices use SXP to pass the IP addresses of the Inspur TrustSec-authenticated devices with their SGTs to the distribution switches. Distribution devices with both Inspur TrustSec-enabled software and hardware can use this information to tag packets appropriately and enforce SGACL policies.

Figure 15： Using SXP to Propagate SGT Information

This figure shows how to use SXP to propagate SGT information in a legacy network.



Tagging packets with SGTs requires hardware support. You might have devices in your network that cannot tag packets with SGTs. To allow these devices to send IP address-to-SGT mappings to a device that has Inspur TrustSec-capable hardware, you must manually set up the SXP connections. Manually setting up an SXP connection requires the following:

- If you require SXP data integrity and authentication, you must configure the same SXP password on both of the peer devices. You can configure the SXP password either explicitly for each peer connection or globally for the device. The SXP password is not required.
- You must configure each peer on the SXP connection as either an SXP speaker or an SXP listener. The speaker device distributes the SXP information to the listener device.
- You can specify a source IP address to use for each peer relationship or you can configure a default source IP address for peer connections where you have not configured a specific source IP address.

11.2.10 Inspur TrustSec with SXPv3

The Security Group Tag (SGT) Exchange Protocol (SXP) is a control protocol, which propagates IP address-SGT binding information across network devices. From Inspur INOS Release 8.2(3), the SXP version 3 (SXPv3) feature provides support to transport the IPv4 subnet to the SGT bindings.

By using the subnet for SGT bindings, you can minimize the forward information base (FIB) entries needed for storing the mapping, which allows users to increase the scale of the TrustSec deployments. In many scenarios, you can use subnet-SGT bindings instead of the L3 interface-SGT.

SXPv3 Subnet Expansion

The SXPv3 protocol allows you to configure the expansion limit for a subnet binding. SXP expands a subnet binding to host address bindings when a connection is set up with a peer with a version earlier than Version

3. SXP binding expansion is applicable only to IPv4 subnet binding.

The characteristics of subnet expansion are as follows:

- When expanding the bindings for overlapping IP addresses with different SGT values, the mapping is obtained from the IP address with the longest prefix length.
- If the subnet expansion reaches the configured limit, a system log is generated for the subnet that cannot be expanded.
- Binding expansion does not expand broadcast IP addresses in a subnet. Also, note that SXP does not summarize host IP addresses to subnet bindings. In the SXP propagation path, if there is a node that does not understand subnet binding, the bindings are expanded and propagated through the rest of the propagation path as host IP binding even though there is a node that understands subnet binding.
- The default expansion limit is zero (0) and the maximum allowed expansion limit is 65535. You can set the expansion limit as 0 when you do not have any devices supporting a lower version of SXP, in the network.

You can use the **cts sxp mapping network-map** [*num_bindings*] command to expand the network limit. The *num_bindings* parameter can accept value from 0 to 65535. The value zero (0) indicates that no expansion is allowed and 65535 is the maximum expansion limit allowed. The default value is zero (0).

Consider an example when the expansion limit is set to 67 and the subnet is /24. Inspur INOS expands the first 67 IP addresses for the first subnet SGT known to Inspur TrustSec. Since subnet /24 contains more hosts, it will never be fully expanded, and a syslog is generated.

SXP Version Negotiation

The SXP session is established between speaker devices and listener devices. By default, the Inspur TrustSec device advertises the highest supported SXP version. The negotiation is made based on the highest common version supported by the speaker and listener devices. A standalone Inspur TrustSec-supported device can establish SXP session with different versions, with its peer devices, depending on the SXP versions of the peer devices.

The following table provides information about version negotiation for interoperability in different scenarios.

Table 20：SXP Version Negotiation Cases

Case Number	Speaker	Listener	SXP Session Status
1	SXPv1	SXPv1	SXPv1 session is established.
2	SXPv1	SXPv2	SXPv1 session is established.
3	SXPv1	SXPv3	SXPv1 session is established.
4	SXPv2	SXPv1	SXPv1 session is established.
5	SXPv2	SXPv2	Not possible because a Inspur CN12700 device does not support SXPv2.
6	SXPv2	SXPv3	If a Inspur CN12700 device with SXPv3 is interoperating with another Inspur SXP device having SXPv2, the Inspur CN12700 device ensures that the connection is established as SXPv1.
7	SXPv3	SXPv1	SXP session is established.
8	SXPv3	SXPv2	If a Inspur CN12700 device with SXPv3 is interoperating with another Inspur SXP device having SXPv2, the Inspur CN12700 device ensures that the connection is established as SXPv1.
9	SXPv3	SXPv3	SXPv3 session is established.
10	SXPv1	SXPv4	SXPv1 session is established.
11	SXPv2	SXPv4	If a Inspur CN12700 device with SXPv4 is interoperating with another Inspur SXP device having SXPv2, the Inspur CN12700 device ensures that the connection is established as SXPv1.
12	SXPv3	SXPv4	SXPv3 session is established.
13	SXPv4	SXPv1	SXPv1 session is established.
14	SXPv4	SXPv2	If a Inspur CN12700 device with SXPv4 is interoperating with another Inspur SXP device having SXPv2, the Inspur CN12700 device ensures that the connection is established as SXPv1.
15	SXPv4	SXPv3	SXPv3 session is established.
16	SXPv4	SXPv4	SXPv4 session is established.

SXP Support for Default Route SGT Bindings

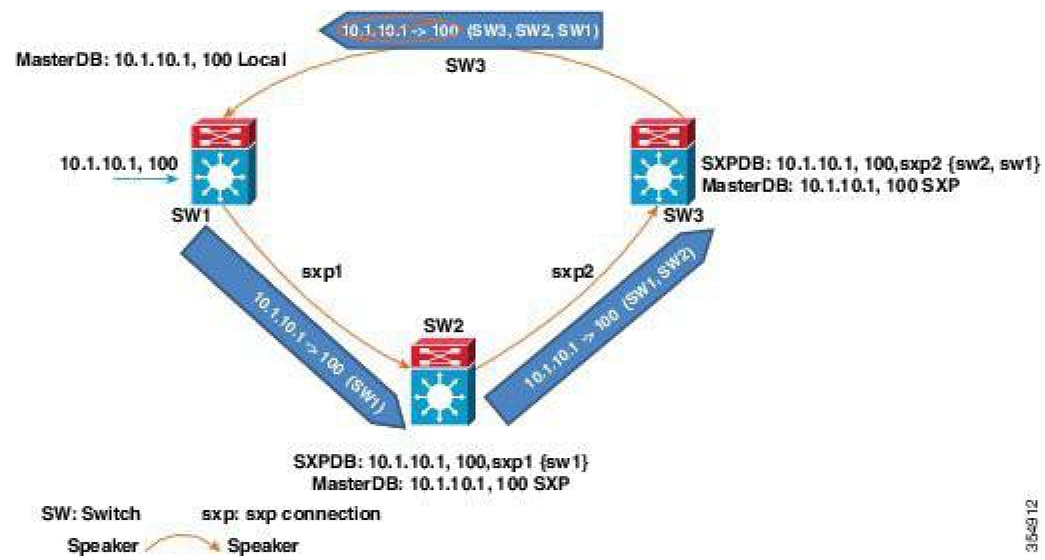
You can provide the default route for SGT bindings, when IP-SGT for the source IP address or destination IP address is not configured. In this scenario, SGT is derived from the default route entry. Note that you can use the default route only for the listener device with SXPv3. By default, the transport of SGT bindings through the default route by using SXP, is disabled. You can enable the transport of SGT bindings through the default route by using the **cts sxp allow default-route-sgt** command. Use the **no** form of this command to disable the default route of the SGT bindings.

11.2.11 Overview of Inspur TrustSec with SXPv4

Inspur TrustSec SXP version 4 (SXPv4) enhances the functionality of SXP by adding a loop detection mechanism to prevent stale binding in the network. SXP connections can be enabled such that the binding forwarded by one switch for an SXP connection can be received from another SXP connection, resulting in SXP connection loops. SXP loop topology might, however, result in stale binding in the network. SXPv4' s built-in loop detection and prevention mechanism addresses the stale binding issue whenever there is a loop between SXP nodes.

Loop prevention is achieved by adding SXP propagation path information when propagating (adding or deleting) bindings. Propagation path information keeps track of the network devices (via their node IDs) that the binding travels in an ordered manner. All nodes that participate in the network with looped SXP connections must run SXPv4 to function correctly. Loop detection is a mandatory capability in SXPv4.

Figure 16 : SXPv4 Loop Detection



In the figure above there are three network devices: SW1, SW2, and SW3. There are also three SXP connections: SXP1, SXP2, and SXP3, together which create an SXP connection loop. A binding (10.1.10.1, 100) is learned at SW1 through local authentication. The binding is exported by SW1 to SW2 together with the path information (that is, SW1, from where the binding is forwarded).

Upon receiving the binding, SW2 exports it to SW3, again prepending the path information (SW2, SW1). Similarly, SW3 forwards the binding to SW1 with path information SW3, SW2, SW1. When SW1 receives the binding, the path information is checked. If its own path attribute is in the binding update received, then a propagation loop is detected. This binding is dropped and not stored in the SXP binding database.

If the binding is removed from SW1, (for example, if a user logs off), a binding deletion event is sent. The deletion event goes through the same path as above. When it reaches SW1, no action will be taken as no such binding exists in the SW1 binding database.

Loop detection is done when a binding is received by an SXP but before it is added to the binding database.

SXP Node ID

An SXP node ID is used to identify the individual devices within the network. The node ID is a four-octet integer that can be configured by the user. If it is not configured by the user, Inspur TrustSec assigns the router ID on the default VRF as the node ID, in the same manner that EIGRP generates its router ID, which is the first IP address on Inspur CN12700 series switches.

The SXP loop detection mechanism drops binding propagation packets based on finding its own node ID in the peer sequence attribute. Changing a node ID in a loop detection-running SXP network could break SXP loop detection functionality and therefore needs to be handled carefully.

The bindings that are associated with the original node ID have to be deleted in all SXP nodes before the new node ID is configured. This can be done by disabling the SXP feature on the network device where you desire to change the node ID. Before you change the node ID, wait until the SXP bindings that are propagated with the particular node ID in the path attribute are deleted.

The node ID configuration is blocked or restricted when SXP is in the enabled state. Router-ID changes in the switch do not affect the SXP node ID, while SXP is enabled. A syslog is generated to indicate that the router ID of the system has changed and this may affect SXP loop detection functionality.

Keepalive and Hold-Time Negotiation with SXPv4

SXP uses a TCP-based, keepalive mechanism to determine if a connection is live. SXPv4 adds an optional negotiated keepalive mechanism within the protocol to provide more predictable and timely detection of connection loss.

SXP connections are asymmetric with almost all of the protocol messages (except for open/open_resp and error messages) sent from an SXP speaker to an SXP listener. The SXP listener can keep a potentially large volume of state per connection, which includes all the binding information learned on a connection. Therefore, it is only meaningful to have a keepalive mechanism that allows a listener to detect the loss of connection with a speaker.

The mechanism is based on two timers:

- **Hold timer**—Used by a listener for detection of elapsing time without successive keepalive or update messages from a speaker
- **Keepalive timer**—Used by a speaker to trigger the dispatch of keepalive messages during intervals when no other information is exported through update messages

The hold-time for the keepalive mechanism may be negotiated during the open or open_resp exchange at connection setup. The following information is important during the negotiation:

- A listener may have desirable range for the hold-time period locally configured or have a default of 90 to 180 seconds. A value of 0xFFFF..0xFFFF indicates that the keepalive mechanism is not used.
- A speaker may have a minimum acceptable hold-time period locally configured or have a default of 120 seconds. This is the shortest period of time a speaker is willing to send keepalive messages for keeping the connection alive. Any shorter hold-time period would require a faster keepalive rate than the rate the speaker is ready to support.
- A value of 0xFFFF implies that the keepalive mechanism is not used.
- The negotiation succeeds when the speaker's minimum acceptable hold-time falls below or within the desirable hold-time range of the listener. If one end turns off the keepalive mechanism, the other end should also turn it off to make the negotiation successful.
- The negotiation fails when the speaker's minimum acceptable hold-time is greater than the upper bound of the listener's hold-time range.
- The selected hold-time period of a successful negotiation is the maximum of the speaker's minimum acceptable hold-time and the lower bound of the listener's hold-time range.
- The speaker calculates the keepalive time to one-third of the selected hold-time by default unless a different keepalive time is locally configured.
- Larger Minimum listener hold-time values are recommended on systems with large number of bindings or connections. Also, these values are recommended if there is a requirement to hold the bindings on the listener during network maintenance events.

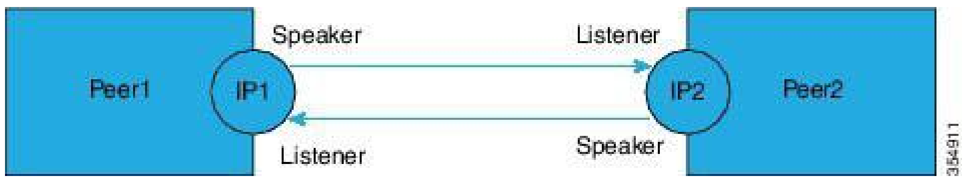
Bidirectional SXP Support Overview

The Bidirectional SXP Support feature enhances the functionality of Inspur TrustSec with SXP version 4 by adding support for SXP bindings that can be propagated in both directions between a speaker and a listener over a single connection.

With the support for bidirectional SXP configuration, a peer can act as both a speaker and a listener and propagate SXP bindings in both directions using a single connection.

The bidirectional SXP configuration is managed with one pair of IP addresses, thereby reducing operational complexity. On either end, only the listener initiates the SXP connection and the speaker accepts the incoming connection.

Figure 17 : Bidirectional SXP Connection



In addition, bidirectional SXP uses the underlying loop-detection benefits of SXPv4 to avoid replay of updates back and forth across the same connection.

Guidelines and Limitations for SXPv4

Inspur TrustSec SXPv4 has the following guidelines and limitations:

- The Bidirectional SXP Support feature enhances the functionality of Inspur TrustSec with SXPv4 by adding support for SXP bindings that can be propagated in both directions between a speaker and a listener over a single connection.
- IPV6 bindings are not learned or transported by the Inspur CN12700 series switches over SXPv4 connections. However, the SXPv4 peering with speakers transporting IPv6 bindings are still supported.
- Inspur CN12700 series switches only expand Subnet-SGT bindings over SXPv3 connections.
- After upgrading to the Inspur CN Release 8.2(3), the default version SXPv4 is advertised by a switch. The appropriate connection versions are re-negotiated with the peers.
- Ensure that there are no overlapping node IDs configured in the network or the node IDs that are configured in the network do not overlap with IP addresses used elsewhere in the network.
- Ensure that there are no overlapping IP addresses to avoid unintentional reuse of default node IDs in the network.
- Before modifying IP addresses in the switch or a router, ensure that the old and the new IP addresses have not been used as default node IDs locally or remotely in the network.
- Ensure that the speaker and listener hold-time values per connection or global or default for each speaker-listener pair are compatible.
- Note that using the hold-time value as 65535 on speaker or listener disables the in-built keepalive mechanism and avoids the staling of bindings upon connectivity loss on SXPv4 devices. Administrative connection resets are required to clear these bindings.
- When migrating existing unidirectional connections to bidirectional connections, ensure that the global hold times are compatible and the bindings learnt in both directions are within the supported scale limits. Also, ensure that the global or default hold-time values on speaker and listener are compatible, since you cannot configure hold-time values for these connections on a per-connection basis.

11.2.12 Inspur TrustSec Subnet-SGT Mapping

Subnet-SGT mapping binds an SGT to all the host addresses of a specified subnet. After this mapping is implemented, Inspur TrustSec imposes SGT on incoming packets having a source IP address that belongs to the specified subnet. This enables you to enforce the Inspur TrustSec policy on the traffic flowing through data center hosts. You can configure IPv4 subnet-SGT bindings under a VRF instance.

In IPv4 networks, SXPv3 and later versions can receive and parse subnet network address or prefix strings from SXPv3 peers.

For example, the IPv4 subnet 198.1.1.0/29 is expanded as follows (only three bits for host addresses):

- Host addresses 198.1.1.1 to 198.1.1.7 are tagged and propagated to the SXP peer.
- Network and broadcast addresses 198.1.1.0 and 198.1.1.8 are not tagged and not propagated.

Subnet bindings are static, which means that active hosts are not learned. They can be used locally for SGT imposition and SGACL enforcement. Packets tagged by subnet-SGT mapping can be propagated on Layer 2 or Layer 3 TrustSec links. Additionally, you can use the **cts sxp allow default-route-sgt** command to enable the transport of SGT bindings through the default route, that is, unknown IP address 0.0.0.0.

11.2.13 SGT Tagging Exemption for Layer 2 Protocols

The Layer 2 (L2) control plane protocols are responsible for creating and maintaining operational states between devices connected through the Inspur TrustSec-enabled links. SGT tagging is enabled by default on Inspur TrustSec-enabled links. A Inspur TrustSec-enabled device applies SGT tags for almost all the L2 packets egressing an interface. The L2 peers on the ingress interfaces process the SGT packets. However, some peers cannot process the SGT-tagged control packets tagged due to limitations. For example, Inspur F3 Series modules do not accept the packets with an SGT tag in the port ingress when the IEEE 802.1Q tag is missing in front of the SGT tag. This causes a peer to drop the L2 control packets such as Inspur Discovery Protocol, Link Level Discovery Protocol (LLDP), Link Aggregation Control Protocol (LACP), or bridge protocol data units (BPDU) with SGT.

From Inspur INOS Release 8.2(3), Inspur TrustSec provides the following enhancements to exempt SGT tagging for the L2 control packets:

1. By default, Inspur INOS assigns null SGT for the L2 control packets even if the device SGT is non-zero.
2. Inspur CN line card modules perform the following action after receiving null SGT and L2 packet from the Supervisor module:

- Inspur CN F Series modules do not tag null SGT for the L2 control packets.
- Inspur CN M Series modules tag null SGT for the L2 control packets. In this case, you can prevent the Inspur CN M series modules from tagging null SGT by using the **no propagate-sgt l2-control** command. This exemption ensures that the L2 control protocols are transmitted without any SGT tags from the Inspur TrustSec-enabled ports.

Use the **no propagate-sgt l2-control** command to exempt the SGT tagging of the L2 control plane protocols for an interface. By default, the SGT tagging is not exempted for the L2 control plane protocols. For example, if the Inspur F3 series module has to interoperate with the Inspur F3 series module by using the Inspur TrustSec enabled link, then enable the **no propagate-sgt l2-control** command for the F3 series module. This ensures that the control packets are accepted by the Inspur F3 series module.

You can also enable or disable the SGT tagging of the L2 control plane protocols under a port profile or a port channel.

11.2.14 Authorization and Policy Acquisition

After authentication ends, the supplicant and AT obtain the security policy from the authentication server. The supplicant and AT enforce the policy against each other. Both the supplicant and AT provide the peer device ID that each receives after authentication. If the peer device ID is not available, Inspur TrustSec can use a manually configured peer device ID.

The authentication server returns the following policy attributes:

Inspur TrustSec Trust

Indicates whether the neighbor device is to be trusted for the purpose of putting the SGT in the packets.

Peer SGT

Indicates the security group that the peer belongs to. If the peer is not trusted, all packets received from the peer are tagged with the SGT configured on the ingress interface. If enforcement is enabled on this interface, the SGACLs that are associated with the peer SGT are downloaded. If the device does not know if the SGACLs are associated with the peer's SGT, the device might send a follow-up request to fetch the SGACLs.

Authorization expiry time

Indicates the number of seconds before the policy expires. The Inspur-proprietary attribute-value (AV) pairs indicate the expiration time of an authorization or policy response to a Inspur TrustSec device. A Inspur TrustSec device should refresh its policy and authorization before it times out.

Tip	Each Inspur TrustSec device should support some minimal default access policy in case it is not able to contact the authentication server to get an appropriate policy for the peer.

11.2.15 Change of Authorization

Inspur TrustSec uses the RADIUS Change of Authorization feature to automatically download policies from Inspur Identity Services Engine (ISE) server to a switch, after an administrator updates the AAA profile on the server.

11.2.16 Environment Data Download

The Inspur TrustSec environment data is a collection of information or policies that assists a device to function as a Inspur TrustSec node. The device acquires the environment data from the authentication server when the device first joins a Inspur TrustSec network cloud, although you might also manually configure some of the data on a device. For example, you must configure the seed Inspur TrustSec device with the authentication server information, which can later be augmented by the server list that the device acquires from the authentication server.

The device must refresh the Inspur TrustSec environment data before it expires. The device can also cache the data and reuse it after a reboot if the data has not expired.

The device uses RADIUS to acquire the following environment data from the authentication server:

Server lists

List of servers that the client can use for future RADIUS requests (for both authentication and authorization)

Device SGT

Security group to which the device itself belongs

Expiry timeout

Interval that controls how often the Inspur TrustSec device should refresh its environment dataRADIUS Relay Functionality

The Inspur INOS device that plays the role of the Inspur TrustSec AT in the 802.1X authentication process has IP connectivity to the authentication server, which allows it to acquire the policy and authorization from the authentication server by exchanging RADIUS messages over UDP/IP. The supplicant device may not have IP connectivity with the authentication server. In such cases, Inspur TrustSec allows the AT to act as a RADIUS relay for the supplicant.

The supplicant sends a special EAP over LAN (EAPOL) message to the Inspur TrustSec AT that contains the RADIUS server IP address and UDP port and the complete RADIUS request. The Inspur TrustSec AT extracts the

RADIUS request from the received EAPOL message and sends it over UDP/IP to the authentication server. When the RADIUS response returns from the authentication server, the Inspur TrustSec AT forwards the message back to the supplicant, encapsulated in an EAPOL frame.

11.2.17 SGT Support for Virtual Port Channel

Effective with Inspur INOS Release 8.2(3), Inspur TrustSec is supported on over Virtual Port Channel (vPC) and vPC+. The following Inspur TrustSec configurations on both vPC or vPC+ peers must be consistent:

- Port-SGT configuration on all interfaces of a vPC (SGT and trust mode)
- IP-SGT configuration
- VLAN-SGT configuration
- SXP peer connections configuration
- SGT caching configuration
- AAA/RADIUS configuration
- SGACL policy configuration
- Enforcing SGACL on VLAN and VRF configuration

11.2.18 Binding Source Priorities

TrustSec resolves conflicts among IP-SGT binding sources with a strict priority scheme. For example, an SGT may be applied to an interface with the **policy** {**dynamic identity** *peer-name* | **static sgt** *tag*} Inspur TrustSec Manual interface mode command (Identity Port Mapping). The current priority enforcement order, from lowest (1) to highest (7), is as follows:

1. Inspur Fabric Services—Inspur TrustSec IP-SGT bindings learned on vPC peer. This is applicable only to vPC peer devices.
2. VLAN-SGT—Bindings learned from snooped ARP or DHCP packets on a VLAN that is configured with a VLAN-SGT mapping.
3. SGT-caching—IP-SGT bindings learned on a VLAN or VRF, where SGT-caching is configured.
4. SXP—Bindings learned from SXP peers.
5. Learned on interface—Bindings of authenticated hosts, which are learned through EPM and device tracking. This type of binding also includes individual hosts that are learned through ARP snooping on L2 [I]PM configured ports.
6. CLI—Address bindings configured using the IP-SGT form of the **cts role-based sgt-map** global configuration command.
7. Port ASIC—SGT bindings derived inline or directly from the port, based on CTS trusted or untrusted configuration.

11.3 Virtualization Support

Inspur TrustSec configuration and operation are local to the virtual device context (VDC). For more information on VDCs, see the Inspur CN12700 Series INOS Virtual Device Context Configuration Guide.

11.4 Licensing Requirements for Inspur TrustSec

The following table shows the licensing requirements for this feature:

Table 21： Licensing Requirements for Inspur TrustSec

Product	License Requirement
Inspur INOS	Beginning with Inspur INOS Release 8.2(3), Inspur TrustSec requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For releases earlier than Inspur INOS 6.1, Inspur TrustSec requires an Advanced Services license. Inspur TrustSec licensing does not have a grace period. You must obtain and install an Advanced Services license before you can use Inspur TrustSec. **Note** For an explanation of the Inspur INOS licensing scheme and how to obtain and apply licenses, see the Inspur INOS Licensing Guide.

11.5 Prerequisites for Inspur TrustSec

Inspur TrustSec has the following prerequisites:

- You must install the Advanced Services license if your device is running a release earlier than Inspur INOS Release 8.2(3).
- You must enable the 802.1X feature before you enable the Inspur TrustSec feature. Although none of the 802.1X interface level features are available, 802.1X is required for the device to authenticate with RADIUS.

11.6 Guidelines and Limitations for Inspur TrustSec

Inspur TrustSec has the following guidelines and limitations:

- Traffic generated from any supervisor is tagged with device-SGT provided that a non-zero value is configured or downloaded and SGT propagation is enabled on the egress interface. However, even if the SGACL enforcement is enabled on the corresponding VRF or VLAN, this traffic would not be subject to SGACL enforcement, if the destination for this traffic is the next hop device.
- The Inspur CN12700 series switch does not support multiple SGACLs for the same source and destination pair. It is recommended that the multi line single SGACL is used.
- Inspur TrustSec MACSec—The following set of requirements must be used when deploying MACSec over SP-provided pseudowire connections. These requirements help to ensure the right service, quality, or characteristics are ordered from the SP.

The Inspur CN12700 series switch supports MACSec over Point-to-Point links, including those using DWDM, as well as non-PtP links such as EoMPLS where the following conditions are met:

- There is no re-ordering or buffering of packets on the MACSec link.
- No additional frames can be injected to the MACSec link.
- There must be end-to-end link event notification—if the edge device or any intermediate device loses a link then there must be notifications sent so that the user is aware of the link failure as the service will be interrupted.
- For MACsec links that have a bandwidth that is greater than or equal to 40G, multiple security associations (SCI/AN pairs) are established with each SA protocol exchange.
- Inspur TrustSec SGT supports IPv4 addressing only.
- Inspur TrustSec SGT in-line tagging is not supported over OTV, VXLAN, FCoE, or Programmable Fabric.
- SXP cannot use the management (mgmt 0) interface.
- You cannot enable Inspur TrustSec on interfaces in half-duplex mode.
- If SGACL is applied to the packets being routed through SVI, SGACL has to be enabled on all the VLANs and the VRF instance involved.
- You cannot configure both Inspur TrustSec and 802.1X on an interface; you can configure only one or the

other. However, you must enable the 802.1X feature for Inspur TrustSec to use EAP-FAST authentication.
- AAA authentication and authorization for Inspur TrustSec is only supported by the Inspur Secure ACS and Inspur ISE.
- To download sgname tables or refresh the environment data, you must use the Inspur ISE Release 8.2(3) or a later release. The Inspur Secure ACS does not support these features.
- Inspur TrustSec supports 200,000 IP-SGT maps. This is subject to the FIB TCAM space availability on each of the modules. Note that the CLI rollback is not supported when more than 100,000 IP-SGT mappings are manually configured.
- The INSPUR-TRUSTSEC-SXP-MIB does not provide an instance number. The object *ctsxSxpConnInstance* does not provide the instance number of the Inspur TrustSec SXP connection. Currently this number is not maintained and cannot be displayed.
- Reloading with Inspur TrustSec configuration on the Non-default VDC triggers a syslog message. When the Inspur TrustSec enforcement is enabled on the VLANs, and if a VDC reload occurs, Inspur TrustSec attempts twice to disable the enforcement on the VLANs. On the second attempt, the following syslog message appears:

```
CTS-2-RBACL_ENFORCEMENT_FAILED:Failed to disable RBACL enf on vdc reload
```

This syslog message can be ignored for the VDC reload because the VLANs are deleted on reload and Inspur TrustSec also deletes the enforcement configurations for those VLANs.
- The Inspur TrustSec configuration commands are not available. The **no cts dev-id pswd dev-pswd** command is currently not supported in INOS software. When the **cts dev-id pass** command is configured, the command configuration can be replaced using the same command, but it cannot be deleted.
- When you change the Inspur TrustSec MACSec port mode from Cache Engine (CE) mode to FabricPath mode, CRC errors are displayed in the Inspur TrustSec MACsec link until native VLAN tagging is disabled on the FabricPath core port. Such configuration changes that occur on a Inspur TrustSec port should be flapped. However, this could cause possible traffic disruptions. In such circumstances, to avoid the display of CRC errors and traffic disruptions, perform the following steps:

1. Disable the cache engine port while having the Inspur TrustSec MACsec enabled.
2. Change the port mode to FabricPath mode.
3. Disable the native VLAN tagging on the FabricPath core port.
4. Enable the port.

- The subnet-to-SGT bindings are not expanded by default. To enable expansion, the **cts sxp mapping network-map**command must be set to a non-zero value.
- An SGT that is associated with a longer prefix is always selected even if a corresponding SGT binding exists. For example, consider the hosts 12.1.0.0/16 with the subnet-SGT binding 10 and 12.1.1.1 with IP-SGT binding 20. SGT 20 is selected for the host 12.1.1.1 even though the parent prefix SGT is 10. Similarly, if VLAN 121 is designated to the subnet 12.1.0.0/16 and configured with a VLAN-SGT binding of 30, host 12.1.1.1 will continue to have the SGT value of 20 and the host 12.1.1.2 will have an SGT value of 10, because the subnet-SGT binding is considered a longer match than a VLAN-SGT mapping.
- To enable the monitoring mode, enable the **cts role-based detailed-logging** command. You can enable or disable logging at the ACE level, as being done currently.
- Monitoring at a per-RBACL or per-ACE level is not supported.
- The monitor mode counter statistics and logging output might not match because the logging output count is rate limited, while counter statistics are directly obtained from the hardware.
- When you enable **monitor all** by using CLI, ISE, or both, the monitoring for all SGT-DGT pairs is turned on, independent of per-pair configuration.
- When you disable the monitor mode feature, the switch reverts to the default behavior. The monitored

SGACLs from ISE will not be installed. All the CLI-installed SGACLS will begin to enforce or deny the policies as configured.

- The traffic hitting SGACL Access Control Entry (ACE) with the log option set is punted to the supervisor, causing network congestion in the supervisor and the packets originated from supervisor such as ping, OSPF hello, and SXP may fail leading to control plane disruption. Therefore, we recommend that you enable log option only for troubleshooting or validation purposes.
- The following guidelines and limitations are applicable for the SGACL Egress Policy Overwrite feature:
- If overlapping RBACL exists from both the sources (CLI and ISE) for an sgt-dgt pair, the respective RBACL is programmed in to the hardware based on the configured priority. The RBACL is programmed as conventional or monitored based on the monitor mode property.
- If RBACL exists only from a single source, irrespective of configured priority, the RBACL is programmed as conventional or monitored based on the monitor mode property.
- Irrespective of the configured priority, RBACL always get updated into the PSS. However, hardware programming is based on the priority and monitor mode property.
- SGACLs are monitored when you enable monitor mode globally and set monitor all. However, based on the install priority set by using the **cts role-based priority-static** command, either the SGACLs downloaded from ISE or the SGACLs configured by using CLI are monitored.
- When SGACL exists only from a single source, that is, either from ISE or CLI, the existing SGACL is used irrespective of the configured install priority of SGACLs.
- When you set **monitor all** by using CLI, ISE, or both, the monitoring for all SGT-DGT pairs is turned on, independent of per-pair configuration.
- Based on the set priority, the monitoring is enabled for the SGACL configured by using CLI or SGACL downloaded from ISE.
- When you disable the monitor mode feature, the switch reverts to the default behavior. The monitored SGACLs from ISE will not be installed. All the CLI-installed SGACLS will begin to enforce or deny the policies as configured.
- The following guidelines and limitations are applicable for the SGACL Egress Policy Overwrite feature:
- Irrespective of whether SGT and DGT are known or unknown for a given network traffic, or an SGACL policy exists for a given SGT and DGT, SGACL policy enforcement disablement on an interface does bypass all SGACLs.
- Per Interface SGACL Bypass feature is configured on an L3 physical interface as well as an L3 port-channel. However, port-channel member ports cannot be configured for this feature.
- SGACL policy enforcement feature is removed from an interface when the IP address is removed.
- When an L3 interface is converted to an L2 interface, the IP configuration is erased. Thereby, the SGACL policy enforcement feature is also erased for the L2 interface.
- When you change a VRF, all L3 configurations are erased on an L3 interface. Thereby, the SGACL policy enforcement feature is also erased for the L3 interface.
- When you enable or disable the Inspur TrustSec SGT Caching feature, by default, Inspur TrustSec reprograms all the RBACLs to add or remove the log option for all the ACEs. Due to this reprogramming, the previously known statistics are deleted for a RBACL and they are not displayed in the **show cts role-based counters** command output.
- The following guidelines and limitations are applicable to SGT tagging exemption for L2 protocols feature:
- You can exempt SGT tagging only on the following control packets by using the **no propagate-sgt l2-control** command:
- Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP)
- IEEE Standard 802.3 Slow Protocols such as Link Aggregation Control Protocol (LACP), Operation, Administration, and Maintenance (OAM), and Link Level Discovery Protocol (LLDP)
- IEEE 802.1X Extensible Authentication Protocol over LAN (EAPOL)
- Inspur Discovery Protocol, Virtual Terminal Protocol (VTP), Dynamic Trunking Protocol (DTP), Port

Aggregation Protocol (PAgP), or Unidirectional Link Detection (UDLD)
- Per VLAN Spanning Tree Plus (PVST+)
- IEEE 802.3 Full Duplex PAUSE Frame
- If the Inspur F3 Series module has to interoperate with the Inspur F3 Series module by using the Inspur TrustSec enabled link, then enable the **no propagate-sgt l2-control** command for the Inspur F3 Series module. This ensures that the control packets are accepted by the Inspur F3 Series module.
- By default, Inspur INOS exempts SGT tagging for any L2 control packets for the Inspur F3series module and Inspur F3 series module because packets are not tagged with null SGT. Therefore, Inspur F3Series modules interoperating with Inspur F3 Series modules or Inspur F3 Series modules interoperating with another Inspur F3 Series modules work without enabling the **no propagate-sgt l2-control** command on the Inspur TrustSec enabled links.
- This table provides information about the support for port interoperability for the Inspur TrustSec-enabled links between the Inspur CN modules:

Table 22 : Support for port interoperability for the Inspur TrustSec-enabled links between the Inspur CN modules

Inspur CN Modules	**Port Interoperability for Inspur TrustSec Enabled Link With SGT Propagation and Without MACSec**	**Port Interoperability for Inspur TrustSec Enabled Link Without SGT Propagation and Without MACSec**	**Port Interoperability for Inspur TrustSec Enabled Link With SGT Propagation and With MACSec**	**Port Interoperability for Inspur TrustSec Enabled Link Without SGT Propagation and With MACSec**
Inspur F3 Series modules	Enable SGT tagging exemption on the Inspur F3 Series module port.	Interoperate by default.	Not interoperable.	Interoperate by default.
Inspur F3 Series modules	Not interoperable because SGT tagging exemption is not supported on Inspur F3 Series modules.	Interoperate by default.	Not interoperable.	Interoperate by default.

11.7 Default Settings for Inspur TrustSec Parameters

This table lists the default settings for Inspur TrustSec parameters.

Table 23 : Default Inspur TrustSec Parameters Settings

Parameter	**Default**

Inspur TrustSec	Disabled
SXP	Disabled
SXP default password	None
SXP reconcile period	120 seconds (2 minutes)
SXP retry period	60 seconds (1 minute)
Caching	Disabled

11.8 Configuring Inspur TrustSec

This section provides information about the configuration tasks for Inspur TrustSec.

11.8.1 Enabling the Inspur TrustSec SGT Feature

You must enable both the 802.1X feature and the Inspur TrustSec feature on the Inspur INOS device before you can configure Inspur TrustSec.

Before you begin

Ensure that you have installed the Advanced Services license, if your device is running a release earlier than Inspur INOS Release 8.2(3).

SUMMARY STEPS

1. **configure terminal**
2. **feature dot1x**
3. **feature cts**
4. **exit**
5. (Optional) **show cts**
6. (Optional) **show feature**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature dot1x** **Example:** `switch(config)# feature dot1x`	Enables the 802.1X feature.
Step 3	**feature cts** **Example:** `switch(config)# feature cts`	Enables the Inspur TrustSec feature.
Step 4	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`switch(config)# exit switch#`	
Step 5	(Optional) **show cts** **Example:** `switch# show cts`	Displays the Inspur TrustSec configuration.
Step 6	(Optional) **show feature** **Example:** `switch# show feature`	Displays the enabled status for features.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

11.8.2 Configuring Inspur TrustSec Device Credentials

You must configure unique Inspur TrustSec credentials on each Inspur TrustSec-enabled Inspur INOS device in your network. Inspur TrustSec uses the password in the credentials for device authentication.

Before you begin

Ensure that you have enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **cts device-id** *name* **password** *password*
3. **exit**
4. (Optional) **show cts**
5. (Optional) **show cts environment**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**cts device-id** *name* **password** *password* **Example:** `switch(config)# cts device-id MyDevice1 password Inspur321`	Configures a unique device ID and password. The *name* argument has a maximum length of 32 characters and is case sensitive. **Note** To remove the configuration of device ID and the password, use the **no** form of the command.
Step 3	**exit** **Example:** `switch(config)# exit switch#`	Exits global configuration mode.

Step 4	(Optional) **show cts** **Example:** `switch# show cts`	Displays the Inspur TrustSec configuration.
Step 5	(Optional) **show cts environment** **Example:** `switch# show cts environment`	Displays the Inspur TrustSec environment data.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .

11.8.3 Configuring Native VLAN Tagging

Configuring Native VLAN Tagging Globally

Perform this task to configure native VLAN tagging globally.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **vlan dot1q tag native** {**fabricpath**} **exclude control**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**vlan dot1q tag native** {**fabricpath**} **exclude control** **Example:** `switch(config)# vlan do1q tag native exclude control`	Tags control and data packets as appropriate. • Use **exclude control** keyword to tag data packets only. • Use **fabricpath** keyword to tag control and data packets on fabricpath ports.

Configuring Native VLAN Tagging on an Interface

Perform this task to configure native VLAN tagging globally.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**

2. **interface** *type slot*/*port*
3. **vlan dot1q tag native {fabricpath} exclude control**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type slot*/*port* **Example:** `switch(config)# interface ethernet 1/4`	Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.
Step 3	**vlan dot1q tag native** {**fabricpath**} **exclude control** **Example:** `switch(config-if)# vlan do1q tag native exclude control`	Tags control and data packets as appropriate. • Use **exclude control** keyword to tag data packets only. • Use **fabricpath** keyword to tag control and data packets on fabricpath ports.

11.8.4 Configuring AAA for Inspur TrustSec

You can use Inspur Secure ACS for Inspur TrustSec authentication. You must configure RADIUS server groups and specify the default AAA authentication and authorization methods on one of the Inspur TrustSec-enabled Inspur INOS devices in your network cloud. Because Inspur TrustSec supports RADIUS relay, you need to configure AAA only on a seed Inspur INOS device that is directly connected to a Inspur Secure ACS. For all the other Inspur TrustSec-enabled Inspur INOS devices, Inspur TrustSec automatically provides a private AAA server group, aaa-private-sg. The seed Inspur INOS devices uses the management virtual routing and forwarding (VRF) instance to communicate with the Inspur Secure ACS.

Configuring AAA on a Seed Inspur INOS Device in a Inspur TrustSec Network

This section describes how to configure AAA on the seed Inspur INOS device in your Inspur TrustSec network cloud.

Before you begin

- Obtain the IPv4 or IPv6 address or hostname for the Inspur Secure ACS.
- Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **key** [**0** | **7**] *key* **pac**
3. (Optional) **show radius-server**
4. **aaa group server radius** *group-name*
5. **server** {*ipv4-address* | *ipv6-address* | *hostname*}
6. **use-vrf** *vrf-name*
7. **exit**
8. **aaa authentication dot1x default group** *group-name*
9. **aaa authorization cts default group** *group-name*
10. **exit**
11. (Optional) **show radius-server groups** [*group-name*]

12. (Optional) **show aaa authentication**
13. (Optional) **show aaa authorization**
14. (Optional) **show cts pacs**
15. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**radius-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **key** [**0** | **7**] *key* **pac** **Example:** `switch(config)# radius-server host 10.10.1.1 key L1a0K2s9 pac`	Configures a RADIUS server host with a key and PAC. The *hostname* argument is alphanumeric, case sensitive, and has a maximum of 256 characters. The *key* argument is alphanumeric, case sensitive, and has a maximum length of 63 characters. The **0** option indicates that the key is in clear text. The **7** option indicates that the key is encrypted. The default is clear text.
Step 3	(Optional) **show radius-server** **Example:** `switch# show radius-server`	Displays the RADIUS server configuration.

Step 4	**aaa group server radius** *group-name*
	Example:
	`switch(config)# aaa group server radius Rad1`
	`switch(config-radius)#`
		Specifies the RADIUS server group and enters RADIUS server group configuration mode.

Step 5	**server** {*ipv4-address* | *ipv6-address* | *hostname*} **Example:** `switch(config-radius)# server 10.10.1.1`	Specifies the RADIUS server host address.
Step 6	**use-vrf** *vrf-name* **Example:** `switch(config-radius)# use-vrf management`	Specifies the management VRF instance for the AAA server group. **Note** If you use the management VRF instance, no further configuration is necessary for the nonseed devices in the network cloud. If you use a different VRF instance, you must configure the nonseed devices with that VRF instance.

Step 7	**exit**
	Example:
	`switch(config-radius)# exit`
	`switch(config)#`
		Exits RADIUS server group configuration mode.

Step 8	**aaa authentication dot1x default group** *group-name*	Specifies the RADIUS server groups to use for 802.1X authentication.

	Command or Action	Purpose
	Example: `switch(config)# aaa authentication dot1x default group Rad1`	
Step 9	**aaa authorization cts default group** *group-name* **Example:** `switch(config)# aaa authentication cts default group Rad1`	Specifies the RADIUS server groups to use for Inspur TrustSec authorization.

Step 10	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 11	(Optional) **show radius-server groups** [*group-name*] **Example:** `switch# show radius-server group rad1`	Displays the RADIUS server group configuration.
Step 12	(Optional) **show aaa authentication** **Example:** `switch# show aaa authentication`	Displays the AAA authentication configuration.
Step 13	(Optional) **show aaa authorization** **Example:** `switch# show aaa authorization`	Displays the AAA authorization configuration.
Step 14	(Optional) **show cts pacs** **Example:** `switch# show cts pacs`	Displays the Inspur TrustSec PAC information.
Step 15	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .
Configuring AAA on Inspur TrustSec Nonseed Inspur INOS Devices .

Configuring AAA on Inspur TrustSec Nonseed Inspur INOS Devices

Inspur TrustSec configures an AAA server group named aaa-private-sg on the nonseed Inspur INOS devices in the network cloud. By default, the aaa-private-sg server group uses the management VRF instance to communicate with the Inspur Secure ACS and no further configuration is required on the nonseed Inspur INOS devices. However, if you choose to use a different VRF instance, you must change the aaa-private-sg on the nonseed Inspur INOS device to use the correct VRF instance.

Before you begin

Ensure that you enabled Inspur TrustSec.
Ensure that you have configured a seed Inspur INOS device in your network.

SUMMARY STEPS

1. **configure terminal**
2. **aaa group server radius aaa-private-sg**
3. **use-vrf** *vrf-name*
4. **exit**
5. (Optional) **show radius-server groups aaa-private-sg**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**aaa group server radius aaa-private-sg**
	Example:
	`switch(config)# aaa group server radius aaa-private-sg`
	`switch(config-radius)#`
		Specifies the RADIUS server group aaa-private-sg and enters RADIUS server group configuration mode.

Step 3	**use-vrf** *vrf-name* **Example:** `switch(config-radius)# use-vrf MyVRF`	Specifies the management VRF instance for the AAA server group.

Step 4	**exit**
	Example:
	`switch(config-radius)# exit`
	`switch(config)#`
		Exits RADIUS server group configuration mode.

Step 5	(Optional) **show radius-server groups aaa-private-sg** **Example:** `switch(config)# show radius-server groups aaa-private-sg`	Displays the RADIUS server group configuration for the default server group.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature.
Configuring AAA on a Seed Inspur INOS Device in a Inspur TrustSec Network.

11.8.5 Configuring Inspur TrustSec Authentication, Authorization, and Data Path Security

This section provides information about the configuration tasks for Inspur TrustSec authentication, authorization, and data path security.

Inspur TrustSec Configuration Process for Inspur TrustSec Authentication and Authorization

Follow these steps to configure Inspur TrustSec authentication and authorization:

Step 1	Enable the Inspur TrustSec feature. See Enabling the Inspur TrustSec SGT Feature
Step 2	Enable Inspur TrustSec authentication. See Enabling Inspur TrustSec Authentication.
Step 3	Enable 802.1X authentication for Inspur TrustSec on the interfaces. See Enabling the 802.1X Feature.

Related Topics

Enabling the Inspur TrustSec SGT Feature
Enabling Inspur TrustSec Authentication .

Enabling Inspur TrustSec Authentication

You must enable Inspur TrustSec authentication on the interfaces. By default, the data path replay protection feature is enabled and the SA protocol operating mode is GCM-encrypt.

Caution	For the Inspur TrustSec authentication configuration to take effect, you must enable and disable the interface, which disrupts traffic on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port* [**-** *port2*]
3. **cts dot1x**
4. (Optional) **no replay-protection**
5. (Optional) **sap modelist** {**gcm-encrypt** | **gcm-encrypt-256** | **gmac** | **no-encap** | **null**}
6. **exit**
7. **shutdown**
8. **no shutdown**
9. **exit**
10. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port* [**-** *port2*]
	
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies a single port or a range of ports and enters interface configuration mode.

Step 3	**cts dot1x** **Example:** `switch(config-if)# cts dot1x`	Enables 802.1X authentication for Inspur TrustSec and enters Inspur TrustSec 802.1X configuration mode.

	Command or Action	Purpose
	`switch(config-if-cts-dot1x)#`	
Step 4	(Optional) **no replay-protection** **Example:** `switch(config-if-cts-dot1x)# no replay-protection`	Disables replay protection. The default is enabled.
Step 5	(Optional) **sap modelist** {**gcm-encrypt** | **gcm-encrypt-256** | **gmac** | **no-encap** | **null**} **Example:** `switch(config-if-cts-dot1x)# sap modelist gcm-encrypt`	Configures the SAP operation mode on the interface. Use the **gcm-encrypt** keyword for GCM encryption. This option is the default. Use the **gcm-encrypt-256** keyword for 256-bit GCM encryption. Use the **gmac** keyword for GCM authentication only. Use the **no-encap** keyword for no encapsulation for SA protocol and no SGT insertion. Use the **null** keyword for encapsulation without authentication or encryption.

Step 6	**exit**
	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`
		Exits Inspur TrustSec 802.1X configuration mode.

Step 7	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 8	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and enables Inspur TrustSec authentication on the interface.

Step 9	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 10	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch(config)# show cts interface all`	Displays the Inspur TrustSec configuration on the interfaces.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature

Configuring Data-Path Replay Protection for Inspur TrustSec on Interfaces and Port Profiles

By default, the Inspur INOS software enables the data-path reply protection feature. You can disable the data-path replay protection feature on the interfaces for Layer 2 Inspur TrustSec if the connecting device does not support SA protocol.

When this task is configured on a port profile, any port profile that joins the group inherits the configuration.

Caution	For the data-path replay protection configuration to take effect, you must enable and disable the interface, which disrupts traffic on the interface.

Before you begin

Ensure that you enabled Inspur TrustSec authentication on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port* [**-** *port2*]
3. **cts dot1x**
4. **no replay-protection**
5. **exit**
6. **shutdown**
7. **no shutdown**
8. **exit**
9. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port* [**-** *port2*]
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies a single port or a range of ports and enters interface configuration mode.

Step 3	**cts dot1x**
	Example:
	`switch(config-if)# cts dot1x`
	`switch(config-if-cts-dot1x)#`
		Enables 802.1X authentication for Inspur TrustSec and enters Inspur TrustSec 802.1X configuration mode.

Step 4	**no replay-protection** **Example:** `switch(config-if-cts-dot1x)# no replay-protection`	Disables data-path replay protection. The default is enabled. Use the **replay-protection** command to enable data-path replay protection on the interface.
Step 5	**exit**	Exits Inspur TrustSec 802.1X configuration mode.

	Command or Action	Purpose

	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`	

Step 6	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 7	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and disables the data-path reply protection feature on the interface.

Step 8	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 9	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch(config)# show cts interface all`	Displays the Inspur TrustSec configuration on the interface.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling Inspur TrustSec Authentication .

Configuring SA Protocol Operation Modes for Inspur TrustSec on Interfaces and Port Profiles

You can configure the SA protocol operation mode on the interfaces for Layer 2 Inspur TrustSec. The default SA protocol operation mode is GCM-encrypt.

When this task is configured on a port profile, any port profile that joins the group inherits the configuration.

Caution	For the SA protocol operation mode configuration to take effect, you must enable and disable the interface, which disrupts traffic on the interface.

Before you begin

Ensure that you enabled Inspur TrustSec authentication on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port* [**-** *port2*]
3. **cts dot1x**
4. **sap modelist** [**gcm-encrypt** | **gcm-encrypt-256** | **gmac** | **no-encap** | **null**]
5. **exit**
6. **shutdown**
7. **no shutdown**

8. **exit**
9. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port* [**-** *port2*]
	
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies a single interface or a range of interfaces and enters interface configuration mode.

Step 3	**cts dot1x**
	
	Example:
	`switch(config-if)# cts dot1x`
	`switch(config-if-cts-dot1x)#`
		Enables 802.1X authentication for Inspur TrustSec and enters Inspur TrustSec 802.1X configuration mode.

Step 4	**sap modelist** [**gcm-encrypt** | **gcm-encrypt-256** | **gmac** | **no-encap** | **null**] **Example:** `switch(config-if-cts-dot1x)# sap modelist gmac`	Configures the SA protocol authentication mode on the interface. Use the **gcm-encrypt** keyword for GCM encryption. This option is the default. Use the **gcm-encrypt-256** keyword for 256-bit GCM encryption. Use the **gmac** keyword for GCM authentication only. Use the **no-encap** keyword for no encapsulation for SA protocol on the interface and no SGT insertion. Use the **null** keyword for encapsulation without authentication or encryption for SA protocol on the interface. Only the SGT is encapsulated.

Step 5	**exit**
	
	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`
		Exits Inspur TrustSec 802.1X configuration mode.

Step 6	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 7	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and SA protocol operation mode on the interface.

	Command or Action	Purpose

Step 8	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 9	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch(config)# show cts interface all`	Displays the Inspur TrustSec configuration on the interface.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling Inspur TrustSec Authentication .

Configuring SGT Propagation for Inspur TrustSec on Interfaces and Port Profiles

The SGT propagation feature on the Layer 2 interface is enabled by default. You can disable the SGT propagation feature on an interface if the peer device connected to the interface cannot handle Inspur TrustSec packets tagged with an SGT.

When this task is configured on a port profile, any port profile that joins the group inherits the configuration.

Caution	For the SGT propagation configuration to take effect, you must enable and disable the interface, which disrupts traffic on the interface.

Before you begin

Ensure that you enabled Inspur TrustSec authentication on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port* [**-** *port2*]
3. **cts dot1x**
4. **no propagate-sgt**
5. **exit**
6. **shutdown**
7. **no shutdown**
8. **exit**
9. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose

	Example:
	`switch# configure terminal`
	`switch(config)#`	

Step 2	**interface ethernet** *slot*/*port* [**-** *port2*]
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies a single port or a range of ports and enters interface configuration mode.

Step 3	**cts dot1x**
	Example:
	`switch(config-if)# cts dot1x`
	`switch(config-if-cts-dot1x)#`
		Enables 802.1X authentication for Inspur TrustSec and enters Inspur TrustSec 802.1X configuration mode.

Step 4	**no propagate-sgt** **Example:** `switch(config-if-cts-dot1x)# no propagate-sgt`	Disables SGT propagation. The default is enabled. Use the **propagate-sgt** command to enable SGT propagation on the interface.

Step 5	**exit**
	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`
		Exits Inspur TrustSec 802.1X configuration mode.

Step 6	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 7	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and disables the data-path reply protection feature on the interface.

Step 8	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 9	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch(config)# show cts interface all`	Displays the Inspur TrustSec configuration on the interface.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling Inspur TrustSec Authentication .

Regenerating SA Protocol Keys on an Interface

You can trigger an SA protocol exchange to generate a new set of keys and protect the data traffic flowing on an interface.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **cts rekey ethernet** *slot*/*port*
2. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**cts rekey ethernet** *slot*/*port* **Example:** `switch# cts rekey ethernet 2/3`	Generates the SA protocol keys for an interface.
Step 2	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch# show cts interface all`	Displays the Inspur TrustSec configuration on the interfaces.

Related Topics

Enabling Inspur TrustSec Authentication.

11.8.6 Configuring Inspur TrustSec Authentication in Manual Mode

You can manually configure Inspur TrustSec on an interface if your Inspur INOS device does not have access to a Inspur Secure ACS or authentication is not needed because you have the MAC address authentication bypass feature enabled. You must manually configure the interfaces on both ends of the connection.

Caution	For the Inspur TrustSec manual mode configuration to take effect, you must enable and disable the interface, which disrupts traffic on the interface.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface slot*/*port*
3. **cts manual**
4. **sap pmk** {*key* [**left-zero-padded**] [**display encrypt**] | **encrypted** *encrypted_pmk* | **use-dot1x**} [**modelist** {**gcm-encrypt** |**gcm-encrypt-256** | **gmac** | **no-encap** | **null**}]
5. (Optional) **policy dynamic identity** *peer-name*
6. (Optional) **policy static sgt** *tag* [**trusted**]
7. **exit**
8. **shutdown**
9. **no shutdown**
10. **exit**
11. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}

12. (Optional) **show cts sap pmk** {**all** | **interface ethernet** *slot*/*port*}
13. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *interface slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies an interface and enters interface configuration mode.

Step 3	**cts manual** **Example:** `switch(config-if)# cts manual switch(config-if-cts-manual)#`	Enters Inspur TrustSec manual configuration mode. **Note** You cannot enable Inspur TrustSec on interfaces in half-duplex mode.
Step 4	**sap pmk** {*key* [**left-zero-padded**] [**display encrypt**] | **encrypted** *encrypted_pmk* | **use-dot1x**} [**modelist** {**gcm-encrypt** |**gcm-encrypt-256** | **gmac** | **no-encap** | **null**}] **Example:** `switch(config-if-cts-manual)# sap pmk fedbaa modelist gmac`	Configures the SA protocol pairwise master key (PMK) and operation mode. SA protocol is disabled by default in Inspur TrustSec manual mode. The *key* argument is a hexadecimal value with an even number of characters and a maximum length of 32 characters. Use the **left-zero-padded** keyword to pad zeros to the left of the entered string if the PMK length is less than 32 bytes. Use the **display encrypt** keyword to specify that the configured PMK be displayed in AES-encrypted format in the running configuration. Use the **encrypted** *encrypted_pmk* keyword to specify an encrypted PMK string of 64 bytes (128 hexadecimal characters). Use the **use-dot1x** keyword when the peer device does not support Inspur TrustSec 802.1X authentication or authorization but does support SA protocol data path encryption and authentication. The mode list configures the cipher mode for the data path encryption and authentication as follows: Use the **gcm-encrypt** keyword for GCM encryption. This option is the default.

	Command or Action	Purpose
		Use the **gcm-encrypt-256** keyword for GCM encryption. Use the **gmac** keyword for GCM authentication. Use the **no-encap** keyword for no encapsulation and no SGT insertion. Use the **null** keyword for encapsulation of the SGT without authentication or encryption.
Step 5	(Optional) **policy dynamic identity** *peer-name* **Example:** `switch(config-if-cts-manual)# policy dynamic identity MyDevice2`	Configures a dynamic authorization policy download. The *peer-name* argument is the Inspur TrustSec device ID for the peer device. The peer name is case sensitive. **Note** Ensure that you have configured the Inspur TrustSec credentials and AAA for Inspur TrustSec. **Note** The **policy dynamic** and **policy static** commands are mutually exclusive. Only one can be applied at a time. To change from one to the other, you must use the **no** form of the command to remove the configuration before configuring the other command.
Step 6	(Optional) **policy static sgt** *tag* [**trusted**] **Example:** `switch(config-if-cts-manual)# policy static sgt 0x2`	Configures a static authorization policy. The *tag* argument is a decimal value or a hexadecimal value in the format **0x***hhhh*. The decimal range is from 2 to 65519, and the hexadecimal range is from 0x2 to 0xffef. The **trusted** keyword indicates that traffic coming on the interface with this SGT should not have its tag overridden. **Note** The **policy dynamic** and **policy static** commands are mutually exclusive. Only one can be applied at a time. To change from one to the other, you must use the **no** form of the command to remove the configuration before configuring the other command.

Step 7	**exit**
	Example:
	`switch(config-if-cts-manual)# exit`
	`switch(config-if)#`
		Exits Inspur TrustSec manual configuration mode.

Step 8	**shutdown**	Disables the interface.

	Command or Action	Purpose
	Example: `switch(config-if)# shutdown`	
Step 9	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and enables Inspur TrustSec authentication on the interface.

Step 10	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 11	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch# show cts interface all`	Displays the Inspur TrustSec configuration for the interfaces.
Step 12	(Optional) **show cts sap pmk** {**all** | **interface ethernet** *slot*/*port*} **Example:** `switch# show cts sap pmk all`	Displays the hexadecimal value of the configured PMK for all interfaces or a specific Ethernet interface.
Step 13	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .

11.8.7 Configuring SGACL Policies

This section provides information about the configuration tasks for SGACL policies.

SGACL Policy Configuration Process

Follow these steps to configure Inspur TrustSec SGACL policies:

Step 1 To improve performance, globally enable SGACL batch programming.

Step 2 For Layer 2 interfaces, enable SGACL policy enforcement for the VLANs with Inspur TrustSec-enabled interfaces.

Step 3 For Layer 3 interfaces, enable SGACL policy enforcement for the VRF instances with Inspur TrustSec-enabled interfaces.

Step 4 If you are not using AAA on a Inspur Secure ACS to download the SGACL policy configuration, manually configure the SGACL mapping and policies.

Enabling SGACL Batch Programming

Perform the following task to enable batching of Security Group Access Control List (SGACL) programming.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **cts role-based policy batched-programming enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	[**no**] **cts role-based policy batched-programming enable**	Enables batching of SGACL programming-related tasks. To disable SGACL batch programming after you have explicitly enabled the feature, use the **no** form of this command.

Enabling SGACL Policy Enforcement on VLANs

If you use SGACLs, you must enable SGACL policy enforcement in the VLANs that have Inspur TrustSec-enabled Layer 2 interfaces.

Before you begin

- Ensure that you enabled Inspur TrustSec.
- Ensure that you enabled SGACL batch programming.

SUMMARY STEPS

1. **configure terminal**
2. **vlan** *vlan-id*
3. **cts role-based enforcement**
4. **exit**
5. (Optional) **show cts role-based enable**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**vlan** *vlan-id*
	
	Example:
	`switch(config)# vlan 10`
	`switch(config-vlan)#`
		Specifies a VLAN and enters VLAN configuration mode.

Step 3	**cts role-based enforcement** **Example:**	Enables Inspur TrustSec SGACL policy enforcement on the VLAN.

	Command or Action	Purpose
	`switch(config-vlan)# cts role-based enforcement`	**Note** If you enable the cts role-based enforcement on a VLAN and no other configuration on ports, the traffic traversing through these ports are subject to (0,0) SGACL. You can either configure this SGACL statically or download it from Inspur ISE.

Step 4	**exit**
	Example:
	`switch(config-vlan)# exit`
	`switch(config)#`
		Saves the VLAN configuration and exits VLAN configuration mode.

Step 5	(Optional) **show cts role-based enable** **Example:** `switch(config)# show cts role-based enable`	Displays the Inspur TrustSec SGACL enforcement configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .

Enabling SGACL Policy Enforcement on VRF Instances

If you use SGACLs, you must enable SGACL policy enforcement in the VRF instances that have Inspur TrustSec-enabled Layer 3 interfaces.

Before you begin

- Ensure that you enabled Inspur TrustSec.
- Ensure that you enabled SGACL batch programming.
- Ensure that you enabled dynamic Address Resolution Protocol (ARP) inspection or Dynamic Host Configuration Protocol (DHCP) snooping.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **cts role-based enforcement**
4. **exit**
5. (Optional) **show cts role-based enable**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.

	Command or Action	Purpose

	Example:
	`switch# configure terminal`
	`switch(config)#`	

Step 2	**vrf context** *vrf-name* **Example:** `switch(config)# vrf context MyVrf switch(config-vrf)#`	Specifies a VRF instance and enters VRF configuration mode.
Step 3	**cts role-based enforcement** **Example:** `switch(config-vrf)# cts role-based enforcement`	Enables Inspur TrustSec SGACL policy enforcement on the VRF instance.

Step 4	**exit**
	Example:
	`switch(config-vrf)# exit`
	`switch(config)#`
		Exits VRF configuration mode.

Step 5	(Optional) **show cts role-based enable** **Example:** `switch(config)# show cts role-based enable`	Displays the Inspur TrustSec SGACL enforcement configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .

Configuring SGACL Logging

Before you begin

Ensure that you have enabled Inspur TrustSec.

Step 1 Enter global configuration mode: switch# **configure terminal**

Step 2 Enable detailed logging for SGACLs:
switch(config)# **cts role-based detailed-logging**

Step 3 Enable detailed logging for the IP access list: switch(config)# **[no] logging ip access-list detailed**

Step 4 (Optional) Change the default value of the logging level such that the ACLLOG SYSLOGs appear using the terminal monitor:

switch(config)# **logging level acllog 6**

Step 5 (Optional) Clear the cache every 15 seconds to limit the cache output to only recent connections: switch(config)# **logging ip access-list cache interval 15**

Step 6 Exit global configuration mode:
switch(config)# **exit**

Step 7 Required: Display information about the detailed logging IP access list and ACE actions: switch# **show logging ip access-list cache detail**

Step 8 (Optional) Display the running configuration for Inspur TrustSec: switch# **show run cts**

Configuring SGACL Logging

This example shows a running configuration, followed by verification commands that display the detailed logging IP access list. The status of the monitor mode and ACE action are highlighted in the output. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts role-based detailed-logging
 logging ip access-list detailed
 logging level acllog 6
 logging ip access-list cache interval 15
.
.
.
switch(config)# sh logging ip access-list cache detail
SGT     Src IP        Dst IP     S-Port    D-Port    Src Intf      Protocol  Monitor
 ACL-Name  ACE-Number ACE-Action      ACL-Direction    ACL-Filter-Type    ACL Applied Intf
    Hits
--------------------------------------------------------------------------------------------
40      4.1.1.2      3.1.1.1     0       0         Ethernet4/11 (1)ICMP    (1 )ON    ----
    -----       Deny          -----          -----             -----           0
   ---------------------------------------------------------------------------------
10      1.1.1.1      2.1.1.2     0       0         Ethernet4/46 (1)ICMP    (1 )ON    ----
    -----       Permit        -----          -----             -----           8

--------------------------------------------------------------------------------
20      2.1.1.2      1.1.1.1     0       0         Ethernet4/34 (1)ICMP   (0 )OFF    ----
    -----       Deny          -----          -----             -----           3

--------------------------------------------------------------------------------
30      3.1.1.1      4.1.1.2     0       0         Ethernet8/48 (1)ICMP   (0 )OFF    ----
    -----       Permit        -----          -----             -----           0

--------------------------------------------------------------------------------

Number of cache entries: 4
--------------------------------------------------------------------------------------------
```

The following example displays detailed logging when **monitor all** is enabled:

```
switch(config)# show logging ip access-list  cache detail
SGT     Src IP        Dst IP     S-Port    D-Port    Src Intf      Protocol
    Monitor ACL-Name  ACE-NumberACE-Action         ACL-Direction ACL-Filter-Type   ACL
    Applied
Intf    Hits
--------------------------------------------------------------------------------------------
26      172.16.2.6       10.1.1.1          0       0         Ethernet6/14 (1)ICMP   (1 )ON
     ----                  -----         Deny           -----              -----
              ----- 20
--------------------------------------------------------------------------------

Number of cache entries: 1
--------------------------------------------------------------------------------------------
```

The following example displays system log:

```
2016 Jan 22 10:48:47 xbow-vdc4 %$ VDC-4 %$ %ACLLOG-6-ACLLOG_FLOW_INTERVAL: Src IP: 172.16.2.6,
 Dst IP: 10.1.1.1, Src Port: 0, Dst Port: 0, Src Intf: Ethernet6/14, Protocol: "ICMP"(1),
Monitor: (1)"ON" , ACL Name: ---, ACE Action: Deny, Appl Intf: ---, Hit-count: 20
```

The following example displays the Inspur TrustSec policy:

```
switch# show cts  role-based  policy

sgt:26
dgt:101 rbacl:test(monitored)
        deny ip log

switch# show running-config  cts

!Command: show running-config cts
!Time: Fri Jan 22 11:01:54 2016

version 7.3(0)D1(1)
feature cts
cts role-based counters enable
cts role-based detailed-logging
cts role-based monitor enable
cts role-based monitor all
cts role-based sgt-map 10.1.1.1 101
cts role-based sgt-map 172.16.2.6 26
cts role-based access-list permit
  permit ip log
cts role-based access-list test
  deny ip log
cts role-based sgt 26 dgt 101 access-list test
cts role-based enforcement

logging level cts 6

switch(config)# show cts role-based counters

RBACL policy counters enabled
Counters last cleared: 01/22/2016 at 10:58:27 AM

sgt:26 dgt:101 [20]
rbacl:test(monitored)
        deny ip log     [20]

switch(config)# show system internal access-list output entries detail  module 6

Flags: F - Fragment entry  E - Port Expansion
       D - DSCP Expansion  M - ACL Expansion
       T - Cross Feature Merge Expansion

                VDC-4 VRF table 1 :
                ====================

INSTANCE 0x0
---------------

  Tcam 0 resource usage:
  ----------------------

  Label_a = 0x200
   Bank 0
   ------
     IPv4  Class
       Policies: Rbacl()
```

```
        Netflow profile: 0
        Netflow deny profile: 0
        Entries:
          [Index] Entry [Stats]
          ---------------------
  [0014:000a:000a] prec 3 permit ip 0.0.0.26/32 0.0.0.101/32 log [0]
  [0015:000b:000b] prec 3 permit ip 0.0.0.0/0 0.0.0.0/0 log [0]
  [0016:000c:000c] prec 3 permit ip 0.0.0.0/0 0.0.0.0/0   [0]


  L4 protocol cam entries usage: none

  No mac protocol cam entries are in use


INSTANCE 0x1
---------------

  Tcam 0 resource usage:
  ----------------------
  Label_a = 0x200
   Bank 0
   ------
     IPv4  Class
       Policies: Rbacl()
       Netflow profile: 0
       Netflow deny profile: 0
       Entries:
         [Index] Entry [Stats]
         ---------------------
  [0014:000a:000a] prec 3 permit ip 0.0.0.26/32 0.0.0.101/32  log  [20]

  [0015:000b:000b] prec 3 permit ip 0.0.0.0/0 0.0.0.0/0 log [0]
  [0016:000c:000c] prec 3 permit ip 0.0.0.0/0 0.0.0.0/0   [0]
```

Configuring SGACL Monitor Mode

Before you begin

- Ensure that you have enabled Inspur TrustSec.
- Ensure that you have enabled counters.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Enable detailed logging for SGACLs:
switch(config)# **cts role-based detailed-logging**

Step 3 Depending on the requirements, perform one of the following actions:

- Enable monitoring mode for all the SGACLs:

  switch(config)# **[no] cts role-based monitor all**

- Enable monitoring for each SGT-DGT pair:

  switch(config)# **[no] cts role-based monitor permissions from** {*sgt*|**unknown**} **to** {*dgt*|**unknown**} [**ipv4**| **ipv6**] Monitoring is enabled for IPv4 Role-Based access control lists (RBACLs) by default. Currently, the IPv6 option is not supported.

Step 4 Required: Display the Inspur TrustSec SGACL policies and details about the monitor mode feature for each pair: switch(config)# **show cts role-based policy**

Step 5 Required: Display the monitoring status of RBACL statistics and lists statistics for all RBACL policies:
switch(config)# **show cts role-based counters**

Note You can also use other **show** commands to display the SGACL syslogs.

Step 6 (Optional) Display the running configuration for Inspur TrustSec:
switch(config)# **show run cts**

Configuring SGACL Monitor Mode

Displaying SGACL Monitor Mode Information

This example shows a running configuration to configure the SGACL monitor mode for SGT 20 to DGT 30. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts role-based detailed-logging
 cts role-based monitor permissions from <20> to <30>
 exit
```

The following example displays the Inspur TrustSec SGACL policies and details about the monitor mode feature for each SGT-DGT pair:

```
switch(config)# sh cts role-based policy

sgt:unknown
dgt:unknown     rbacl:rbacl1
        permit ip log

sgt:10
dgt:20 rbacl:rbacl1(monitored) permit
        ip log

sgt:20
dgt:10  rbacl:rbacl2
        deny ip log

sgt:30
dgt:40  rbacl:rbacl1
        permit ip

sgt:40

dgt:30 rbacl:rbacl2(monitored)
        deny ip

sgt:any
dgt:any rbacl:rbacl1
        permit ip log
```

The following example displays the monitoring status of RBACL statistics and lists the statistics for all the RBACL policies:

```
switch(config)# sh cts role-based counters

RBACL policy counters enabled
Counters last cleared: 12/23/2015 at 01:41:46 AM

sgt:unknown dgt:unknown [0]
rbacl:rbacl1
        permit ip log   [0]

sgt:10 dgt:20   [5]
```

```
rbacl:rbacl1(monitored)
        permit ip log   [5]

sgt:20 dgt:10   [5]
rbacl:rbacl2
        deny ip log     [5]

sgt:30 dgt:40   [0]
rbacl:rbacl1
        permit ip   [0]

sgt:40 dgt:30   [0]
rbacl:rbacl2(monitored)
        deny ip     [0]

sgt:any dgt:any [0]
rbacl:rbacl1
        permit ip log   [0]
```

The following example displays a running configuration for Inspur TrustSec:

```
switch(config)# show run cts

!Command: show running-config cts
!Time: Wed Dec 23 02:01:43 2015

version 7.3(0)D1(1)
feature cts
cts role-based counters enable
cts role-based detailed-logging
cts role-based monitor enable
cts role-based sgt-map 1.1.1.1 10
cts role-based sgt-map 2.1.1.2 20
cts role-based sgt-map 3.1.1.1 30
cts role-based sgt-map 4.1.1.2 40
cts role-based access-list rbacl1
  permit ip log
cts role-based access-list rbacl2
  deny ip log
cts role-based sgt 0 dgt 0 access-list rbacl1
cts role-based sgt 10 dgt 20 access-list rbacl1
cts role-based sgt 20 dgt 10 access-list rbacl2
cts role-based sgt 30 dgt 40 access-list rbacl1
cts role-based sgt 40 dgt 30 access-list rbacl2
cts role-based sgt any dgt any access-list rbacl1
cts role-based monitor permissions from 10 to 20
cts role-based monitor permissions from 40 to 30
cts role-based enforcement
```

The following example displays the running configuration for Inspur TrustSec, that does not include the SGACL logging:

```
switch(config)# show run cts

!Command: show running-config cts
!Time: Wed Dec 23 02:01:43 2015

version 7.3(0)D1(1)
feature cts
cts role-based counters enable
cts role-based detailed-logging
cts role-based monitor enable
cts role-based sgt-map 1.1.1.1 10
cts role-based sgt-map 2.1.1.2 20
```

```
cts role-based sgt-map 3.1.1.1 30
cts role-based sgt-map 4.1.1.2 40
cts role-based access-list rbacl1
  permit ip log
cts role-based access-list rbacl2
  deny ip log
cts role-based access-list rbacl1_no_log
  permit ip
cts role-based access-list rbacl2_no_log
  deny ip
cts role-based sgt 0 dgt 0 access-list rbacl1
cts role-based sgt 10 dgt 20 access-list rbacl1
cts role-based sgt 20 dgt 10 access-list rbacl2
cts role-based sgt 30 dgt 40 access-list rbacl1_no_log
cts role-based sgt 40 dgt 30 access-list rbacl2_no_log
cts role-based sgt any dgt any access-list rbacl1
cts role-based monitor permissions from 10 to 20
cts role-based monitor permissions from 40 to 30
cts role-based enforcement
```

Manually Configuring Inspur TrustSec SGTs

You can manually configure unique Inspur TrustSec security group tags (SGTs) for the packets originating from this device.

Before you begin

Ensure that you have enabled Inspur TrustSec.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Configure the SGT for packets sent from the device: switch(config)# **cts sgt** *tag*
Note The *tag* argument is a decimal value or a hexadecimal value in the format **0x***hhhh*. The decimal range is from 2 to 65519, and the hexadecimal range is from 0x2 to 0xffef.

Step 3 Exit global configuration mode:
switch(config)# **exit**

Step 4 (Optional) Display the Inspur TrustSec environment data information:
switch# **show cts environment-data**

Step 5 (Optional) Copy the running configuration to the startup configuration:
switch# **copy running-config startup-config**

Manually Configuring IPv4-Address-to-SGACL SGT Mapping for a VLAN

You can manually configure an IPv4 address to SGACL SGT mapping on a VLAN if you do not have Inspur Secure ACS, dynamic ARP inspection, or DHCP snooping available on your Inspur INOS device.

Before you begin

- Ensure that you enabled Inspur TrustSec.
- Ensure that you enabled SGACL policy enforcement on the VLAN.

SUMMARY STEPS

1. **configure terminal**
2. **vlan** *vlan-id*
3. **cts role-based sgt-map** *ipv4-address tag*

4. **exit**
5. (Optional) **show cts role-based sgt-map** [**summary** | **sxp peer** *peer-ipv4-addr* | **vlan** *vlan-id* | **vrf** *vrf-name*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**vlan** *vlan-id*
	Example:
	`switch(config)# vlan 10`
	`switch(config-vlan)#`
		Specifies a VLAN and enters VLAN configuration mode.

Step 3	**cts role-based sgt-map** *ipv4-address tag* **Example:** `switch(config-vlan)# cts role-based sgt-map 10.10.1.1 100`	Configures SGT mapping for the SGACL policies for the VLAN.

Step 4	**exit**
	Example:
	`switch(config-vlan)# exit`
	`switch(config)#`
		Saves the VLAN configuration and exits VLAN configuration mode.

Step 5	(Optional) **show cts role-based sgt-map** [**summary** | **sxp peer** *peer-ipv4-addr* | **vlan** *vlan-id* | **vrf** *vrf-name*] **Example:** `switch(config)# show cts role-based sgt-map`	Displays the Inspur TrustSec SGACL SGT mapping configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature
Enabling SGACL Policy Enforcement on VLANs .
Enabling SGACL Policy Enforcement on VRF Instances.

Manually Configuring IPv4-Address-to-SGACL SGT Mapping for a VRF Instance

You can manually configure IPv4-address-to-SGACL SGT mapping on a VRF instance if a Inspur Secure ACS is not available to download the SGACL policy configuration. You can use this feature if you do not have Inspur Secure ACS, dynamic ARP inspection, or DHCP snooping available on your Inspur INOS device.

Before you begin

- Ensure that you enabled Inspur TrustSec.
- Ensure that you enabled SGACL policy enforcement on the VRF instance.
- Ensure that the Layer-3 module is enabled.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **cts role-based sgt-map** *ipv4-address tag*
4. **exit**
5. (Optional) **show cts role-based sgt-map** [**summary** | **sxp peer** *peer-ipv4-addr* | **vlan** *vlan-id* | **vrf** *vrf-name*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**vrf context** *vrf-name*
	Example:
	`switch(config)# vrf context accounting`
	`switch(config-vrf)#`
		Specifies a VRF instance and enters VRF configuration mode.

Step 3	**cts role-based sgt-map** *ipv4-address tag* **Example:** `switch(config-vrf)# cts role-based sgt-map 10.10.1.1 100`	Configures SGT mapping for the SGACL policies for the VLAN.

Step 4	**exit**
	Example:
	`switch(config-vrf)# exit`
	`switch(config)#`
		Exits VRF configuration mode.

Step 5	(Optional) **show cts role-based sgt-map** [**summary** | **sxp peer** *peer-ipv4-addr* | **vlan** *vlan-id* | **vrf** *vrf-name*] **Example:** `switch(config)# show cts role-based sgt-map`	Displays the Inspur TrustSec SGACL SGT mapping configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring VLAN to SGT Mapping

You can map VLANs to SGTs. This procedure is useful for deploying Inspur TrustSec for devices that are VLAN capable but not SGT capable. A host or server can be assigned an SGT based on the assigned VLAN, and any traffic from the VLAN would be marked with the given SGT.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **vlan** *vlan-id*
3. **cts role-based sgt** *sgt-value*
4. **exit**
5. (Optional) **show cts role-based sgt vlan** {**all** | *vlan-id*}
6. (Optional) **show cts role-based sgt-map** [**summary** | **sxp peer** *peer-ipv4-addr* | **vlan** *vlan-id* | **vrf** *vrf-name*]
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal switch(config)#`	Enters global configuration mode.

Step 2	**vlan** *vlan-id*
	Example:
	`switch(config)# vlan 10`
	`switch(config-vlan)#`
		Specifies a VLAN and enters VLAN configuration mode.

Step 3	**cts role-based sgt** *sgt-value* **Example:** `switch(config-vlan)# cts role-based sgt 3`	Maps the VLAN to an SGT. The *sgt-value* argument range is from 1 to 65519.

Step 4	**exit**
	Example:
	`switch(config-vlan)# exit`
	`switch(config)#`
		Saves the VLAN configuration and exits VLAN configuration mode.

Step 5	(Optional) **show cts role-based sgt vlan** {**all** | *vlan-id*} **Example:** `switch(config)# show cts role-based sgt vlan all`	Displays the configured SGT for the specified VLAN.
Step 6	(Optional) **show cts role-based sgt-map** [**summary** | **sxp peer** *peer-ipv4-addr* | **vlan** *vlan-id* | **vrf** *vrf-name*] **Example:** `switch(config)# show cts role-based sgt-map summary`	Displays the SGT mappings.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Manually Configuring SGACL Policies

You can manually configure SGACL policies on your Inspur INOS device if a Inspur Secure ACS is not available to download the SGACL policy configuration.

Before you begin

Ensure that you have enabled Inspur TrustSec.

For Inspur TrustSec logging to function, you must enable Inspur TrustSec counters or statistics. Ensure that you have enabled SGACL policy enforcement on the VLAN and VRF instance.

SUMMARY STEPS

1. **configure terminal**
2. **cts role-based access-list** *list-name*
3. (Optional) {**deny** | **permit**} **all**
4. (Optional) {**deny** | **permit**} **icmp**
5. (Optional) {**deny** | **permit**} **igmp**
6. (Optional) {**deny** | **permit**} **ip**
7. (Optional) {**deny** | **permit**} **tcp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]
8. {**deny** | **permit**} **udp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]
9. **exit**
10. **cts role-based sgt** {*sgt-value* | **any** | **unknown**} **dgt** {*dgt-value* | **any** | **unknown**} **access-list** *list-name*
11. (Optional) **show cts role-based access-list**
12. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**cts role-based access-list** *list-name*
	Example:
	`switch(config)# cts role-based access-list MySGACL`
	`switch(config-rbacl)#`
		Specifies an SGACL and enters role-based access list configuration mode. The *list-name* argument value is alphanumeric, case sensitive, and has a maximum length of 32 characters.

Step 3	(Optional) {**deny** | **permit**} **all** **Example:** `switch(config-rbacl)# deny all`	Denies or permits all traffic.
Step 4	(Optional) {**deny** | **permit**} **icmp** **Example:** `switch(config-rbacl)# permit icmp`	Denies or permits Internet Control Message Protocol (ICMP) traffic.
Step 5	(Optional) {**deny** | **permit**} **igmp** **Example:** `switch(config-rbacl)# deny igmp`	Denies or permits Internet Group Management Protocol (IGMP) traffic.
Step 6	(Optional) {**deny** | **permit**} **ip** **Example:** `switch(config-rbacl)# permit ip`	Denies or permits IP traffic.

	Command or Action	Purpose
Step 7	(Optional) {**deny** | **permit**} **tcp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}] **Example:** `switch(config-rbacl)# deny tcp dst eq 100`	Denies or permits TCP traffic. The default permits all TCP traffic. The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535.
Step 8	{**deny** | **permit**} **udp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}] **Example:** `switch(config-rbacl)# permit udp src eq 1312`	Denies or permits UDP traffic. The default permits all UDP traffic. The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535.

Step 9	**exit**
	Example:
	`switch(config-rbacl)# exit`
	`switch(config)#`
		Exits role-based access-list configuration mode.

Step 10	**cts role-based sgt** {*sgt-value* | **any** | **unknown**} **dgt** {*dgt-value* | **any** | **unknown**} **access-list** *list-name* **Example:** `switch(config)# cts role-based sgt 3 dgt 10 access-list MySGACL`	Maps the SGT values to the SGACL. The *sgt-value* and *dgt-value* argument values range from 0 to 65520. **Note** You must create the SGACL before you can map SGTs to it.
Step 11	(Optional) **show cts role-based access-list** **Example:** `switch(config)# show cts role-based access-list`	Displays the Inspur TrustSec SGACL configuration.
Step 12	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature
Enabling SGACL Policy Enforcement on VLANs .
Enabling SGACL Policy Enforcement on VRF Instances.

Displaying the Downloaded SGACL Policies

After you configure the Inspur TrustSec device credentials and AAA, you can verify the Inspur TrustSec SGACL policies downloaded from the Inspur Secure ACS. The Inspur INOS software downloads the SGACL policies when it learns of a new SGT through authentication and authorization on an interface, from SXP, or from manual IPv4 address to SGACL SGT mapping.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **show cts role-based access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show cts role-based access-list** **Example:** `switch# show cts role-based access-list`	Displays Inspur TrustSec SGACLs, both downloaded from the Inspur Secure ACS and manually configured on the Inspur INOS device.

Related Topics

Enabling the Inspur TrustSec SGT Feature .

Refreshing the Downloaded SGACL Policies

You can refresh the SGACL policies downloaded to the Inspur INOS device by the Inspur Secure ACS.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **cts refresh role-based-policy sgt** {*sgt-value* | **any** | **unknown**}
2. (Optional) **show cts role-based policy**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**cts refresh role-based-policy sgt** {*sgt-value* | **any** | **unknown**}
	Example:
	`switch# cts refresh role-based-policy`
	Example:
	`switch# cts refresh role-based-policy sgt any`
		Refreshes the Inspur TrustSec SGACL policies from the Inspur Secure ACS.
		• **sgt**—Refreshes the egress policy for an SGT.
		• *sgt-value* —Refreshes the egress policy for a specified SGT.
		• **any**—Refreshes the egress policy for any SGT.
		• **unknown**—Refreshes the egress policy for an unknown SGT.

Step 2	(Optional) **show cts role-based policy** **Example:** `switch# show cts role-based policy`	Displays the Inspur TrustSec SGACL policies.

Related Topics

Enabling the Inspur TrustSec SGT Feature .

Refreshing the Environment Data

You can refresh the environment data download from the AAA server.

Before you begin

Ensure that you enabled Inspur TrustSec.
Ensure that you are using the Inspur Identity Services Engine (ISE) Release 8.2(3) or later releases.

SUMMARY STEPS

1. **cts refresh environment-data**
2. **show cts environment-data**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**cts refresh environment-data** **Example:** `switch# cts refresh environment-data`	Refreshes the environment data from the AAA server.
Step 2	**show cts environment-data** **Example:** `switch# show cts environment-data`	Displays the downloaded environment data pertaining to the local device. **Note** The SGT name table entries can be downloaded from the ISE.

Clearing Inspur TrustSec SGACL Policies

You can clear the Inspur TrustSec SGACL policies.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. (Optional) **show cts role-based policy**
2. **clear cts policy** {**all** | **peer** *device-name* | **sgt** *sgt-value*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **show cts role-based policy** **Example:** `switch# clear cts policy all`	Displays the Inspur TrustSec RBACL policy configuration.
Step 2	**clear cts policy** {**all** | **peer** *device-name* | **sgt** *sgt-value*} **Example:** `switch# clear cts policy all`	Clears the policies for Inspur TrustSec connection information.

Related Topics

Enabling the Inspur TrustSec SGT Feature .

11.8.8 Manually Configuring SXP

You can use the SGT Exchange Protocol (SXP) to propagate the SGTs across network devices that do not have hardware support for Inspur TrustSec. This section describes how to configure Inspur TrustSec SXP on Inspur INOS devices in your network.

Inspur TrustSec SXP Configuration Process

Follow these steps to manually configure Inspur TrustSec SXP:

SUMMARY STEPS

1. Enable the Inspur TrustSec feature.
2. Enable SGACL policy enforcement on the VRF instance.
3. Enable Inspur TrustSec SXP.
4. Configure SXP peer connections.

DETAILED STEPS

Step 1 Enable the Inspur TrustSec feature.

Step 2 Enable SGACL policy enforcement on the VRF instance.

Step 3 Enable Inspur TrustSec SXP.

Step 4 Configure SXP peer connections.

Note You cannot use the management (mgmt 0) connection for SXP.

Related Topics

Enabling SGACL Policy Enforcement on VLANs
Enabling SGACL Policy Enforcement on VRF Instances.
Manually Configuring IPv4-Address-to-SGACL SGT Mapping for a VLAN
Manually Configuring SGACL Policies.
Enabling the Inspur TrustSec SGT Feature .
Enabling Inspur TrustSec SXP .
Configuring Inspur TrustSec SXP Peer Connections.

Enabling Inspur TrustSec SXP

You must enable Inspur TrustSec SXP before you can configure peer connections.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **cts sxp enable**
3. **exit**
4. (Optional) **show cts sxp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**cts sxp enable** **Example:**	Enables SXP for Inspur TrustSec.

	`switch(config)# cts sxp enable`	

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show cts sxp** **Example:** `switch# show cts sxp`	Displays the SXP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature.

Configuring Inspur TrustSec SXP Peer Connections

You must configure the SXP peer connection on both the speaker and listener devices. When using password protection, make sure to use the same password on both ends.

Before you begin

Ensure that you enabled Inspur TrustSec. Ensure that you enabled SXP.
Ensure that you enabled RBACL policy enforcement in the VRF instance.

SUMMARY STEPS

1. **configure terminal**
2. **cts sxp connection peer** *peer-ipv4-addr* [**source** *src-ipv4-addr*] **password** {**default** | **none** | **required** *password*} **mode** {**speaker** | **listener** | **local** | **peer** | **speaker**} } [**vrf** *vrf-name*]
3. **exit**
4. (Optional) **show cts sxp connections**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**cts sxp connection peer** *peer-ipv4-addr* [**source** *src-ipv4-addr*] **password** {**default** | **none** | **required** *password*} **mode** {**speaker** | **listener** | **local** | **peer** | **speaker**} } [**vrf** *vrf-name*] **Example:** `switch(config)# cts sxp connection peer 10.10.1.1 source 20.20.1.1 password default mode listener`	Configures the SXP address connection. The **source** keyword specifies the IPv4 address of the source device. The default source is IPv4 address you configured using the **cts sxp default source-ip** command. The **password** keyword specifies the password that SXP should use for the connection using the

	Command or Action	Purpose
		following options: • Use the **default** option to use the default SXP password that you configured using the **cts sxp default password** command. • Use the **none** option to not use a password. • Use the **required** option to use the password specified in the command. • Use the **local** keyword to use the listener as speaker and vice versa • Use the **peer** keyword to use peer device as the SXP listener. The **speaker** and **listener** keywords specify the role of the remote peer device. The **vrf** keyword specifies the VRF instance to the peer. The default is the default VRF instance. • **Note** You cannot use the management (mgmt 0) interface for SXP.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show cts sxp connections** **Example:** `switch# show cts sxp connections`	Displays the SXP connections and their status.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .
Enabling Inspur TrustSec SXP .
Enabling SGACL Policy Enforcement on VRF Instances.

Configuring the Default SXP Password

By default, SXP uses no password when setting up connections. You can configure a default SXP password for the Inspur INOS device.

Before you begin

Ensure that you enabled Inspur TrustSec. Ensure that you enabled SXP.

SUMMARY STEPS

1. **configure terminal**
2. **cts sxp default password** *password*
3. **exit**
4. (Optional) **show cts sxp**
5. (Optional) **show running-config cts**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**cts sxp default password** *password* **Example:** `switch(config)# cts sxp default password A2Q3d4F5`	Configures the SXP default password.
Step 3	**exit** **Example:** `switch(config)# exit switch#`	Exits global configuration mode.
Step 4	(Optional) **show cts sxp** **Example:** `switch# show cts sxp`	Displays the SXP configuration.
Step 5	(Optional) **show running-config cts** **Example:** `switch# show running-config cts`	Displays the SXP configuration in the running configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .
Enabling Inspur TrustSec SXP .

Configuring the Default SXP Source IPv4 Address

The Inspur INOS software uses the default source IPv4 address in all new TCP connections where a source IPv4 address is not specified. When you change the default source IP address, the existing SXP connections are reset and the IP-SGT bindings learned over SXP are cleared. The SXP connections, for which a source IP address has been configured, will continue to use the same IP address, while coming back up.

The SXP connections, for which a source IP address has not been configured, uses the default IP address as the source IP address. Note that for such connections, correct destination IP address configuration on the peer and the reachability to the default source IP address are the required conditions before such connections can become operational. It is recommended to ensure that these conditions are met for existing operational connections, before configuring default source IP address on a device.

Before you begin

Ensure that you enabled Inspur TrustSec.
Ensure that you enabled SXP.

SUMMARY STEPS

1. **configure terminal**
2. **cts sxp default source-ip** *src-ip-addr*
3. **exit**
4. (Optional) **show cts sxp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**cts sxp default source-ip** *src-ip-addr* **Example:** `switch(config)# cts sxp default source-ip 10.10.3.3`	Configures the SXP default source IPv4 address.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show cts sxp** **Example:** `switch# show cts sxp`	Displays the SXP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature . Enabling Inspur TrustSec SXP.

Changing the SXP Reconcile Period

After a peer terminates an SXP connection, an internal hold-down timer starts. If the peer reconnects before the internal hold-down timer expires, the SXP reconcile period timer starts. While the SXP reconcile period timer is active, the Inspur INOS software retains the SGT mapping entries learned from the previous connection and removes invalid entries. The default value is 120 seconds (2 minutes). Setting the SXP reconcile period to 0 seconds disables the timer and causes all entries from the previous connection to be removed.

Before you begin

Ensure that you enabled Inspur TrustSec.
Ensure that you enabled SXP.

SUMMARY STEPS

1. **configure terminal**
2. **cts sxp reconcile-period** *seconds*
3. **exit**
4. (Optional) **show cts sxp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**cts sxp reconcile-period** *seconds* **Example:** `switch(config)# cts sxp reconcile-period 180`	Changes the SXP reconcile timer period. The default value is 120 seconds (2 minutes). The range is from 0 to 64000.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show cts sxp** **Example:** `switch# show cts sxp`	Displays the SXP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .
Enabling Inspur TrustSec SXP .

Changing the SXP Retry Period

The SXP retry period determines how often the Inspur INOS software retries an SXP connection. When an SXP connection is not successfully set up, the Inspur INOS software makes a new attempt to set up the connection after the SXP retry period timer expires. The default value is 60 seconds (1 minute). Setting the SXP retry period to 0 seconds disables the timer and retries are not attempted.

Before you begin

Ensure that you enabled Inspur TrustSec.
Ensure that you enabled SXP.

SUMMARY STEPS

1. **configure terminal**
2. **cts sxp retry-period** *seconds*

3. **exit**
4. (Optional) **show cts sxp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**cts sxp retry-period** *seconds* **Example:** `switch(config)# cts sxp retry-period 120`	Changes the SXP retry timer period. The default value is 60 seconds (1 minute). The range is from 0 to 64000.
Step 3	**exit** **Example:** `switch(config)# exit switch#`	Exits global configuration mode.
Step 4	(Optional) **show cts sxp** **Example:** `switch# show cts sxp`	Displays the SXP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .
Enabling Inspur TrustSec SXP .

Configuring SXPv3

Before you begin

- Ensure that you have enabled Inspur TrustSec.
- Ensure that you have enabled SXP.
- Ensure that you have configured Inspur TrustSec SXP peer connections.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 (Optional) Expand the network limit:

switch(config)# **[no] cts sxp mapping network-map** [*num_bindings*]

Note The *num_bindings* parameter can accept a value from 0 to 65535. The value zero (0) indicates that no expansion is allowed and 65535 is the maximum expansion limit allowed. The default value is zero (0).

Step 3 Configure a subnet-SGT binding:

switch(config)# **cts role-based sgt-map** {*A.B.C.D/<0-32>*} *sgt-number*

Step 4 Required: Display the Inspur TrustSec SXP configuration details:
switch (config)# **show cts sxp**

Step 5 Required: Display the supported SXP version:
switch(config)# **show cts sxp connection**

Example: Configuring SXPv3

This example shows a running configuration, followed by verification commands that display the Inspur TrustSec SXP configuration details and the supported SXP version. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp enable
 cts sxp mapping network-map <64>
 cts role-based sgt-map <10.10.10.10/29> <1032>
.
.
.
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP retry timeout:60
SXP    reconcile    timeout:120
Highest supported SXP version: 3
SXP network-map limit: 64
SXP default-route-SGT transport: Enabled
Unsupported SXP version(s): 2

switch(config)# show cts sxp connection
PEER_IP_ADDR    VRF            PEER_SXP_MODE   SELF_SXP_MODE   CONNECTION STATE    VERSION
30.1.1.3        default         listener        speaker         connected           3
```

Configuring Default Route for SGT Bindings

Before you begin

- Ensure that you have enabled Inspur TrustSec.
- Ensure that you have enabled SXP.
- Ensure that you have configured Inspur TrustSec SXP peer connections.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Required: Enable the default route for the SGT bindings:
switch(config)# **[no] cts sxp allow default-route-sgt**

Step 3 Specify the default route for the SGT bindings for a speaker:
switch(config)# **cts role-based sgt-map** {*0.0.0.0/0*} *sgt-number*

Step 4 Required: Display the Inspur TrustSec SXP configuration details:
switch(config)# **show cts sxp**

Example: Configuring a Default Route for SGT Bindings

This example shows a running configuration, followed by a verification command that displays a Inspur TrustSec SXP configuration details. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp enable
 cts sxp allow default-route-sgt
```

```
 cts role-based sgt-map <0.0.0.0/0> <200>
.
.
.
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP retry timeout:60
SXP reconcile timeout:120
Highest supported SXP version:3
Network Map expansion limit:0
Default Route SGT Propagation: Enabled
Unsupported SXP version(s):2
```

11.8.9 How to Configure SXPv4

Configuring the Node ID of a Network Device

Before you begin

Enable the Inspur TrustSec feature.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Configure the node ID of a network device:

switch(config)# **cts sxp node-id** {*sxp-node-id* | **interface** *interface-type* | *ipv4-address*}
Note Use the **no** form of this command to delete a node ID.

Step 3 Exit global configuration modes:
switch(config)# **exit**

Step 4 (Optional) Display the node ID of a network device by using one of the following commands: switch# **show cts sxp sgt-map**
switch# **show run | include node-id**

switch# **show cts sxp sgt-map detail**

Example: Configuring the Node ID of a Network Device

The following running configuration shows how to configure the node ID of a network device. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp node-id <172.16.1.3>
 exit
```

The following example shows how to configure node ID as an interface.

```
switch(config)# cts sxp node-id interface ethernet 1/1
```

Note that the specified interface should have a valid IP configuration. Otherwise, you cannot configure the node ID. The following example shows how to display the node ID.

```
switch(config)# show cts sxp sgt-map
SXP Node ID(configured):0x00006789

switch(config)# show run | include node-id
cts sxp node-id interface Eth1/1
```

Configuring the Hold-Time for the SXPv4 Protocol on a Network Device

Before you begin

Enable the Inspur TrustSec feature.

Step 1 Enter global configuration mode: switch# **configure terminal**

Step 2 Configure a minimum and maximum acceptable hold-time period in seconds for the listener device:
switch(config)# **cts sxp listener hold-time** *minimum-period maximum-period*
The valid range is from 1-65534 seconds. The default hold-time range for a listener is 90-180 seconds.
Note The maximum-period value must be greater than the minimum-period value.

Step 3 Configure a minimum acceptable hold-time period in seconds for the speaker device:
switch(config)# **cts sxp speaker hold-time** *minimum-period*
The valid range is 1-65534. The default hold-time for a speaker is 120 seconds.

Step 4 Exit global configuration modes:
switch(config)# **exit**

Step 5 (Optional) Display the hold-time configuration value:
switch# **show run | grep speaker**
switch# **show run | grep listener**

Example: Configuring the Hold-Time for the SXPv4 Protocol on a Network Device

The following running configuration shows how to configure the hold-time for the SXPv4 protocol on a listener device. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp listener hold-time <100> <200>
 exit
```

The following running configuration shows how to configure the hold-time for the SXPv4 protocol on a speaker device. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts  sxp  speaker  hold-time  <100>
 exit
```

The following example shows how to display the hold-time configuration values.

```
switch(config)# show run | grep speaker
 cts sxp speaker hold-time 456


switch(config)# show run | grep listener
cts sxp listener hold-time 20 30
```

Configuring the Hold-Time for the SXPv4 Protocol for Each Connection

The peer connection must be configured on both devices. One device is the speaker and the other is the listener. When using password protection, make sure to use the same password on both ends.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Configure a minimum and maximum acceptable hold-time period in seconds for the listener device:

switch(config)# **cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **required** *password*} **mode [[both | local {listener | speaker} | peer {listener | speaker} | listener | speaker] hold-time** *minimum-period maximum-period*] [**vrf** *vrf-name*]] Configures the Inspur TrustSec-SXP peer address connection.

Note A **hold-time** *maximum-period* value is required only when you use the following keywords: **peer speaker** and **local listener**. In other instances, only a **hold-time** *minimum-period* value is required.

The **source** keyword specifies the IPv4 address of the source device. If no address is specified, the connection uses the default source address, if configured, or the address of the port.

The **password** keyword specifies the password that Inspur TrustSec-SXP uses for the connection using the following options:

- **default**—Use the default Inspur TrustSec-SXP password you configured using the **cts sxp default password** command.
- **none**—A password is not used.

The **mode** keyword specifies the role of the remote peer device:

- **both** — The specified mode refers that the device is both the speaker and the listener in the bidirectional SXP connection.
- **local**—The specified mode refers to the local device.
- **peer**—The specified mode refers to the peer device.
- **listener**— Specifies that the peer device is the listener.
- **speaker**— Specifies that the peer device is the speaker.

The **hold-time** keyword allows you to specify the length of the hold-time period for the speaker or listener device. The valid range is from 0-65534 seconds. The value 0 is the global or default hold-time. You can disable the keep-alive mechanism by specifying the maximum hold-time value as 65535. If the **hold-time** option is not specified, the global hold-time value is used. However, if the global hold-time configuration is missing, the default hold-time is used.

Note A **hold-time** *maximum-period* value is required only when you use the following keywords: **peer speaker** and **local listener**. In other instances, only a **hold-time** *minimum-period* value is required.

The optional **vrf** keyword specifies the VRF to the peer. The default is the default VRF.

You cannot use the management (mgmt 0) interface for SXP.

Note The maximum-period value must be greater than or equal to the minimum-period value.

Step 3 Configure a minimum acceptable hold-time period in seconds for the speaker device: switch(config)# **cts sxp speaker hold-time** *minimum-period* The valid range is 1-65534. The default hold-time for a speaker is 120 seconds.

Step 4 Exit global configuration mode: switch(config)# **exit**

Step 5 (Optional) Displays Inspur TrustSec-SXP status and connections: switch# **show cts sxp {connections | sgt-map}** [**detail**| **vrf** *vrf-name*]

Example: Configuring the Hold-Time for the SXPv4 Protocol for Each Connection

Example: Disabling Keep-Alive Mechanism at Listener and Speaker Devices

The following running configuration shows how to configure the hold-time for the SXPv4 protocol for each connection. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp connection peer <10.20.2.2> password default mode local speaker hold-time <500>
 exit
```

The following example shows how to display the hold-time for the SXPv4 protocol for a connection.

```
switch(config)# show run cts | include connection
cts sxp connection peer 1.2.3.4 source 3.4.5.6 password none mode speaker hold-time 113 314
 vrf default

switch-listener(config)# show cts sxp sgt-map detail
SXP Node ID(generated):0x14141409
IP-SGT   Mappings   as   follows:
IPv4,SGT : <1.34.56.45/32 , 119>
Vrf       1
Peer IP  :5.1.1.1
Status   : Active
Seq Num  : 3
Peer Seq :0b0b0b0a
IPv4,SGT : <2.3.11.0/28 , 123>
Vrf       1
Peer IP  :5.1.1.1
Status   : Active
Seq Num  : 3
Peer Seq :0b0b0b0a,0e0e0e01
Total  number  of  IP-SGT  Mappings:  2

switch # show cts sxp connection detail

----------------------------------------------
Peer IP           :3.1.1.2
VRF               :default


PEER MODE         :speaker
Connection State :connected
Version            4
Node ID           :0x0e0e0e01
Capability        :UNKNOWN
Conn Hold Time    :120 seconds
```

The following example shows how to display the hold-time configuration values.

```
switch(config)# show run | grep speaker
cts sxp speaker hold-time 456

switch(config)# show run | grep listener
cts sxp listener hold-time 20 30
```

The following example shows how to disable keep-alive mechanism at listener and speaker devices by configuring maximum values for hold-time.

```
switch# configure terminal
switch(config)# cts sxp connection peer 1.2.3.4 source 3.4.5.6 password none mode speaker
hold-time 65535 65535 vrf default
switch(config)# exit

switch# configure terminal
switch(config)# cts sxp connection peer 4.5.6.7 source 6.7.8.9 password none mode listener
 hold-time 65535 vrf default
switch(config)# exit
```

Configuring Bidirectional SXP Support

Before you begin

Enable the Inspur TrustSec feature.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Configure the Inspur TrustSec SXP peer address connection for a bidirectional SXP configuration:

switch(config)# **cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **required** *password*} **mode both** [**vrf** *vrf-name*]

Note The **both** keyword configures the bidirectional SXP configuration.

Step 3 Exit global configuration mode:
switch(config)# **exit**

Step 4 (Optional) Displays Inspur TrustSec-SXP status and connections:
switch# **show cts sxp {connections | sgt-map}** [**detail**| **vrf** *vrf-name*]

Example: Configuring Bidirectional SXP Support

The following running configuration shows how to configure bidirectional SXP support. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp connection peer <3.3.3.2> source <3.3.3.1> password <none> mode both vrf <default>
Warning: The peer should also be configured as both when this peer is configured as both.
```

The following example shows how to display bidirectional SXP configuration details.

```
switch(config)# show run | include connection
cts sxp connection peer 3.3.3.2 source 3.3.3.1 password none mode both vrf default
```

The following example shows the SXP learnt SGT bindings:

```
switch(config)# show cts sxp sgt-map detail
SXP Node ID(generated):0x00000000
IP-SGT Mappings as follows:
Total number of IP-SGT Mappings: 0
```

Verifying Inspur TrustSec with SXPv4

The following table provides information about how to verify SXPv4 configuration details.

Commands	Purpose
show cts sxp sgt-map vrf *vrf-name*	Displays information about SXP connection.
show cts sxp connection	Displays detailed information about SXP connections.
show cts sxp connection detail	Displays SXP connection for the specified VRF.
show cts sxp connection vrf *vrf-name*	Displays IP address to SGT mapping.
show cts sxp sgt-map	Displays SXP learnt SGT bindings in detail.
show cts sxp sgt-map detail	Displays the SGT mapping for the specified VRF.

11.8.10 Configuring Subnet to SGT Mapping

Before you begin

Ensure that you have enabled Inspur TrustSec.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Configure the subnet to SGT mapping:

switch(config)# **cts role-based sgt-map** {*ip-addr*/*prefix length*} *sgt*

Note The sgt number keyword pair specifies the SGT number that is to be bound to every host address in the specified subnet.

Step 3 Display all the SGT bindings:

switch(config)# **show cts role-based sgt-map**

Step 4 Exit global configuration mode:
switch(config)# **exit**

Configuring Subnet to SGT Mapping

This example shows a running configuration, followed by a verification command that displays all the SGT bindings. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts role-based sgt-map <10.10.10.8/29> <6>
.
.
.
switch(config)# show cts role-based sgt-map
IP ADDRESS                              SGT     VRF/VLAN    SGT CONFIGURATION
10.10.10.8/29                           6        vrf:1      CLI Configured
12.1.0.0/16                             10       vrf:1      CLI Configured
12.1.1.1                                20       vrf:1      CLI Configured
12.1.1.2                                30       vlan:121   CLI Configured
```

11.8.11 Configuring SGT Tagging Exemption for Layer 2 Protocols

Before you begin

Ensure that you have enabled Inspur TrustSec.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Specify an interface or a port channel:
switch(config)# **interface** *interface slot/port*
switch(config)# **interface port-channel** *port-channel*

Step 3 Required: Enter Inspur TrustSec manual configuration mode:
switch(config-if)# **cts manual**

Note You cannot enable Inspur TrustSec on interfaces that are in the half-duplex mode.

Step 4 Enable SGT tagging exemption for the L2 control protocols:
switch(config-if-cts-manual)# **no propagate-sgt l2-control**

Note Use the **propagate-sgt l2-control** command to disable SGT tagging exemption for the L2 control protocols.

Step 5 Exit Inspur TrustSec manual configuration mode, interface configuration mode, and global configuration mode: switch(config-if-cts-manual)# **exit**
switch(config-if)# **exit**

switch(config)# **exit**

Step 6 (Optional) Display the status of SGT tagging for the L2 control protocols:
switch# **show cts propagate-status**

Step 7 (Optional) Display the Inspur TrustSec information for interfaces:
switch# **show cts interface all**

Example: Configuring SGT Tagging Exemption for L2 Protocols

This running configuration shows how to enable SGT tagging exemption for the L2 protocols. Replace the *placeholders* with relevant values for your setup.

```
configure terminal
 interface <Ethernet2/27>
  cts manual
    no propagate-sgt l2-control
    exit
   exit
 exit
```

This running configuration displays the error message when you enable the SGT tagging exemption for the L2 protocols on non-supported modules:

```
configure terminal
 interface <e7/2>
  cts manual
   no propagate-sgt l2-control
ERROR: 'no propagate-sgt l2-control' is not allowed on any port of this line card type.
```

This example displays the status of the SGT tagging for the L2 control protocols on interfaces.

```
switch(config)# show cts propagate-status
Interface: Ethernet2/13
Propagate Exemption:
        Protocols: CDP, LLDP, LACP, EAPoL, BPDUs

Interface: Ethernet2/27
Propagate Exemption:
        Protocols: CDP, LLDP, LACP, EAPoL, BPDUs

switch(config)# show cts interface all
CTS Information for Interface Ethernet2/13:
    CTS is enabled, mode:   CTS_MODE_MANUAL
    IFC state:              CTS_IFC_ST_CTS_OPEN_STATE
    Authentication  Status:  CTS_AUTHC_SKIPPED_CONFIG
      Peer Identity:
      Peer is:              Unknown in manual mode
      802.1X role:          CTS_ROLE_UNKNOWN

      Last Re-Authentication:
    Authorization Status:   CTS_AUTHZ_SKIPPED_CONFIG
      PEER SGT:             0
      Peer SGT assignment:  Not Trusted
    SAP Status:             CTS_SAP_SKIPPED_CONFIG
```

```
      Version:
      Configured pairwise ciphers:
      Replay protection:
      Replay protection mode:
      Selected cipher:
    Propagate SGT: Enabled
        Propagation exempted protocols: CDP, LLDP, LACP, EAPoL, BPDUs

CTS Information for Interface Ethernet2/27:
    CTS is enabled, mode:   CTS_MODE_MANUAL
    IFC state:              CTS_IFC_ST_CTS_OPEN_STATE
    Authentication Status: CTS_AUTHC_SKIPPED_CONFIG
      Peer Identity:
      Peer is:              Unknown in manual mode
      802.1X role:          CTS_ROLE_UNKNOWN
      Last Re-Authentication:
    Authorization Status:   CTS_AUTHZ_SKIPPED_CONFIG
      PEER SGT:             0
      Peer SGT assignment:  Not Trusted
    SAP Status:             CTS_SAP_SKIPPED_CONFIG
      Version:
      Configured pairwise ciphers:
      Replay protection:
      Replay protection mode:
      Selected cipher:
    Propagate SGT: Enabled
        Propagation exempted protocols: CDP, LLDP, LACP, EAPoL, BPDUs
```

11.8.12 Configuring SGACL Egress Policy Overwrite

Use this task to configure SGACL Egress Policy Overwrite feature.

Before you begin

Enable the Inspur TrustSec feature.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Set the install priority for SGACLs:

switch(config)# **[no] cts role-based policy priority-static** *slot/ethernet*
Note By default, the SGACLs configured by using CLI have higher priority in Inspur INOS. Use the **no cts role-based policy priority-static** command to set the install priority for the SGACLs downloaded from ISE.

Step 3 (Optional) Refresh the SGACL policy, if you have upgraded from a release below Inspur INOS Release8.2(3): switch(config)# **cts refresh role-based policy**
Note You need to refresh the SGACL policy, if you have set the SGACL install priority to use the SGACLs downloaded from ISE.

Step 4 Exit the global configuration mode:
switch(config)# **exit**

Step 5 (Optional) Display the Inspur TrustSec SGACL policies and their details:
switch# **show cts role-based policy [configured| downloaded| monitored]** The following information is displayed based on the specified filter:

- **configured** – Displays the SGACLs configured by using CLI.
- **downloaded** – Displays the SGACLs downloaded from ISE.

- **monitored** – Displays the monitored SGACLs.

Step 6 (Optional) Display the monitoring status of RBACL statistics and lists statistics for all policies:
switch# **show cts role-based counters**

Example: Configuring SGACL Egress Policy Overwrite

The following running configuration shows how to set install priority for SGACLs downloaded from ISE.

```
configure terminal
  no cts role-based policy priority-static
  exit
```

The following example displays the SGACL policies.

```
switch# show cts role-based policy
sgt:unknown
dgt:unknown     rbacl:deny_ip(Downloaded,Monitored)
deny ip
sgt:101(101)
dgt:102(102)    rbacl:rb2(Configured)
deny eigrp
sgt:101(101)
dgt:102(102)    rbacl:ise_rbacl_1_ace(Downloaded)
deny gre
```

The following example displays statistics for the enforced SGACLs.

```
switch(config)# show cts role-based counters
RBACL policy counters enabled
Counters last cleared: 08/22/2016 at 09:16:07 AM
sgt:unknown dgt:unknown [0]
rbacl:deny_ip(monitored)
        deny ip [0]
sgt:unknown dgt:2000(2000)      [0]
rbacl:Deny IP(monitored)
        deny ip [0]
sgt:10(10) dgt:20(20)   [0]
rbacl:rb1(monitored)
        deny udp        [0]
        permit tcp      [0]
        deny ip         [0]

rbacl:dummy_test (monitored)
        permit icmp     [0]
        permit tcp      [0]
        permit ip log   [0]
sgt:any dgt:any [0]
rbacl:Permit IP(monitored)
        permit ip       [0]
```

11.8.13 Enabling SGACL Policy Enforcement Per Interface

Use this task to enable SGACL policy enforcement per interface feature.

Before you begin

Enable the Inspur TrustSec feature.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Specify interface or port channel by entering one of the commands:
switch(config)# **interface ethernet** *slot/ethernet*
switch(config)# **interface port-channel** *channel-number*

Step 3 Enable Inspur TrustSec SGACL policy enforcement on the routed interface or port channel:
switch(config-if)# **[no] cts role-based enforcement**

Step 4 Exit interface and global configuration modes:
switch(config-if)# **exit**
switch(config)# **exit**

Step 5 (Optional) Verify that SGACL policy enforcement is disabled on interfaces:
switch# **show cts role-based disabled-interface**

Example: Disabling SGACL Policy Enforcement Per Interface

The following running configuration shows how to disable SGACL policy enforcement per interface for ethernet 1/2. Replace the placeholders with relevant values for your setup.

```
configure terminal
 interface <ethernet 1/2>
  no cts role-based enforcement
  exit
 exit
```

The following example shows how to verify that SGACL policy enforcement is disabled on interfaces.

```
switch# show cts role-based disabled-interface

    Ethernet4/5
    Ethernet4/17
```

11.9 Inspur TrustSec Support on Port-Channel Members

Before Inspur INOS Release 8.2(3), configuration compatibility on port-channel member interfaces with respect to TrustSec configuration was not enforced. Also, Inspur TrustSec configuration was not allowed on port-channel interfaces.

However, from Inspur INOS Release 8.2(3), TrustSec configuration compatibility on port-channel members is enforced and also Trustsec configuration on port-channel interfaces is allowed. The following sections provide more information:

11.9.1 Configuration Models

The following are the configuration models:

- Inspur TrustSec configuration on port-channel interfaces:

Any Inspur TrustSec configuration performed on a port-channel interface is inherited by all its member interfaces.

- Inspur TrustSec configuration on port-channel member interfaces:

Port-channel compatibility parameters are not allowed to be configured on port-channel member interfaces.

Other Inspur TrustSec configurations, such as MACSec configuration, which would not result in incompatibility, are allowed on port-channel member interfaces.

- Adding new members to a port-channel:
- Using the **channel-group** command:

Addition of new members is accepted, if the configuration on the port-channel and that on all members are compatible; if not, the addition is rejected.

- Using the **channel-group force** command:

If the interfaces being added are capable of supporting the port-channel configuration, they inherit the compatibility parameters from the port-channel and the addition is accepted. However, if some interfaces being added are not capable of supporting the port-channel configuration, the addition is rejected.

The following are the updates to the user interfaces after Inspur INOS Release 8.2(3):

- When the **channel group** or **channel-group force** command is issued, if there is any incompatibility in the Inspur TrustSec configuration, an error message is displayed to the user pointing to the incompatible configuration.
- The **show run** and **show start** command displays the Inspur TrustSec configuration on port-channel interfaces as well along with that on physical ethernet interfaces.
- The **show cts role-based sgt-map** command displays the port-sgt learnt mappings that was learnt on the port-channel interface, if applicable.

11.9.2 In-Service Software Upgrades

When In-Service Software Upgrades (ISSU) is performed from a lower version that does not support this feature, as soon as the ISSU is completed, all port-channels inherit the compatibility parameters from their first configured member interface. A warning level syslog is generated for port-channels on which the configuration incompatibility is detected.

11.10 Verifying the Inspur TrustSec Configuration

To display Inspur TrustSec configuration information, use one of the following commands:

Command	Purpose
show cts	Displays Inspur TrustSec information.
show cts capability interface {**all** | **ethernet** *slot*/*port*}	Displays the Inspur TrustSec capability of all interfaces or a specific Ethernet interface.
show cts authorization entries [**interface ethernet** *slot*/*port*.*subinterface*]	Displays the peer-policy data that is downloaded and stored as part of the Inspur TrustSec authorization for all interfaces or a specific Ethernet interface.
show cts credentials	Displays Inspur TrustSec credentials for EAP-FAST.
show cts environment-data	Displays Inspur TrustSec environmental data.
show cts interface {**all** | **brief** | **ethernet** *slot*/*port*}	Displays the Inspur TrustSec configuration for the interfaces.
show cts pacs	Displays Inspur TrustSec authorization information and PACs in the device key store.
show cts role-based access-list	Displays Inspur TrustSec SGACL information.
show cts role-based enable	Displays Inspur TrustSec SGACL enforcement status.
show cts role-based policy [[**dgt** | **sgt**]{*value* | **any** | **unknown**}]	Displays Inspur TrustSec SGACL

Command	Purpose
	policy information for all destination security group tag (DGT) and source security group tag (SGT) pairs or for the specified DGTs or SGTs.
show cts role-based sgt-map [**summary** | **sxp peer** *peer-ipv4-addr* | **vlan** *vlan-id* | **vrf** *vrf-name* | **cached** | **synched**]	Displays the Inspur TrustSec SGACL SGT map configuration. • **summary**—Displays a summary of the SGT mappings. • **sxp peer**—Displays the SGT map configuration for a specific SXP peer. • **vlan**—Displays the SGT map configuration for a specific VLAN. • **vrf**—Displays the SGT map configuration for a specific VRF. • **cached**—Displays SGT maps learnt via caching. • **synched**—Displays SGT maps learnt via Inspur Fabric Services synchronization.
show cts role-based sgt vlan {**all** | *vlan-id*}	Displays the configured SGT for all VLANs or a specific VLAN.
show cts server-list	Displays only the stored list of RADIUS servers available to Inspur TrustSec seed and nonseed devices.
show cts sxp [**connection** | **sgt-map**] [**vrf** *vrf-name*]	Displays Inspur TrustSec SXP information.
show running-config cts	Displays the Inspur TrustSec information in the running configuration.

11.11 Configuration Examples for Inspur TrustSec

This section provides configuration examples for Inspur TrustSec.

11.11.1 Example: Enabling Inspur TrustSec

The following example shows how to enable Inspur TrustSec:

```
feature dot1x
feature cts
```

```
cts device-id device1 password Inspur321
```

11.11.2 Example: Configuring AAA for Inspur TrustSec on a Seed Inspur INOS Device

The following example shows how to configure AAA for Inspur TrustSec on the seed Inspur INOS device:

```
radius-server host 10.10.1.1 key Inspur123
pac aaa group server radius Rad1
  server 10.10.1.1
  use-vrf management
aaa authentication dot1x default group Rad1
aaa authorization cts default group Rad1
```

11.11.3 Example: Enabling Inspur TrustSec Authentication on an Interface

The following example shows how to enable Inspur TrustSec authentication with a clear text password on an interface:

```
interface ethernet 2/1
  cts dot1x
  shutdown
  no shutdown
```

11.11.4 Example: Configuring Inspur TrustSec Authentication in Manual Mode

The following example shows how to configure Inspur TrustSec authentication in manual mode static policy on an interface:

```
interface ethernet 2/1
  cts manual
    sap pmk abcdef modelist gmac
    policy static sgt 0x20
```

The following example shows how to configure Inspur TrustSec authentication in manual mode dynamic policy on an interface:

```
interface ethernet 2/2
  cts manual
    policy dynamic identity device2
```

The following example shows how to specify that the configured PMK be displayed in AES-encrypted format in the running configuration:

```
interface ethernet 2/2
  cts manual
    sap pmk fedbaa display encrypt

show cts sap pmk interface ethernet 2/2
show running-config
```

11.11.5 Example: Configuring Inspur TrustSec Role-Based Policy Enforcement for the Default VRF Instance

The following example shows how to enable Inspur TrustSec role-based policy enforcement for the default VRF instance:

```
cts role-based enforcement
```

11.11.6 Example: Configuring Inspur TrustSec Role-Based Policy Enforcement for a Nondefault VRF

The following example shows how to enable Inspur TrustSec role-based policy enforcement for a nondefault VRF:

```
vrf context test
  cts role-based enforcement
```

11.11.7 Example: Configuring Inspur TrustSec Role-Based Policy Enforcement for a VLAN

The following example shows how to enable Inspur TrustSec role-based policy enforcement for a VLAN:

```
vlan 10
  cts role-based enforcement
```

11.11.8 Example: Configuring IPv4 Address to SGACL SGT Mapping for the Default VRF Instance

The following example shows how to manually configure IPv4 address to SGACL SGT mapping for Inspur TrustSec role-based policies for the default VRF instance:

```
cts role-based sgt-map 10.1.1.1 20
```

11.11.9 Example: Configuring IPv4 Address to SGACL SGT Mapping for a Nondefault VRF Instance

The following example shows how to manually configure IPv4 address to SGACL SGT mapping for Inspur TrustSec role-based policies for a nondefault VRF instance:

```
vrf context test
  cts role-based sgt-map 30.1.1.1 30
```

11.11.10 Example: Configuring IPv4 Address to SGACL SGT Mapping for a VLAN

The following example shows how to manually configure IPv4 address to SGACL SGT mapping for Inspur TrustSec role-based policies for a VLAN:

```
vlan 10
```

```
cts role-based sgt-map 20.1.1.1 20
```

11.11.11 Example: Manually Configuring Inspur TrustSec SGACLs

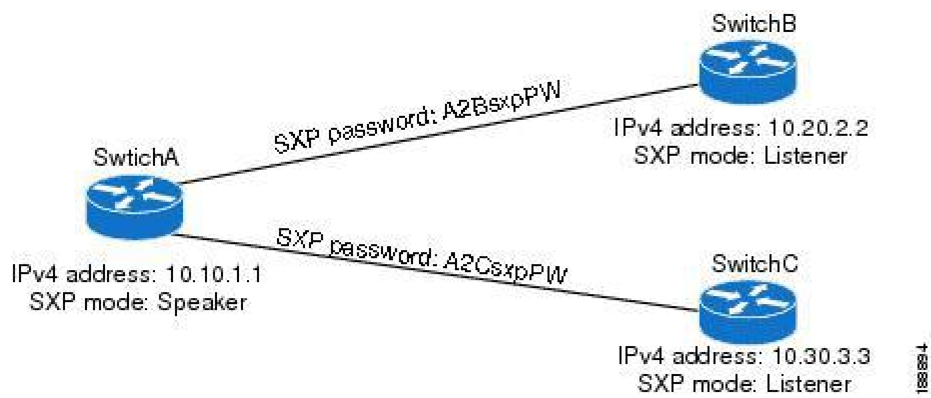
The following example shows how to manually configure Inspur TrustSec SGACLs:

```
cts role-based access-list abcd
  permit icmp
cts role-based sgt 10 dgt 20 access-list abcd
```

11.11.12 Example: Manually Configuring SXP Peer Connections

This figure shows an example of SXP peer connections over the default VRF instance.

Figure 18： Example SXP Peer Connections



The following example shows how to configure the SXP peer connections on SwitchA:

```
feature cts
cts role-based enforcement
cts sxp enable
cts sxp connection peer 10.20.2.2 password required A2BsxpPW mode listener
cts sxp connection peer 10.30.3.3 password required A2CsxpPW mode listener
```

The following example shows how to configure the SXP peer connection on SwitchB:

```
feature cts
cts role-based enforcement
cts sxp enable
cts sxp connection peer 10.10.1.1 password required A2BsxpPW mode speaker
```

The following example shows how to configure the SXP peer connection on SwitchC:

```
feature cts
cts role-based enforcement
cts sxp enable
cts sxp connection peer 10.10.1.1 password required A2CsxpPW mode speaker
```

11.12 Troubleshooting Inspur TrustSec

Problem: Inspur TrustSec commands fail with the following error message:

```
F: ERROR: send failed ret=-1 errno 16
```

Scenario: A VDC is shared between two different Inspur CN modules, such as Inspur F3and F3 Series modules. In this setup, when you configure the IP-SGT mappings beyond the scale limit of a module, responses can be slower than usual. This slow response eventually leads to a configuration command failure, if the configured IP-SGT mappings exceed the module response rate.

Solution: To prevent the Inspur TrustSec command failure, reload the switch by performing the following task:

1. Ensure that the SGACL enforcement configuration is removed for all the VRFs or VLANs from the configuration file or the startup configuration file.
2. Reload the switch.
3. Copy the configuration file to the running configuration.
4. Enable SGACL enforcement by using the **cts role-based enforcement** command on all the required VRFs and VLANs.

11.13 Additional References for Inspur TrustSec

This sections provides additional information related to implementing Inspur TrustSec.

Related Documentation

Related Topic	Document Title
Inspur INOS licensing	*Inspur INOS Licensing Guide*
Command Reference	*Inspur CN12700 Series INOS Security Command Reference*

11.14 Feature History for Inspur TrustSec

This table lists the release history for this feature.

Table 24： Feature History for Inspur TrustSec

Feature Name	Release	Feature Information
SGT Tagging Exemption for Layer 2 Protocols	8.2(3)	Added the functionality to exempt SGT tagging for the L2 control plane protocols. The following commands were introduced: • **no propagate-sgt l2-control** • **show cts propagate-status**
SGACL Policy Enforcement Per Interface	8.2(3)	Added the functionality to enable or disable SGACL policy enforcement on L3 physical interfaces and port-channels.
SGACL Egress Policy Overwrite	8.2(3)	Added the support for the SGACL Egress Policy Overwrite feature.
SXPv4	8.2(3)	Added the support for the SGT Exchange Protocol Version 4.
SGACL Monitoring	8.2(3)	Added the functionality to enable

Feature Name	Release	Feature Information
		monitoring of the SGACLs.
SXPv3	8.2(3)	Added the support for the SGT Exchange Protocol Version 3.
Inspur TrustSec Subnet to SGT Mapping	8.2(3)	Added the support for the Inspur TrustSec Subnet to SGT Mapping.
Inspur TrustSec MACsec over FabricPath on F3	8.2(3)	Added support for Inspur TrustSec MACsec on F3 series modules on FabricPath.
Inspur TrustSec Support on Port-Channel Members	8.2(3)	Added Inspur TrustSec Support o Port-Channel members.
Inspur TrustSec	8.2(3)	Added SGT support for F3 Series modules.
Inspur TrustSec	8.2(3)	Added the ability to map VLANs to SGTs.
Inspur TrustSec	8.2(3)	Added the ability to encrypt the SAP PMK and display the PMK in encrypted format in the running configuration.
Inspur TrustSec	8.2(3)	Added the **show cts sap pmk** command to display the hexadecimal value of the configured PMK.
Inspur TrustSec	8.2(3)	Added the **show cts capability interface** command to display the Inspur TrustSec capability of interfaces.
Inspur TrustSec	8.2(3)	Enabled the **cts sgt**, **policy static sgt**, and **clear cts policy sqt** commands to accept decimal values.
Inspur TrustSec	8.2(3)	Added the ability to download sgname tables from ISE and to refresh the environment data manually and upon environment data timer expiry.
Inspur TrustSec	8.2(3)	Added optional keywords to the **show cts role-based sgt-map** command to display a summary of the SGT mappings or the SGT map configuration for a specific SXP peer, VLAN, or VRF.
Inspur TrustSec	8.2(3)	Added the **brief** keyword to the **show cts interface** command to display a brief summary for all Inspur TrustSec-enabled interfaces.
Inspur TrustSec	8.2(3)	Added SGT support for F3and F3Series modules.
Inspur TrustSec	8.2(3)	Removed the requirement for the Advanced Services license.
Inspur TrustSec	8.2(3)	Added MACsec support for 40G and 100G Series modules.
Inspur TrustSec	8.2(3)	Updated for F3Series modules.

Feature Name	Release	Feature Information
Inspur TrustSec	8.2(3)	Supports pause frame encryption and decryption on interfaces.
SGACL policies	8.2(3)	Supports the enabling or disabling of RBACL logging.
SGACL policies	8.2(3)	Supports the enabling, disabling, monitoring, and clearing of RBACL statistics.
Inspur TrustSec	8.2(3)	No change from Release 8.2(3)

CHAPTER 12 Configuring Inspur TrustSec MACSec

This chapter describes how to configure Inspur TrustSec MACSec on Inspur INOS devices. This chapter includes the following sections:

- Finding Feature Information
- Information About MACsec
- Prerequisites for Inspur TrustSec MACSec
- Licensing Requirements for Inspur TrustSec MACSec
- Default Settings for Inspur TrustSec Parameters
- Feature History for Inspur TrustSec MACSec
- Guidelines and Limitations for Inspur TrustSec MACSec
- Configuring Inspur TrustSec MACSec
- Inspur TrustSec Support on Port-Channel Members
- Verifying the Inspur TrustSec MACSec Configuration
- Additional References for Inspur TrustSec MACSec

12.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

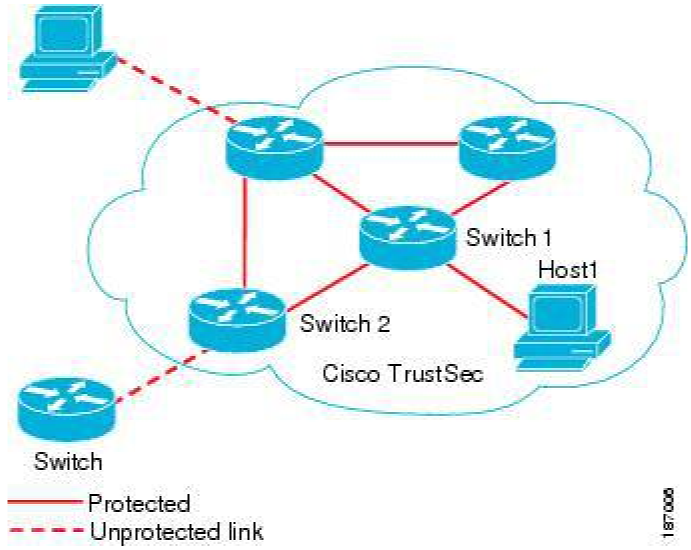
12.2 Information About MACsec

This section provides information about MACsec, and contains the following sections:

12.2.1 Inspur TrustSec Architecture

The Inspur TrustSec security architecture builds secure networks by establishing clouds of trusted network devices. Each device in a cloud is authenticated by its neighbors. Communication on the links between devices in the cloud is secured with a combination of encryption, message integrity checks, and data-path replay protection mechanisms. Inspur TrustSec uses the device and user identification information acquired during authentication for classifying, or coloring, the packets as they enter the network. This packet classification is maintained by tagging packets on ingress to the Inspur TrustSec network so that they can be properly identified for the purpose of applying security and other policy criteria along the data path. The tag, also called the security group tag (SGT), allows the network to enforce the access control policy by enabling the endpoint device to act upon the SGT to filter traffic.

Figure 19： Inspur TrustSec Network Cloud Example



This figure shows an example of a Inspur TrustSec network cloud. In this example, several networking devices and an endpoint device are inside the cloud. One endpoint device and one networking device are outside the cloud because they are not Inspur TrustSec-capable devices or they have been refused access.

The Inspur TrustSec architecture consists of the following major components:

Authentication

Verifies the identity of each device before allowing it to join the Inspur TrustSec network

Authorization

Decides the level of access to the Inspur TrustSec network resources for a device based on its authenticated identity

Access Control

Applies access policies on a per-packet basis using the source tags on each packet

Secure communication

Provides encryption, integrity, and data-path replay protection for the packets that flow over each link in the Inspur TrustSec network

A Inspur TrustSec network has the following entities:

Supplicants

Devices that attempt to join a Inspur TrustSec network

Authenticators (AT)

Devices that are already part of a Inspur TrustSec network

Authorization Server

Servers that might provide authentication information, authorization information, or both

When the link between the supplicant and the AT comes up, the following sequence of events might occur:

Authentication (802.1X)

The authentication server authenticates the supplicant or the authentication is completed if you configure the devices to unconditionally authenticate each other.

Authorization

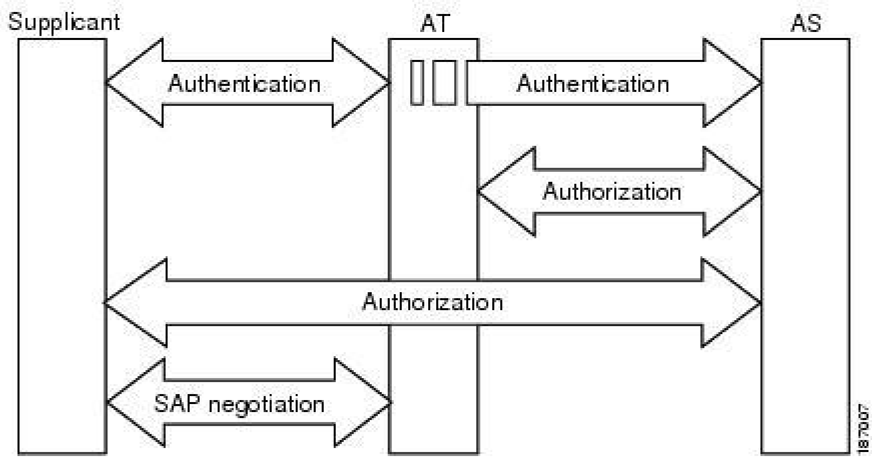
Each side of the link obtains policies, such as SGT and ACLs, that apply to the link. A supplicant might need to use the AT as a relay if it has no other Layer 3 route to the authentication server.

Security Association Protocol Negotiation

The EAPOL-Key exchange occurs between the supplicant and the AT to negotiate a cipher suite, exchange security parameter indexes (SPIs), and manage keys. Successful completion of all three tasks results in the establishment of a security association (SA).

The ports stay in the unauthorized state (blocking state) until the SA protocol negotiation is complete.

Figure 20： SA Protocol Negotiation



This figure shows the SA protocol negotiation, including how the ports stay in unauthorized state until the SA protocol negotiation is complete.

SA protocol negotiation can use any of the following modes of operation:

- Galois/Counter Mode (GCM) encryption
- GCM authentication (GMAC)
- No encapsulation (clear text)
- Encapsulation with no encryption or authentication

Based on the IEEE 802.1AE standard, Inspur TrustSec uses ESP-128 GCM and GMAC.

12.2.2 Authentication

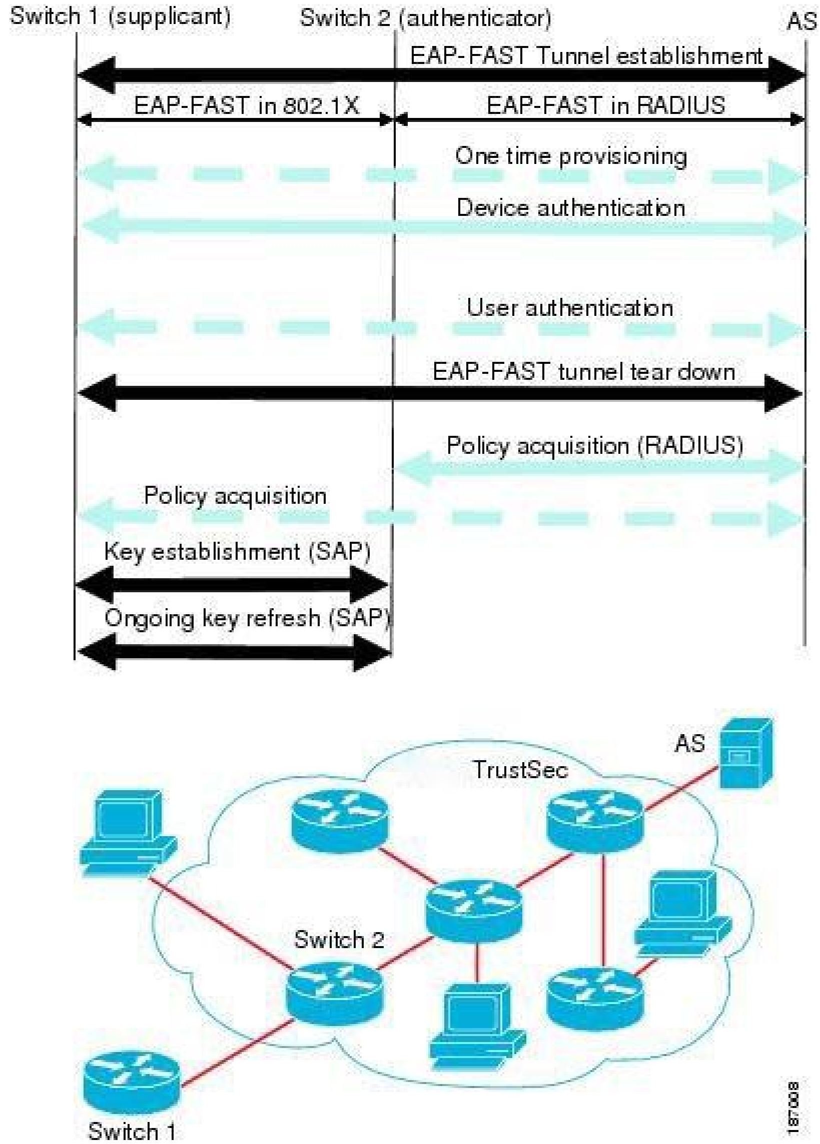
Inspur TrustSec authenticates a device before allowing it to join the network. Inspur TrustSec uses 802.1X authentication with Extensible Authentication Protocol Flexible Authentication through Secure Tunnel (EAP-FAST) as the Extensible Authentication Protocol (EAP) method to perform the authentication.

Inspur TrustSec and Authentication

Inspur TrustSec uses EAP-FAST for authentication. EAP-FAST conversations allow other EAP method exchanges inside the EAP-FAST tunnel using chains, which allows administrators to use traditional user authentication methods, such as Microsoft Challenge Handshake Authentication Protocol Version 2 (MSCHAPv2), while still having security provided by the EAP-FAST tunnel.

Figure 21： Inspur TrustSec Authentication

This figure shows the EAP-FAST tunnel and inner methods used in Inspur TrustSec.



Inspur TrustSec Enhancements to EAP-FAST

The implementation of EAP-FAST for Inspur TrustSec has the following enhancements:

Authenticate the authenticator

Securely determines the identity of the AT by requiring the AT to use its protected access credential (PAC) to derive the shared secret between itself and the authentication server. This feature also prevents you from configuring RADIUS shared secrets on the authentication server for every possible IP address that can be used by the AT.

Notify each peer of the identity of its neighbor

By the end of the authentication exchange, the authentication server has identified the supplicant and the AT. The authentication server conveys the identity of the AT, and whether the AT is Inspur TrustSec-capable, to the supplicant by using additional type-length-value parameters (TLVs) in the protected EAP-FAST termination. The authentication server also conveys the identity of the supplicant and whether the supplicant is Inspur TrustSec-capable to the AT by using RADIUS attributes in the Access-Accept message. Because each peer knows the identity of its neighbor, it can send additional RADIUS Access-Requests to the authentication server to acquire the policy to be applied on the link.

AT posture evaluation

The AT provides its posture information to the authentication server whenever it starts the authentication exchange with the authentication server on behalf of the supplicant.

802.1X Role Selection

In 802.1X, the AT must have IP connectivity with the authentication server because it has to relay the authentication exchange between the supplicant and the AT using RADIUS over UDP/IP. When an endpoint device, such as a PC, connects to a network, it is obvious that it should act as a supplicant. However, in the case of a Inspur TrustSec connection between two network devices, the 802.1X role of each network device might not be immediately apparent to the other network device.

Instead of requiring manual configuration of the AT and supplicant roles for the Inspur INOS devices, Inspur TrustSec runs a role-selection algorithm to automatically determine which Inspur INOS device acts as the AT and which device acts as the supplicant. The role-selection algorithm assigns the AT role to the device that has IP reachability to a RADIUS server. Both devices start both the AT and supplicant state machines. When a Inspur INOS device detects that its peer has access to a RADIUS server, it terminates its own AT state machine and assumes the role of the supplicant. If both Inspur INOS devices have access to a RADIUS server, the algorithm compares the MAC addresses used as the source for sending the EAP over LAN (EAPOL) packets. The Inspur INOS device that has the MAC address with the higher value becomes the AT and the other Inspur INOS device becomes the supplicant.

Inspur TrustSec Authentication Summary

By the end of the Inspur TrustSec authentication process, the authentication server has performed the following actions:

- Verified the identities of the supplicant and the AT
- Authenticated the user if the supplicant is an endpoint device At the end of the Inspur TrustSec authentication process, the AT and the supplicant have the following information:
- Device ID of the peer
- Inspur TrustSec capability information of the peer
- Key used for the SA protocol

Device Identities

Inspur TrustSec does not use IP addresses or MAC addresses as device identities. Instead, assign a name (device ID) to each Inspur TrustSec-capable Inspur INOS device to identify it uniquely in the Inspur TrustSec network. This device ID is used for the following:

- Looking up authorization policy
- Looking up passwords in the databases during authentication

Device Credentials

Inspur TrustSec supports password-based credentials. The authentication servers may use self-signed certificates instead. Inspur TrustSec authenticates the supplicants through passwords and uses MSCHAPv2 to provide mutual authentication even if the authentication server certificate is not verifiable.

The authentication server uses these credentials to mutually authenticate the supplicant during the EAP-FAST phase 0 (provisioning) exchange, where a PAC is provisioned in the supplicant. Inspur TrustSec does not perform the EAP-FAST phase 0 exchange again until the PAC expires and only performs EAP-FAST phase 1 and phase 2 exchanges for future link bringups. The EAP-FAST phase 1 exchange uses the PAC to mutually authenticate the authentication server and the supplicant. Inspur TrustSec uses the device credentials only during the PAC provisioning (or reprovisioning) steps.

The authentication server uses a temporarily configured password to authenticate the supplicant when the supplicant first joins the Inspur TrustSec network. When the supplicant first joins the Inspur TrustSec network, the authentication server authenticates the supplicant using a manufacturing certificate and then generates a strong password and pushes it to the supplicant with the PAC. The authentication server also keeps the new password in its database. The authentication server and the supplicant use this password for mutual authentication in all future EAP-FAST phase 0 exchanges.

User Credentials

Inspur TrustSec does not require a specific type of user credentials for endpoint devices. You can choose any type of authentication method for the user (for example, MSCHAPv2, LEAP, generic token card (GTC), or OTP) and use the corresponding credentials. Inspur TrustSec performs user authentication inside the EAP-FAST tunnel as part of the EAP-FAST phase 2 exchange.

12.2.3 Native VLAN Tagging on Trunk and FabricPath Ports

MACSec is supported over FabricPath through native VLAN tagging on trunk and FabricPath ports feature. Native VLAN tagging can be configured either globally or on an interface for control packets and data packets. Use the following commands to enable native VLAN tagging globally:

- **vlan dot1q tag native exclude control**
- **vlan dot1q tag native fabricpath**
- **vlan dot1q tag native fabricpath exclude control**

Use the following commands to enable native VLAN tagging on FabricPath ports:

- **switchport trunk native vlan tag exclude control**
- **switchport fabricpath native vlan tag**
- **switchport fabricpath native vlan tag exclude control**

Native VLAN tagging provides support for tagged and untagged modes when sending or receiving packets. The following table explains the mode for a packet on a global configuration or port configuration for the above commands.

Tagging Configuration	TX-Control	TX-Data (Native VLAN)	RX-Control	RX-Data
Global trunk port tagging	Untagged	Tagged	Untagged and tagged	Tagged
Global FabricPath tagging	Untagged	Untagged	Untagged and tagged	Untagged and tagged
Global FabricPath tagging for data packets	Untagged	Tagged	Untagged and tagged	Tagged
Port-level trunk port tagging	Untagged	Tagged	Untagged and tagged	Tagged
Port-level Fabricpath tagging	Untagged	Untagged	Untagged and tagged	Untagged and tagged
Port-level FabricPath tagging for data packets	Untagged	Tagged	Untagged and tagged	Tagged

12.2.4 MACsec

MACsec is an IEEE 802.1AE standards based Layer 2 hop-by-hop encryption that provides data confidentiality and integrity for media access independent protocols.

MACsec, provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) Protocol provides the required session keys and manages the required encryption keys.

The 802.1AE encryption with MKA is supported on all types of links, that is, host facing links (links between network access devices and endpoint devices such as a PC or IP phone), or links connected to other switches or routers.

MACsec encrypts the entire data except for the Source and Destination MAC addresses of an Ethernet packet.

To provide MACsec services over the WAN or Metro Ethernet, service providers offer Layer 2 transparent services such as E-Line or E-LAN using various transport layer protocols such as Ethernet over Multiprotocol Label Switching (EoMPLS) and L2TPv3.

The packet body in an EAP-over-LAN (EAPOL) Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). When no MKPDU is received from a participants after 3 hearbeats (each hearbeat is of 2 seconds), peers are deleted from the live peer list. For example, if a client disconnects, the participant on the switch continues to operate MKA until 3 heartbeats have elapsed after the last MKPDU is received from the client.

12.2.5 CTS MACSEC GCM 256-Bit and Extended Packet Sequence Number Support

The SAP GCM cipher suite that is available in the releases earlier than Inspur CN Release 8.2(3), supports 128-bit AES key generation, which is used to encrypt and decrypt data. support for which is introduced in Inspur CN Release 8.2(3), has the capability to encrypt or decrypt data with 256-bit AES key with 64-bit sequence number.

CTS MACsec GCM 256-bit feature, which is an extension of the SAP GCM cipher suite, is introduced in the Inspur CN Release 8.2(3) leverages the 256-bit AES key capability of the hardware.The F3line card has the capability to support the 64-bit sequence number, which is the Extended Packet Sequence Number (XPN). The CTS Manager makes the driver to program the XPN bit in the hardware when GCM-256 encryption mode is enabled. As per XPN standard, the encryption input vector requires the following two fields:

- 32-bit Short Secure Channel Identifier (SSCI)
- 96-bit salt

These fields are constant values for the SAP protocol and are sent by the CTS manager to the driver to enable them to be programmed in the hardware.

12.3 Prerequisites for Inspur TrustSec MACSec

Inspur TrustSec has the following prerequisites:

- You must install the Advanced Services license if your device is running a Inspur INOS release prior to 6.1.
- You must enable the 802.1X feature.
- You must enable the 802.1X feature before you enable the Inspur TrustSec feature. Although none of the 802.1X interface level features are available, 802.1X is required for the device to authenticate with RADIUS.
- You must enable the Inspur TrustSec feature.

12.4 Licensing Requirements for Inspur TrustSec MACSec

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS	Beginning with Inspur INOS Release 8.2(3), Inspur TrustSec requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no

Product	License Requirement
	extra charge to you. For Inspur INOS releases prior to 8.2(3), Inspur TrustSec requires an Advanced Services license. Inspur TrustSec licensing does not have a grace period. You must obtain and install an Advanced Services license before you can use Inspur TrustSec. **Note** For an explanation of the Inspur INOS licensing scheme and how to obtain and apply licenses, see the *Inspur INOS Licensing Guide*.

12.5 Default Settings for Inspur TrustSec Parameters

This table lists the default settings for Inspur TrustSec parameters.

Table 25： Default Inspur TrustSec Parameters Settings

Parameter	Default
Inspur TrustSec	Disabled
SXP	Disabled
SXP default password	None
SXP reconcile period	120 seconds (2 minutes)
SXP retry period	60 seconds (1 minute)
Caching	Disabled

12.6 Feature History for Inspur TrustSec MACSec

This table lists the release history for this feature.

Table 26： Feature History for Inspur TrustSec MACSec

Feature Name	Releases	Feature Information
CTS MACSEC GCM 256-Bit and Extended Packet Sequence Number Support	8.2(3)	Added support for the feature.
Inspur TrustSec MACsec over FabricPath on F3	8.2(3)	Added support for Inspur TrustSec MACsec on F3 series modules on FabricPath.
Inspur TrustSec Support on Port-Channel Members	8.2(3)	Added Inspur TrustSec Support o Port-Channel members.
Inspur TrustSec	8.2(3)	Added the ability to encrypt the SAP PMK and display the PMK in encrypted format in the running configuration.
Inspur TrustSec	8.2(3)	Added the **show cts sap pmk** command to display the hexadecimal value of the configured PMK.
Inspur TrustSec	8.2(3)	Added the **show cts capability interface** command to display the Inspur TrustSec capability of interfaces.
Inspur TrustSec	8.2(3)	Added the **brief** keyword to the **show cts interface** command to display a brief summary for all CTS-enabled interfaces.

Inspur TrustSec	8.2(3)	Added MACsec support for 40G and 100G Series modules.
Inspur TrustSec	8.2(3)	No change from Release 8.2(3).

12.7 Guidelines and Limitations for Inspur TrustSec MACSec

Inspur TrustSec has the following guidelines and limitations:

Inspur TrustSec MACSec—The following set of requirements must be used when deploying MACSec over SP-provided pseudowire connections. These requirements help to ensure the right service, quality, or characteristics are ordered from the SP.

The CN12700 supports MACSec over Point-to-Point links, including those using DWDM, as well as non-PtP links such as EoMPLS where the following conditions are met:

- There is no re-ordering or buffering of packets on the MACSec link.
- No additional frames can be injected to the MACSec link.
- There must be end-to-end link event notification—if the edge device or any intermediate device loses a link then there must be notifications sent so that the customer is aware of the link failure as the service will be interrupted.

For MACSec links that have a bandwidth that is greater than or equal to 40G, multiple security associations (SCI/AN pairs) are established with each Security Association Protocol (SAP) exchange.

When you change the CTS MACSec port mode from Cache Engine (CE) mode to FabricPath mode, CRC errors are displayed in the CTS MACSec link until native VLAN tagging is disabled on the FabricPath core port. Such configuration changes that occur on a CTS port should be flapped. However, this could cause possible traffic disruptions. In such circumstances, to avoid the display of CRC errors and traffic disruptions, perform the following steps:

- Disable the cache engine port while having the CTS MACSec enabled.
- Change the port mode to FabricPath mode.
- Disable the native VLAN tagging on the FabricPath core port.
- Enable the port.

When the F3 line card interoperates with older line cards, the user must configure only the legacy modes on the F3 line card for the link to be up. The configuration on both the peers must be consistent. On older line cards, the GCM-256 bit option is prevented because capability is not available.

On F3line cards when MACSEC is enabled on a port with 1G operating speed, all MACSEC dropped packets will be reported as CRC error packets in addition to the actual CRC packets. This is a known limitation.

Inspur CN12700 Series Switches has the debounce timer feature to delay the notification of link change, which can decrease traffic loss due to network reconfiguration. This feature affects the CTS MACSec and if delays on links are higher, the MACSec-enabled links may not come up. To bring the link up, increase the value of debounce timer link down from its default value 100.

12.8 Configuring Inspur TrustSec MACSec

This section provides information about the configuration tasks for Inspur TrustSec MACSec.

12.8.1 Enabling the Inspur TrustSec MACSec Feature

You must enable both the 802.1X feature and the Inspur TrustSec feature on the Inspur INOS device before you can configure Inspur TrustSec MACSec feature.

SUMMARY STEPS

1. **configure terminal**
2. **feature dot1x**
3. **feature cts**
4. **exit**
5. (Optional) **show cts**
6. (Optional) **show feature**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**feature dot1x** **Example:** `switch(config)# feature dot1x`	Enables the 802.1X feature.
Step 3	**feature cts** **Example:** `switch(config)# feature cts`	Enables the Inspur TrustSec feature.

Step 4	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 5	(Optional) **show cts** **Example:** `switch# show cts`	Displays the Inspur TrustSec configuration.
Step 6	(Optional) **show feature** **Example:** `switch# show feature`	Displays the enabled status for features.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

12.8.2 Configuring Inspur TrustSec Device Credentials

You must configure unique Inspur TrustSec credentials on each Inspur TrustSec-enabled Inspur INOS device in your network. Inspur TrustSec uses the password in the credentials for device authentication.

Before you begin

Ensure that you have enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **cts device-id** *name* **password** *password*
3. **exit**
4. (Optional) **show cts**
5. (Optional) **show cts environment**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**cts device-id** *name* **password** *password* **Example:** `switch(config)# cts device-id MyDevice1 password Inspur321`	Configures a unique device ID and password. The *name* argument has a maximum length of 32 characters and is case sensitive. **Note** To remove the configuration of device ID and the password, use the **no** form of the command.

Step 3	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	(Optional) **show cts** **Example:** `switch# show cts`	Displays the Inspur TrustSec configuration.
Step 5	(Optional) **show cts environment** **Example:** `switch# show cts environment`	Displays the Inspur TrustSec environment data.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .

12.8.3 Configuring Native VLAN Tagging

Configuring Native VLAN Tagging Globally

Perform this task to configure native VLAN tagging globally.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **vlan dot1q tag native** {**fabricpath**} **exclude control**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**vlan dot1q tag native** {**fabricpath**} **exclude control** **Example:** `switch(config)# vlan do1q tag native exclude control`	Tags control and data packets as appropriate. • Use **exclude control** keyword to tag data packets only. • Use **fabricpath** keyword to tag control and data packets on fabricpath ports.

Configuring Native VLAN Tagging on an Interface

Perform this task to configure native VLAN tagging globally.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **vlan dot1q tag native {fabricpath} exclude control**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type slot/port* **Example:** `switch(config)# interface ethernet 1/4`	Specifies the interface that you want to add to a channel group, and enters the interface configuration mode.
Step 3	**vlan dot1q tag native** {**fabricpath**} **exclude control** **Example:** `switch(config-if)# vlan do1q tag native exclude control`	Tags control and data packets as appropriate. • Use **exclude control** keyword to tag data packets only. • Use **fabricpath** keyword to tag control and data packets on fabricpath ports.

12.8.4 Configuring Inspur TrustSec Authentication, Authorization, and Data Path Security

This section provides information about the configuration tasks for Inspur TrustSec authentication, authorization, and data path security.

Inspur TrustSec Configuration Process for Inspur TrustSec Authentication and Authorization

Follow these steps to configure Inspur TrustSec authentication and authorization:

Step 1 Enable the Inspur TrustSec feature. See Enabling the Inspur TrustSec SGT Feature ,.

Step 2 Enable Inspur TrustSec authentication. See Enabling Inspur TrustSec Authentication ,.

Step 3 Enable 802.1X authentication for Inspur TrustSec on the interfaces. See Enabling the 802.1X Feature.

Related Topics

Enabling the Inspur TrustSec SGT Feature
Enabling Inspur TrustSec Authentication .

Enabling Inspur TrustSec Authentication

You must enable Inspur TrustSec authentication on the interfaces. By default, the data path replay protection feature is enabled and the SA protocol operating mode is GCM-encrypt.

Caution	For the Inspur TrustSec authentication configuration to take effect, you must enable and disable the interface, which disrupts traffic on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port* [**-** *port2*]
3. **cts dot1x**
4. (Optional) **no replay-protection**
5. (Optional) **sap modelist** {**gcm-encrypt** | **gcm-encrypt-256** | **gmac** | **no-encap** | **null**}
6. **exit**
7. **shutdown**
8. **no shutdown**
9. **exit**
10. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port* [**-** *port2*]
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies a single port or a range of ports and enters interface configuration mode.

Step 3	**cts dot1x**	Enables 802.1X authentication for Inspur

	Command or Action	Purpose

	Example:
	`switch(config-if)# cts dot1x`
	`switch(config-if-cts-dot1x)#`
		TrustSec and enters Inspur TrustSec 802.1X configuration mode.

Step 4	(Optional) **no replay-protection** **Example:** `switch(config-if-cts-dot1x)# no replay-protection`	Disables replay protection. The default is enabled.
Step 5	(Optional) **sap modelist** {**gcm-encrypt** | **gcm-encrypt-256** | **gmac** | **no-encap** | **null**} **Example:** `switch(config-if-cts-dot1x)# sap modelist gcm-encrypt`	Configures the SAP operation mode on the interface. Use the **gcm-encrypt** keyword for GCM encryption. This option is the default. Use the **gcm-encrypt-256** keyword for 256-bit GCM encryption. Use the **gmac** keyword for GCM authentication only. Use the **no-encap** keyword for no encapsulation for SA protocol and no SGT insertion. Use the **null** keyword for encapsulation without authentication or encryption.

Step 6	**exit**
	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`
		Exits Inspur TrustSec 802.1X configuration mode.

Step 7	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 8	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and enables Inspur TrustSec authentication on the interface.

Step 9	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 10	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch(config)# show cts interface all`	Displays the Inspur TrustSec configuration on the interfaces.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .

Configuring Data-Path Replay Protection for Inspur TrustSec on Interfaces and Port Profiles

By default, the Inspur INOS software enables the data-path reply protection feature. You can disable the data-path replay protection feature on the interfaces for Layer 2 Inspur TrustSec if the connecting device does not support SA protocol.

When this task is configured on a port profile, any port profile that joins the group inherits the configuration.

Caution	For the data-path replay protection configuration to take effect, you must enable and disable the interface, which disrupts traffic on the interface.

Before you begin

Ensure that you enabled Inspur TrustSec authentication on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port* [**-** *port2*]
3. **cts dot1x**
4. **no replay-protection**
5. **exit**
6. **shutdown**
7. **no shutdown**
8. **exit**
9. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port* [**-** *port2*]
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies a single port or a range of ports and enters interface configuration mode.

Step 3	**cts dot1x**
	Example:
	`switch(config-if)# cts dot1x`
	`switch(config-if-cts-dot1x)#`
		Enables 802.1X authentication for Inspur TrustSec and enters Inspur TrustSec 802.1X configuration mode.

Step 4	**no replay-protection** **Example:** `switch(config-if-cts-dot1x)# no replay-protection`	Disables data-path replay protection. The default is enabled. Use the **replay-protection** command to enable data-path replay protection on the interface.

	Command or Action	Purpose

Step 5	**exit**
	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`
		Exits Inspur TrustSec 802.1X configuration mode.

Step 6	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 7	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and disables the data-path reply protection feature on the interface.

Step 8	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 9	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch(config)# show cts interface all`	Displays the Inspur TrustSec configuration on the interface.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling Inspur TrustSec Authentication .

Configuring SA Protocol Operation Modes for Inspur TrustSec on Interfaces and Port Profiles

You can configure the SA protocol operation mode on the interfaces for Layer 2 Inspur TrustSec. The default SA protocol operation mode is GCM-encrypt.

When this task is configured on a port profile, any port profile that joins the group inherits the configuration.

Caution	For the SA protocol operation mode configuration to take effect, you must enable and disable the interface, which disrupts traffic on the interface.

Before you begin

Ensure that you enabled Inspur TrustSec authentication on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port* [**-** *port2*]
3. **cts dot1x**
4. **sap modelist** [**gcm-encrypt** | **gcm-encrypt-256** | **gmac** | **no-encap** | **null**]
5. **exit**
6. **shutdown**

7. **no shutdown**
8. **exit**
9. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port* [**-** *port2*]
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies a single interface or a range of interfaces and enters interface configuration mode.

Step 3	**cts dot1x**
	Example:
	`switch(config-if)# cts dot1x`
	`switch(config-if-cts-dot1x)#`
		Enables 802.1X authentication for Inspur TrustSec and enters Inspur TrustSec 802.1X configuration mode.

Step 4	**sap modelist** [**gcm-encrypt** | **gcm-encrypt-256** | **gmac** | **no-encap** | **null**] **Example:** `switch(config-if-cts-dot1x)# sap modelist gmac`	Configures the SA protocol authentication mode on the interface. Use the **gcm-encrypt** keyword for GCM encryption. This option is the default. Use the **gcm-encrypt-256** keyword for 256-bit GCM encryption. Use the **gmac** keyword for GCM authentication only. Use the **no-encap** keyword for no encapsulation for SA protocol on the interface and no SGT insertion. Use the **null** keyword for encapsulation without authentication or encryption for SA protocol on the interface. Only the SGT is encapsulated.

Step 5	**exit**
	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`
		Exits Inspur TrustSec 802.1X configuration mode.

Step 6	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 7	**no shutdown**	Enables the interface and SA protocol operation mode on the interface.

	Command or Action	Purpose
	Example: `switch(config-if)# no shutdown`	

Step 8	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 9	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch(config)# show cts interface all`	Displays the Inspur TrustSec configuration on the interface.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling Inspur TrustSec Authentication.

Regenerating SA Protocol Keys on an Interface

You can trigger an SA protocol exchange to generate a new set of keys and protect the data traffic flowing on an interface.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **cts rekey ethernet** *slot*/*port*
2. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**cts rekey ethernet** *slot*/*port* **Example:** `switch# cts rekey ethernet 2/3`	Generates the SA protocol keys for an interface.
Step 2	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch# show cts interface all`	Displays the Inspur TrustSec configuration on the interfaces.

Related Topics

Enabling Inspur TrustSec Authentication .

12.8.5 Configuring Inspur TrustSec Authentication in Manual Mode

You can manually configure Inspur TrustSec on an interface if your Inspur INOS device does not have access to a Inspur Secure ACS or authentication is not needed because you have the MAC address authentication bypass feature enabled. You must manually configure the interfaces on both ends of the connection.

Caution	For the Inspur TrustSec manual mode configuration to take effect, you must enable and disable the interface, which disrupts traffic on the interface.

Before you begin

Ensure that you enabled Inspur TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface slot*/*port*
3. **cts manual**
4. **sap pmk** {*key* [**left-zero-padded**] [**display encrypt**] | **encrypted** *encrypted_pmk* | **use-dot1x**} [**modelist** {gcm-encrypt |gcm-encrypt-256 | gmac | no-encap | null}]
5. (Optional) **policy dynamic identity** *peer-name*
6. (Optional) **policy static sgt** *tag* [**trusted**]
7. **exit**
8. **shutdown**
9. **no shutdown**
10. **exit**
11. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}
12. (Optional) **show cts sap pmk** {**all** | **interface ethernet** *slot*/*port*}
13. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *interface slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Specifies an interface and enters interface configuration mode.

Step 3	**cts manual**
	Example:
	`switch(config-if)# cts manual`
	`switch(config-if-cts-manual)#`
		Enters Inspur TrustSec manual configuration mode.
		Note You cannot enable Inspur TrustSec on interfaces in half-duplex mode.

	Command or Action	Purpose
Step 4	**sap pmk** {*key* [**left-zero-padded**] [**display encrypt**] | **encrypted** *encrypted_pmk* | **use-dot1x**} [**modelist** {**gcm-encrypt** |**gcm-encrypt-256** | **gmac** | **no-encap** | **null**}] **Example:** `switch(config-if-cts-manual)# sap pmk fedbaa modelist gmac`	Configures the SA protocol pairwise master key (PMK) and operation mode. SA protocol is disabled by default in Inspur TrustSec manual mode. The *key* argument is a hexadecimal value with an even number of characters and a maximum length of 32 characters. Use the **left-zero-padded** keyword to pad zeros to the left of the entered string if the PMK length is less than 32 bytes. Use the **display encrypt** keyword to specify that the configured PMK be displayed in AES-encrypted format in the running configuration. Use the **encrypted** *encrypted_pmk* keyword to specify an encrypted PMK string of 64 bytes (128 hexadecimal characters). Use the **use-dot1x** keyword when the peer device does not support Inspur TrustSec 802.1X authentication or authorization but does support SA protocol data path encryption and authentication. The mode list configures the cipher mode for the data path encryption and authentication as follows: Use the **gcm-encrypt** keyword for GCM encryption. This option is the default. Use the **gcm-encrypt-256** keyword for GCM encryption. Use the **gmac** keyword for GCM authentication. Use the **no-encap** keyword for no encapsulation and no SGT insertion. Use the **null** keyword for encapsulation of the SGT without authentication or encryption.
Step 5	(Optional) **policy dynamic identity** *peer-name* **Example:** `switch(config-if-cts-manual)# policy dynamic identity MyDevice2`	Configures a dynamic authorization policy download. The *peer-name* argument is the Inspur TrustSec device ID for the peer device. The peer name is case sensitive. **Note** Ensure that you have configured the Inspur TrustSec credentials and AAA for Inspur TrustSec. **Note** The **policy dynamic** and **policy static** commands are mutually exclusive. Only

	Command or Action	Purpose
		one can be applied at a time. To change from one to the other, you must use the **no** form of the command to remove the configuration before configuring the other command.
Step 6	(Optional) **policy static sgt** *tag* [**trusted**] **Example:** `switch(config-if-cts-manual)# policy static sgt 0x2`	Configures a static authorization policy. The *tag* argument is a decimal value or a hexadecimal value in the format **0x***hhhh*. The decimal range is from 2 to 65519, and the hexadecimal range is from 0x2 to 0xffef. The **trusted** keyword indicates that traffic coming on the interface with this SGT should not have its tag overridden. **Note** The **policy dynamic** and **policy static** commands are mutually exclusive. Only one can be applied at a time. To change from one to the other, you must use the **no** form of the command to remove the configuration before configuring the other command.

Step 7	**exit**
	Example:
	`switch(config-if-cts-manual)# exit`
	`switch(config-if)#`
		Exits Inspur TrustSec manual configuration mode.

Step 8	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 9	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and enables Inspur TrustSec authentication on the interface.

Step 10	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 11	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch# show cts interface all`	Displays the Inspur TrustSec configuration for the interfaces.
Step 12	(Optional) **show cts sap pmk** {**all** | **interface ethernet** *slot*/*port*} **Example:** `switch# show cts sap pmk all`	Displays the hexadecimal value of the configured PMK for all interfaces or a specific Ethernet interface.
Step 13	(Optional) **copy running-config startup-config**	Copies the running configuration to the

	Command or Action	Purpose
	Example: `switch# copy running-config startup-config`	startup configuration.

Related Topics

Enabling the Inspur TrustSec SGT Feature .

12.8.6 Configuring Inspur TrustSec Authentication in Dot1x Mode

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface slot*/*port*
3. **cts manual**
4. **sap pmk** {*key* [**left-zero-padded**] [**display encrypt**] | **encrypted** *encrypted_pmk* | **use-dot1x**} [**modelist** {gcm-encrypt | gcm-encrypt-256 | gmac | no-encap | null}]
5. **exit**
6. **shutdown**
7. **no shutdown**
8. **exit**
9. (Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*}
10. (Optional) **show cts sap pmk** {**all** | **interface ethernet** *slot*/*port*}
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *interface slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/29-30`
	`switch(config-if-range)#`
		Specifies an interface and enters interface configuration mode.

Step 3	**cts manual**
	Example:
	`switch(config-if-range)# cts dot1x`
	`switch(config-if-cts-dot1x)#`
		Enters Inspur TrustSec Dot1x configuration mode.

	Command or Action	Purpose
Step 4	**sap pmk** {*key* [**left-zero-padded**] [**display encrypt**] | **encrypted** *encrypted_pmk* | **use-dot1x**} [**modelist** {**gcm-encrypt** | **gcm-encrypt-256** | **gmac** | **no-encap** | **null**}] **Example:** `switch(config-if-cts-dot1x)# sap modelist gcm-encrypt-256`	Configures the SAP pairwise master key (PMK) and operation mode. SAP is disabled by default in Inspur TrustSec manual mode. The *key* argument is a hexadecimal value with an even number of characters and a maximum length of 32 characters. Use the **left-zero-padded** keyword to pad zeros to the left of the entered string if the PMK length is less than 32 bytes. Use the **display encrypt** keyword to specify that the configured PMK be displayed in AES-encrypted format in the running configuration. Use the **encrypted** *encrypted_pmk* keyword to specify an encrypted PMK string of 64 bytes (128 hexadecimal characters). Use the **use-dot1x** keyword when the peer device does not support Inspur TrustSec 802.1X authentication or authorization but does support SAP data path encryption and authentication. The mode list configures the cipher mode for the data path encryption and authentication as follows: Use the **gcm-encrypt** keyword for GCM encryption. This option is the default. Use the **gcm-encrypt-256** keyword for 256-bit GCM encryption. Use the **gmac** keyword for GCM authentication. Use the **no-encap** keyword for no encapsulation and no SGT insertion. Use the **null** keyword for encapsulation of the SGT without authentication or encryption.

Step 5	**exit**
	Example:
	`switch(config-if-cts-dot1x)# exit`
	`switch(config-if)#`
		Exits Inspur TrustSec Dot1x configuration mode.

Step 6	**shutdown** **Example:** `switch(config-if)# shutdown`	Disables the interface.
Step 7	**no shutdown** **Example:** `switch(config-if)# no shutdown`	Enables the interface and enables Inspur TrustSec authentication on the interface.

	Command or Action	Purpose

Step 8	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 9	(Optional) **show cts interface** {**all** | **brief** | **ethernet** *slot*/*port*} **Example:** `switch# show cts interface all`	Displays the Inspur TrustSec configuration for the interfaces.
Step 10	(Optional) **show cts sap pmk** {**all** | **interface ethernet** *slot*/*port*} **Example:** `switch# show cts sap pmk all`	Displays the hexadecimal value of the configured PMK for all interfaces or a specific Ethernet interface.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

12.9 Inspur TrustSec Support on Port-Channel Members

Before Inspur INOS Release 8.2(3), configuration compatibility on port-channel member interfaces with respect to TrustSec configuration was not enforced. Also, Inspur TrustSec configuration was not allowed on port-channel interfaces.

However, from Inspur INOS Release 8.2(3), TrustSec configuration compatibility on port-channel members is enforced and also Trustsec configuration on port-channel interfaces is allowed. The following sections provide more information:

12.9.1 Configuration Models

The following are the configuration models:

- Inspur TrustSec configuration on port-channel interfaces:

Any Inspur TrustSec configuration performed on a port-channel interface is inherited by all its member interfaces.

- Inspur TrustSec configuration on port-channel member interfaces:

Port-channel compatibility parameters are not allowed to be configured on port-channel member interfaces.

Other Inspur TrustSec configurations, such as MACSec configuration, which would not result in incompatibility, are allowed on port-channel member interfaces.

- Adding new members to a port-channel:
- Using the **channel-group** command:

Addition of new members is accepted, if the configuration on the port-channel and that on all members are compatible; if not, the addition is rejected.

- Using the **channel-group force** command:

If the interfaces being added are capable of supporting the port-channel configuration, they inherit the compatibility parameters from the port-channel and the addition is accepted. However, if some interfaces being added are not capable of supporting the port-channel configuration, the addition is rejected.

12.9.2 User Interface Updates for Inspur INOS Release 8.2(3)

The following are the updates to the user interfaces after Inspur INOS Release 8.2(3):

- When the **channel group** or **channel-group force** command is issued, if there is any incompatibility in the Inspur TrustSec configuration, an error message is displayed to the user pointing to the incompatible configuration.
- The **show run** and **show start** command displays the Inspur TrustSec configuration on port-channel interfaces as well along with that on physical ethernet interfaces.
- The **show cts role-based sgt-map** command displays the port-sgt learnt mappings that was learnt on the port-channel interface, if applicable.

12.9.3 In-Service Software Upgrades

When In-Service Software Upgrades (ISSU) is performed from a lower version that does not support this feature, as soon as the ISSU is completed, all port-channels inherit the compatibility parameters from their first configured member interface. A warning level syslog is generated for port-channels on which the configuration incompatibility is detected.

12.10 Verifying the Inspur TrustSec MACSec Configuration

To display Inspur TrustSec MACSec configuration information, perform one of the following tasks:

Command	Purpose
show cts	Displays Inspur TrustSec information.
show cts capability interface {**all** | **ethernet** *slot*/*port*}	Displays the Inspur TrustSec capability of all interfaces or a specific Ethernet interface.
show cts credentials	Displays Inspur TrustSec credentials for EAP-FAST.
show cts environment-data	Displays Inspur TrustSec environmental data.
show cts interface {**all** | **brief** | **ethernet** *slot*/*port*}	Displays the Inspur TrustSec configuration for the interfaces.
show cts pacs	Displays Inspur TrustSec authorization information and PACs in the device key store.
show running-config cts	Displays the Inspur TrustSec information in the running configuration.

12.11 Additional References for Inspur TrustSec MACSec

This sections provides additional information related to implementing Inspur TrustSec.

Related Documentation

Related Topic	Document Title
Inspur INOS licensing	*Inspur INOS Licensing Guide*
Command Reference	*Inspur CN12700 Series INOS Security Command Reference*

CHAPTER 13 Configuring MACsec Key Agreement

This chapter describes how to configure MACsec Key Agreement (MKA), and includes the following sections:

- Finding Feature Information
- Information About MACsec
- Feature History for MKA
- Default Settings for MKA
- Guidelines and Limitations for MKA
- Configuring MKA
- Configuring a Non-standard Ethernet Type Value for EAPOL
- Configuring a Non-standard DMAC Address Value for EAPOL
- Displaying MKA Statistics and Capability
- Additional References for MKA

13.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

13.2 Information About MACsec

This section provides information about MACsec, and contains the following sections:

13.2.1 MACsec

MACsec is an IEEE 802.1AE standards-based Layer 2 hop-by-hop encryption that provides data confidentiality, integrity, and replay protection for media access-independent protocols.

MACsec provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying. The infrastructure required to set up a MACsec service is achieved by the Inspur proprietary protocol, which is the security association (SA) protocol, or the MKA protocol based on the 802.1x-rev2010 standard.

The MKA protocol provides the required session keys and manages the required encryption keys. 802.1AE encryption with MKA is supported on all link types for encryption between any of the following devices, which are capable of MKA:

- Between switches
- Between routers
- Between switches and routers
- Between hosts and access switches

MACsec encrypts all the data, except the source and destination MAC addresses of an Ethernet packet. You can secure data on physical media using MACsec, which prevents data compromise at higher layers. As a result, MACsec encryption takes priority over any other encryption method, such as IPsec and SSL, at higher layers. MACsec provides integrity for the entire frame including the source and destination MAC addresses.

SECurity entitY MIB IEEE8021-SECY-MIB Support

MACsec supports the IEEE8021-SECY-MIB from Inspur INOS Release 8.2(3) onwards.

- The IEEE8021-SECY-MIB provides Simple Network Management Protocol (SNMP) access to the MAC security entity (SecY) MIB running with MACsec-enabled line cards. The IEEE8021-SECY-MIB is used to query on the SecY data, encryption, decryption, and the hardware statistics.
- The IEEE8021-SECY-MIB contains tables that specifies the detailed attributes of the MACsec Controlled Port interface index.

MKA Unique PSK Support

With this MACsec enhancement in Inspur INOS Release 8.2(3), pre-shared keys (PSK) are supported on break out interfaces..

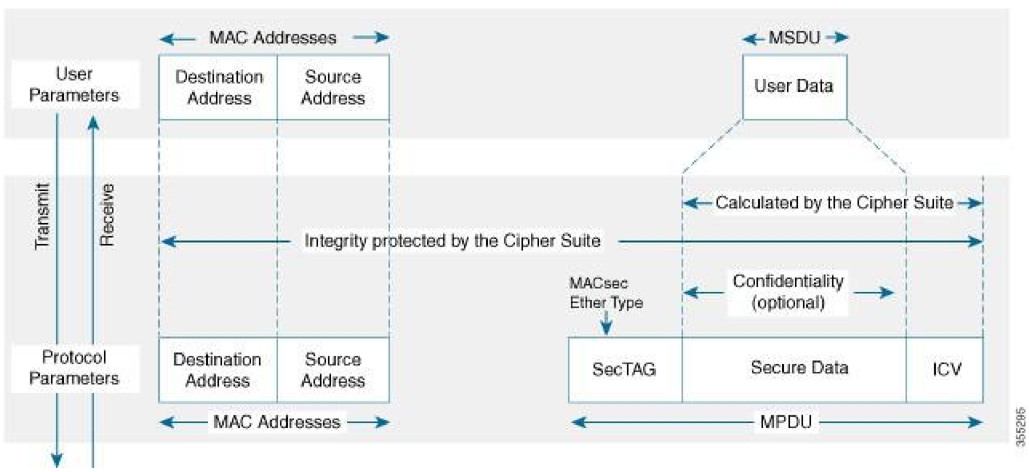
MKA Unrecoverable SAK Support

This MACsec enhancement in Inspur INOS Release 8.2(3) makes the Secure Association Key (SAK) unrecoverable. A SAK rekey occurs every time a session comes up (such as power cycle, reload, failover, and so on).

MACsec Frame Format

The following figure shows the MACsec frame:

Figure 22： MACsec Frame

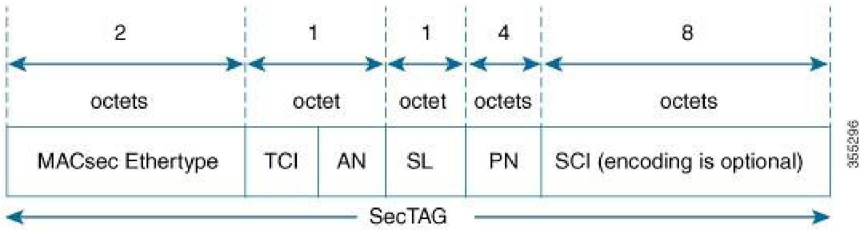


The MAC Protocol Data Unit (MPDU) in the MACsec frame has the following components:

- Security tag (SecTag)—The SecTag is 8 to 16 bytes in length and identifies the secure association key (SAK) to be used for the frame. With Secure Channel Identifier (SCI) encoding, the security tag is 16 bytes in length, and without the encoding, 8 bytes in length (SCI encoding is optional). The SecTag also provides replay protection when frames are not received in a sequence.

The following figure shows the components of the SecTag:

Figure 23： SecTag



- Secure data—The data, which is encrypted using MACsec, in the frame. It can be two or more octets in length.
- Integrity check value (ICV)—The ICV provides an integrity check for the frame. It is 8 to 16 bytes in length depending on the cipher suite. Frames that do not match the expected ICV are dropped at the remote end's ingress port.

MKA Protocol

From Inspur INOS Release 8.2(1), MKA is supported only on Inspur CN M-3 series modules. The MKA protocol performs the following tasks:

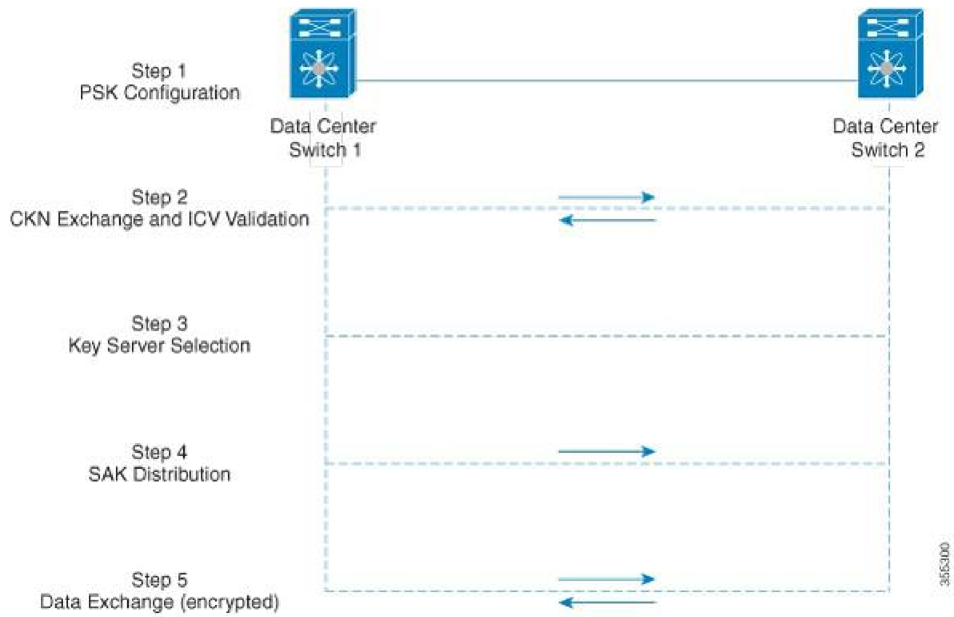
- Authenticating the members
- Establishing and managing connectivity association
- Managing the live or potential peers that are a part of a connectivity association, using keepalive every two seconds
- Negotiating the cipher suite
- Electing the key server from among the members of connectivity association
- Generating Secure Association Key (SAK) and managing the key server
- Distributing SAKs in an encrypted format by the key server to its members
- Installing a key on the SecY of each member
- Refreshing SAK before the old SAK expires

The packet body in an Extensible Authentication Protocol (EAP) over LAN (EAPOL) protocol data unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). When no MKPDU is received from a participant after three heartbeats, the corresponding participant is deleted from the live peer list. Each heartbeat is 2-seconds long. For example, when one of the remote participant switches gets disconnected, the corresponding local participant switch considers the remote participant switch as lost after three heartbeats.

MACsec provides encryption using Advanced Encryption Standard (AES) algorithm in Layer 2. MACsec uses the MKA protocol to exchange session keys and manage encryption keys.

The following figure shows the MKA encryption process resulting in a secured data link:

Figure 24： MKA Encryption Process



The following is a description of the MKA encryption process:

1. When a link is established between two switches, they become peers. Mutual peer authentication takes place by configuring a pre-shared key (PSK). In a switch-to-switch connection using PSK, there is no concept of authenticator because of the EAP authentication on the switch. PSK can be configured only manually. From Inspur INOS Release 8.2(3) pre-shared keys (PSK) are supported on break out interfaces.

2. After successful peer authentication, a connectivity association is formed between the peers, and a secure connectivity association key name (CKN) is exchanged. After the exchange, the MKA ICV is validated with a connectivity association key (CAK), which is effectively a secret key.

3. A key server is selected from among the switches, based on the configured key server priority. The lower the priority value, the higher the chances of a switch becoming the key server. If no value is configured, the default value 16 is taken to be the key server priority value for a switch. The lowest priority value leads to a switch being configured as the key server, while the other switch functions as a key client. The following rules apply to the key server selection process:

- Numerically lower values of key server priority and SCI are accorded the highest preference.
- Each switch selects a peer that advertises the highest preference as its key server, provided that peer has not selected another switch as its key server, or is not willing to function as the key server.
- In the event of a tie for highest preferred key server, the switch with the highest SCI priority is chosen as the key server.

4. A security association is formed between the peers. The key server generates and distributes the SAK to the key client, or the peer. SAKs are generated for every data exchange between the peers.

5. Encrypted data is exchanged between the peers.

13.2.2 Behavior of MKA Protocol

A switch handles MACsec and non-MACsec frames based on the security policy configured locally. The security policy can be **should-secure** or **must-secure**. The **should-secure** policy allows any unencrypted frame until its link is secured. After the link is secured, this policy allows only encrypted frames. The **must-secure** policy does not allow any unencrypted frame except EAPOL until its link is secured. After the link is secured, this policy allows only encrypted frames.

MACsec frames are encrypted and protected with an ICV using the security credentials provided by MKA. When a switch receives encrypted frames from the peers, it decrypts them and calculates the correct ICV by using the session keys provided by MKA. Any unencrypted frame received on a secured port is dropped. The switch compares the resulting ICV to the ICV within the frame. If they are not identical, the frame is dropped.

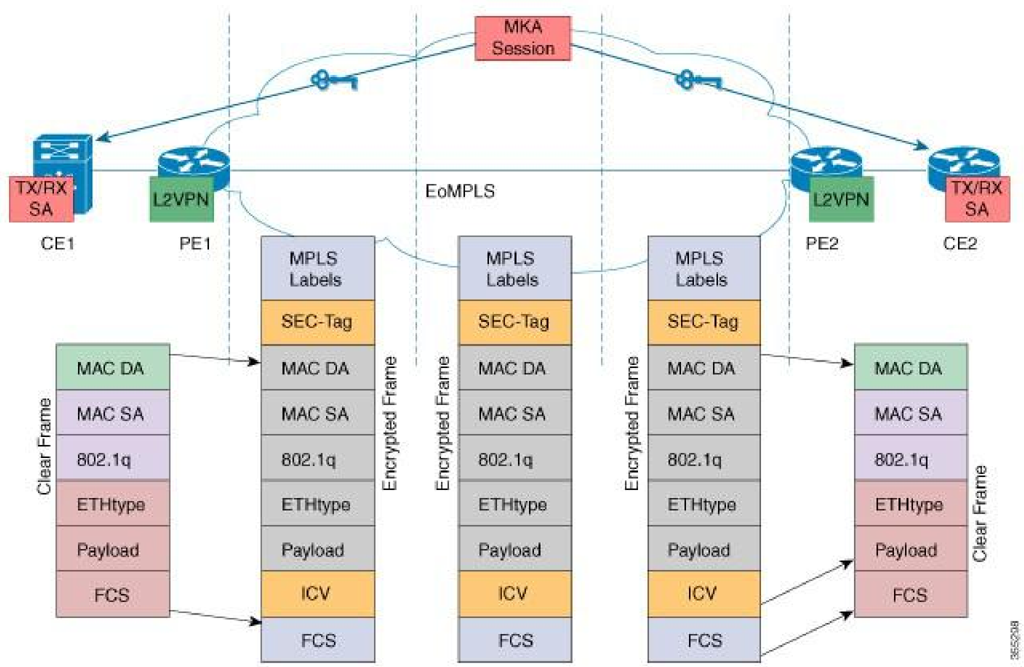
13.2.3 Use Cases for MKA

The following is a list of MKA use cases:

- MACsec on port channels
- Securing Provider Edge-to-Customer Edge links in a Multiprotocol Label Switching (MPLS) network
- Securing PE-to-PE links using dark fiber
- Securing CE-to-CE links using the EoMPLS network

The following figure shows how MKA is used to secure CE-to-CE devices using the EoMPLS network:

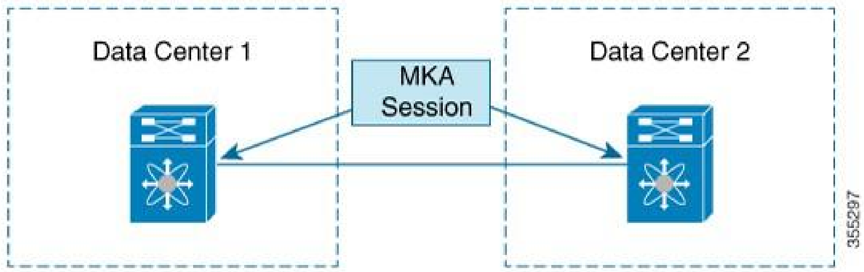
Figure 25： Securing CE-to-CE



- Securing Data Center Interconnect (DCI)

The following figure shows how MKA is used in securing DCI:

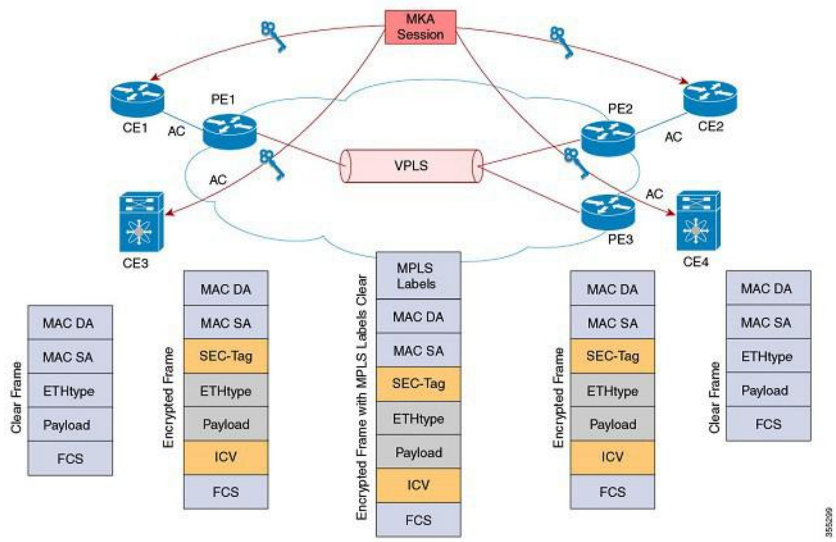
Figure 26：Securing DCI



- Securing a CE to multiple CEs using the Virtual Private LAN Services (VPLS) network

The following figure shows how MKA is used to secure a CE to multiple CEs using the VPLS network:

Figure 27：Securing CE to multiple CEs



13.2.4 Non-Standard Ethernet Type and DMAC Support for MACsec

From Inspur INOS Release 8.2(3), Inspur enables networks with WAN MACsec to change the Extensible Authentication Protocol (EAP) over LAN (EAPOL) protocol destination address, and the Ethernet type values to nonstandard values. The EAPOL destination Ethernet type can be changed from the default Ethernet type of 0x888E to an alternate value or, the EAPOL destination MAC address can be changed from the default DMAC of 01:80:C2:00:00:03 to an alternate value, to avoid being consumed by a provider bridge.

Prior to Inspur INOS 8.3(1), to establish a MACsec session in a WAN environment, MACsec and MKA implementation would negotiate MKA keys using an EAPOL packet. These EAPOL packets used Metro Ethernet Forum-defined (MEF defined) DMAC address (01:80:c2:00:00:03) and Ethernet type (0x888E). These well-defined MAC addresses used to be consumed by the provider switches.

The following table shows the combinations that are supported on the Inspur CN12700 Series Switches for EAPOL packets with DMAC and Ethernet type:

Table 27： Supported EAPOL Packets with Ethernet Type and DMAC

Ethernet Type Value	DMAC Value	Supported Combination
Standard (0x888E)	Standard (01:80:c2:00:00:03)	Yes
Non-standard	Standard	Yes
Standard	Non-standard	Yes
Non-standard	Non-standard	Yes

13.3 Feature History for MKA

The following table lists the release history for this feature.

Table 28： Feature History for MKA

Feature Name	Releases	Feature Information
MACSec should-secure	8.2(3)	Added support for the should-secure policy.
Non-Standard Ethernet Type or DMAC Support for MACSec	8.3(1)	Added support for this feature.
MKA	8.2(1)	The MKA feature was included in the Inspur CN F3 Series modules.

13.4 Default Settings for MKA

This table lists the default settings for MKA.

Table 29： Default MKA Settings

Parameter	Default
MKA	Disabled
MACsec policy	**system-default-macsec-policy**
Key server priority value for MACsec encryption	16
Cipher suite	**GCM-AES-XPN-25**
Confidentiality offset	0

13.5 Guidelines and Limitations for MKA

MKA has the following guidelines and limitations:

- MKA can be enabled or disabled independent of the Inspur TrustSec feature.
- MKA and Inspur TrustSec SGT cannot be used together at the same time on a given physical port.
- MKA interacts with a Inspur TrustSec process to obtain the Inspur TrustSec SA protocol and SGT details.
- MKA and Inspur TrustSec processes can be used at the same time on a system.
- MKA and Inspur TrustSec MACsec, that is, the SA protocol, cannot be used together at the same time on a given physical port.
- MKA is currently supported only on physical ports and port channels. It is not supported on subinterfaces.
- MKA cannot be configured on member ports.

- Interfaces configured with MKA cannot be introduced into a port channel.
- MKA does not support stateful restart, stateful system switchover, or In-Service Software Upgrades (ISSU).
- Inspur CN12700 Series switches do not support the should-secure mode for the MKA security policy. The default mode is must-secure. From Inspur INOS Release 8.2(3) the should-secure security policy is supported.
- From Inspur INOS Release 8.2(3) syslog messages are displayed when the MACSEC session goes up or down.
- The MKA SecY statistics can only be obtained from the line card module, and not from the supervisor.

The Non-Standard Ethernet Type and DMAC Support for MACSec feature has the following guidelines and restrictions:

- You can configure only one DMAC, or Ethernet type, or DMAC and Ethernet type combination value for MACSec. For example, if you have configured the nonstandard Ethernet Type value **macsec non-standard eapol ethertype 0x8976**, you cannot configure another nonstandard Ethernet Type called **macsec non-standard eapol ethertype 0x8972**. The same principle holds good for nonstandard DMAC option too.
- You cannot modify or delete a nonstandard Ethernet type or DMAC if any interface with a policy is using the nonstandard values. To modify the globally configured Ethernet type or DMAC, you have to disassociate the policies from all the interfaces.
- When a Provider Edge device is a Inspur CN12700 Series Switch, you need to configure Virtual Private LAN Services (VPLS) with VLAN 1 or native VLAN to bring up the nonstandard MKA sessions.

13.6 Configuring MKA

Before configuring MKA on an interface, the MACsec keychain and the MACsec policy must be defined. If a keychain does not exist before configuring the interface, an empty keychain will be created. If a policy does not exist before configuring the interface, the default policy is used. The default policy is **system-default-macsec-policy**. Configuring MKA involves the following steps:

1. Enable the MKA feature.
2. (Optional) Create a MACsec keychain.
3. (Optional) Create a MACsec policy.
4. Apply a MACsec on a physical port.

13.6.1 Enabling MKA

Step 1 Enter the global configuration mode:
switch# **configure terminal**

Step 2 Enable the MKA feature:

switch(config)# **feature mka**
Note Use the **no** form of this command to disable the MKA feature.

Step 3 Exit the global configuration mode: switch(config)# **exit**

Example: Enabling MKA

This running configuration example shows how to enable the MKA feature:

```
configure terminal
 feature mka
 exit
```

13.6.2 Configuring a MACsec Keychain

Step 1 Enter the global configuration mode:
switch# **configure terminal**

Step 2 Configure a keychain, and enter the macseckeychain configuration mode: switch(config)# **key chain** *keychain-name* **macsec**
Note Use the no form of this command to remove the keychain.

Step 3 Configure a MACsec key and enter the macseckeychain-macseckey configuration mode: switch(config-macseckeychain)# **key** *key-ID*
Note Valid MACsec key identifier range is from 1 to 32 octet. The maximum size of the octet string is 64 characters. Use the **no** form of this command to remove the key.

Step 4 Set the key octet string and the 128-bit AES encryption algorithm:

switch(config-macseckeychain-macseckey)# **key-octet-string** *string* **cryptographic-algorithm AES-128-CMAC**
Note The maximum size of the octet string is 64 characters. Use the no form of this command to remove the string.

Step 5 Exit all the configuration modes:

switch(config-macseckeychain-macseckey)# **end**

Step 6 (Optional) Verify the MACsec keychain:
switch# **show key chain** *keychain-name*

Example: Configuring a MACsec Keychain

This running configuration example shows how to configure a MACsec keychain. Replace the <*placeholders*> with relevant values for your setup.

```
configure  terminal
 key chain <k1> macsec
 key <01>
 key-octet-string <0123456789aabbcc0123456789aabbcc> cryptographic-algorithm AES_128_CMAC
 end
```

This example shows how to verify a MACsec keychain:

```
switch# show key chain
 Key-Chain k1 Macsec
 Key 01 -- text 7 "075f701e1d5d4c53404a520d052829272b63647040534355560e005952560c001b"
    cryptographic-algorithm AES_128_CMAC
    send lifetime (always valid) [active]
```

13.6.3 Configuring a MACsec Policy

Step 1 Enter the global configuration mode:
switch# **configure terminal**

Step 2 Enter the MACsec policy configuration mode:
switch(config)# **macsec policy** *policy-name*
Note Use the **no** form of this command to disable the policy.

Step 3 Configure a security policy to define the handling of data and control packets: switch(config-macsec-policy)# **security-policy** {**must-secure** | **should-secure** }

Note

- should-secure: This policy allows any unencrypted frame until its link is secured. After the link is secured, this policy allows only encrypted frames.
- must-secure: This policy does not allow any unencrypted frame until its link is secured. After the link is secured, this policy allows only encrypted frames.

Step 4 Configure the confidentiality offset:

switch(config-macsec-policy)# **conf-offset** {**CONF-OFFSET-0** | **CONF-OFFSET-30** | **CONF-OFFSET-50**}

Note Use the no form of this command to disable the confidentiality offset. If the confidentiality offset is unspecified, the encryption is not offset.

Step 5 Configure the cipher suite:

switch(config-macsec-policy)# **cipher-suite** {**GCM-AES-128** | **GCM-AES-256** | **GCM-AES-XPN-128** | GCM-AES-XPN-256}

Note Use the no form of this command to set the default value. If the cipher suite is unspecified, the default is G**CM-AES-XPN-256.**

Step 6 Set the key server priority value:

switch(config-macsec-policy)# **key-server-priority** *value*

Note The valid range is from 0 to 255. The default is 16. Use the no form of this command to set the default value.

Step 7 Set the SAK expiry time:

switch(config-macsec-policy)# **sak-expiry-time** *seconds*

Note The range is from 1 to 2592000 seconds. The default is pn-exhaust. Use the no form of this command to set the default value.

Step 8 Exit all the configuration modes:
switch(config-macsec-policy)# **end**

Step 9 (Optional) Verify MKA:
switch# **show run mka**

Step 10 (Optional) Verify the MACsec policy:
switch# **show macsec policy** [*policy-name*]

Example: Configuring a MACsec Policy

This running configuration example shows how to configure a MACsec policy. Replace the <*placeholders*> with relevant values for your setup.

```
configure terminal
 macsec policy <p1>
  security-policy <must-secure | should-secure>
  conf-offset CONF-OFFSET-0

  cipher-suite GCM-AES-XPN-256
  key-server-priority <9>
  sak-expiry-time <60>
  end
```

This example shows how to configure a should-secure security policy.

```
configure terminal
 macsec policy p100
  security-policy should-secure
  end
```

This example shows how to verify a configured security policy:

```
switch# show macsec policy p100
MACSec Policy      Cipher           Pri  Window   Offset   Security        SAK Rekey time
------------------------------ ---------------- ---- -------- -------- ------------ ------
p100               GCM-AES-XPN-256  16   0        0        should-security pn-exhaust
```

This example displays the status of MKA:

```
switch# show run mka
 !Command: show running-config mka
!Time: Wed Apr 19 05:08:01 2017
version 8.2(0)SK(1)
feature mka
macsec policy p1
   cipher-suite GCM-AES-XPN-128
   key-server-priority 9
   security-policy must-secure
   sak-expiry-time 60
```

This example shows how to verify a configured MACsec policy:

```
switch# show macsec policy p1
MACsec Policy            Cipher           Pri  Window   Offset   Security     SAK Rekey
time
------------------------ ---------------- ---- -------- -------- ------------ --------------
p1                       GCM-AES-XPN-128    9     0        0     must-secure       60
```

This example shows how to view all the MACsec policies in a switch:

```
switch# show macsec policy
MACsec Policy                   Cipher          Pri Window Offset   Security     SAK Rekey
 time
------------------------------- --------------- --- ------ -------- ------------ -----------
p1                               GCM-AES-XPN-128  9     0    0        must-secure  60

system-default-macsec-policy    GCM-AES-XPN-256  16    0    0       must-secure  pn-exhaust
```

13.6.4 Configuring MKA on an Interface or a Port Channel

Step 1 Enter the global configuration mode:
switch# **configure terminal**

Step 2 Configure an interface or a port channel:
switch(config)# **interface ethernet** *slot/port*
switch(config)# **interface port-channel** *port-channel*

Step 3 Configure a policy and the policy name for the MACsec keychain:
switch(config-if)# **macsec keychain** *keychain-name* **policy** *policy-name*
Note Use the **no** form of this command to disable the policy on the interface or the port channel.

Step 4 Exit all the configuration modes:

switch(config-if)# **end**

Step 5 (Optional) Verify the MKA session details:

switch# **show macsec mka session** [**interface ethernet** *slot*/*port*] [**details**] [**internal-details**]

Step 6 (Optional) View the MKA summary information:
switch# **show macsec mka summary**

Example: Configuring MKA on an Interface or Port Channel

This running configuration example shows how to configure MKA on an interface. Replace the <*placeholders*> with relevant values for your setup.

```
configure terminal
 interface ethernet <11>/<31>
 macsec keychain <k3> policy <p1>
 end
```

This running configuration example shows how to configure MKA on a port channel. Replace the <*placeholders*> with relevant values for your setup.

```
configure terminal
 interface port channel <100>
 macsec keychain <k3> policy <p1>
 end
```

The following example shows information about all the interfaces in the MKA session:

```
switch# show macsec mka session
Interface         Local-TxSCI               # Peers            Status       Key-Server
----------------- ------------------------- ------------------ ------------ ----------
Ethernet2/1       0000.0043.0038/0001       1                  Secured      Yes
Ethernet2/7       0000.0043.003e/0001       1                  Secured      Yes
Ethernet2/25      0000.0043.0050/0001       1                  Secured      No
Ethernet2/30      0000.0043.0055/0001       1                  Secured      No
----------------- ------------------------- ------------------ ------------ ----------
Total Number of Sessions : 4
        Secured Sessions : 4
        Pending Sessions : 0
```

The following example shows detailed information about all the interfaces in the MKA session:

```
switch# show macsec mka session details
Detailed Status for MKA Session
-----------------------------------
Interface Name          :       Ethernet11/25
Session Status                    : Secured
Local Tx-SCI                      : 00b0.e135.9c24/0001
Local Tx-SSCI                     : 3
MKA Port Identifier               : 3
CAK Name (CKN)                    :
0100000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI)            : 17173194E288E086B275A49F
Message Number (MN)               : 12465
MKA Policy Name                   : p1
Key Server Priority               : 9
Key Server                        : No
SAK Cipher Suite                  : GCM-AES-XPN-128
SAK  Cipher  Suite  (Operational)  :  GCM-AES-XPN-128
Replay Window Size                : 0
```

```
Confidentiality Offset              : CONF-OFFSET-0
Confidentiality Offset (Operational): CONF-OFFSET-0
Latest SAK Status                   : Rx & TX
Latest SAK AN                       : 0
Latest SAK KI                       : 10314879
Latest SAK KN                       : 57
Last SAK key time                   : 06:59:24 UTC Wed Apr 19 2017
Number of Macsec Capable Live Peers: 3
Number of SA consumed in Hardware  : 3
Number of Macsec Capable Live Peers Responded: 0
Live Peer List:
MI                          MN         SCI                      SSCI Key-Server Priority
-------------------------------------------------------------------------------------
7F649D00075CA2B14065F50D  12466      00b0.e135.9c23/0001       4     9
67DF7F5DE06AFC9A2F125914  12464      9c57.adfd.8acb/0001       2     9
57BCB803EB00453525F7382C  12466      9c57.adfd.8acc/0001       1     9

Detailed Status for MKA Session
-----------------------------------
Interface Name          : Ethernet4/27
    Session Status                      : Secured
    Local Tx-SCI                        : 5006.ab91.9f4e/0001
    Local Tx-SSCI                       : 2
    MKA Port Identifier                 : 2
    CAK Name (CKN)                      :
1000000000000000000000000000000000000000000000000000000000000000
    Member Identifier (MI)              : 4B18586C685B28F2354B1E2B
    Message Number (MN)                 : 49
    MKA Policy Name                     : mustsecureks
    Key Server Priority                 : 9
    Key Server                          : Yes
    SAK Cipher Suite                    : GCM-AES-256
    SAK  Cipher  Suite  (Operational)  :  GCM-AES-256
    Replay Window Size                  : 0
    Confidentiality Offset              : CONF-OFFSET-0
    Confidentiality Offset (Operational): CONF-OFFSET-0
    Latest SAK Status                   : Rx & TX
    Latest SAK AN                       : 2
    Latest SAK KI                       : 1817712715
    Latest SAK KN                       : 1
    Last SAK key time                   : 20:42:51 UTC Thu May 04 2017
    Number of Macsec Capable Live Peers: 2
    Number of SA consumed in Hardware  : 2
    Number of Macsec Capable Live Peers Responded: 2
Live Peer List:
MI                         MN   SCI              SSCI Key-Server-Priority Tx/Rx Programmed

------------------------------------------------------------------------------- ------
3634B7ADE028833E219C2304 7624 9c57.adfc.0f34/0001 1     16                    Yes
92D6F93C2BC4058AD25FA0E5 7655 5006.ab91.4584/0001 3     16                    Yes
```

The following example shows information about a configured port channel in an MKA session:

```
switch# show macsec mka session interface port-channel 100
Interface          Local-TxSCI                        # Peers         Status   Key-Server
----------------- -------------------------------- --------------- -------- ----------
Ethernet2/7        0000.0043.003e/0001               1               Secured     Yes
Ethernet2/30       0000.0043.0055/0001               1               Secured     No
```

The following example shows detailed information about a configured port channel in the MKA session:

```
switch# show macsec mka session interface port-channel 100 details
Detailed Status for MKA Session
-----------------------------------
```

```
Interface Name          : Ethernet2/7
Session Status          : Secured
Local Tx-SCI            : 0000.0043.003e/0001
Local Tx-SSCI           : 2
MKA Port Identifier     : 2
CAK Name (CKN)          : 0300000000000000000000000000000000000000000000000000000000000000
Member Identifier (MI) : 057D3366D35DA9A19D259D7F
Message Number (MN)     : 1534
MKA Policy Name         :  p1
Key Server Priority     :  16
Key Server              : Yes
SAK Cipher Suite        : GCM-AES-XPN-256
SAK Cipher Suite (Operational)      : GCM-AES-XPN-256
Replay Window Size                  : 0
Confidentiality Offset              : CONF-OFFSET-0
Confidentiality Offset (Operational): CONF-OFFSET-0
Latest SAK Status                   : Rx & TX
Latest SAK AN                       : 0
Latest SAK KI                       : 1714650373
Latest SAK KN                       : 49
Last SAK key time                   : 08:07:29 UTC Fri Jan 06 2017
Number of Macsec Capable Live Peers: 1
Number of SA consumed in Hardware  : 1
Number of Macsec Capable Live Peers Responded: 1

Live Peer List:
MI                        MN   SCI                SSCI Key-Server-Priority Tx/Rx Programmed
------------------------------------------------------------------------------------------
E7A5637789614DB8550C8967 1533 0000.0043.0055/0001  1     16                 Yes
Interface Name          : Ethernet2/30
Session Status          : Secured
Local Tx-SCI            : 0000.0043.0055/0001
Local Tx-SSCI           : 1
MKA Port Identifier     : 1
CAK Name (CKN)          : 0300000000000000000000000000000000000000000000000000000000000000

Member Identifier (MI) : E7A5637789614DB8550C8967
Message Number (MN)     : 1534
MKA Policy Name         : p1
Key Server Priority     : 16
Key Server              : No
SAK Cipher Suite        : GCM-AES-XPN-256
SAK Cipher Suite (Operational)      : GCM-AES-XPN-256
Replay Window Size                  : 0
Confidentiality Offset              : CONF-OFFSET-0

Confidentiality Offset (Operational): CONF-OFFSET-0
Latest SAK Status                   : Rx & TX
Latest SAK AN                       : 0
Latest SAK KI                       : 1714650373
Latest SAK KN                       : 49
Last SAK key time                   : 08:07:29 UTC Fri Jan 06 2017
Number of Macsec Capable Live Peers: 1
Number of SA consumed in Hardware  : 1
Number of Macsec Capable Live Peers Responded: 0
Live Peer List:
MI                       MN    SCI                SSCI Key-Server-Priority Tx/Rx Programmed
-----------------------------------------------------------------------------------------
057D3366D35DA9A19D259D7F 1533  0000.0043.003e/0001 2    16                 Yes
```

The following example shows the summary of the MKA session for the configured interface:

```
switch# show macsec mka summary
Interface      Status   Cipher           Key-Server   MACSEC-policy     CKN
```

```
                                                Keychain
------------- --------- ---------------- ------------ -----------------
----------------------------------------------------------------- --------
Ethernet11/25  Secured  GCM-AES-XPN-128  No           p1
0100000000000000000000000000000000000000000000000000000000000000 k1
Ethernet11/31  Secured  GCM-AES-XPN-128  Yes          p1
0300000000000000000000000000000000000000000000000000000000000000 k3
```

13.7 Configuring a Non-standard Ethernet Type Value for EAPOL

Before you begin

Enable the MKA feature.

SUMMARY STEPS

1. **configure terminal**
2. **[no] macsec non-standard eapol ethertype** *ethernet-type*
3. **macsec policy** *policy-name*
4. **mka enable non-std-eapol {DMAC-ONLY | ETYPE-AND-DMAC-BOTH | ETYPE-ONLY}**
5. **exit**
6. **interface** *interface-name*
7. **macsec keychain** *keychain-name* **policy** *policy-name*
8. (Optional) **show macsec policy** *policy-name*
9. (Optional) **show macsec mka session**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] macsec non-standard eapol ethertype** *ethernet-type* **Example:** `switch(config)# macsec non-standard eapol ethertype 0x8976`	Configures a non-standard Ethernet type for a EAPOL. Use the **no** form of the command to disassociate the Ethernet type for a EAPOL.
Step 3	**macsec policy** *policy-name* **Example:** `switch(config)# macsec policy test`	Configures a MACSec policy and enters MACSec configuration mode.
Step 4	**mka enable non-std-eapol {DMAC-ONLY | ETYPE-AND-DMAC-BOTH | ETYPE-ONLY}** **Example:** `switch(config-macsec-policy)# mka enable non-std-eapol ETYPE-ONLY`	Configures the non-standard EAPOL type for the MACSec policy. You can choose either a non-standard DMAC, or Ethernet type, or both.
Step 5	**exit** **Example:** `switch(config-macsec-policy)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 6	**interface** *interface-name* **Example:** `switch(config-if)# interface ethernet2/1`	Enters interface configuration mode.
Step 7	**macsec keychain** *keychain-name* **policy** *policy-name* **Example:** `switch(config-if)# macsec keychain 1 policy etype-only`	Configure a policy and the policy name for the MACSec keychain and applies it to the interface.
Step 8	(Optional) **show macsec policy** *policy-name* **Example:** `switch(config)# show macsec policy test`	Displays the MACSec policies on the interface.
Step 9	(Optional) **show macsec mka session** **Example:** `switch(config)# show macsec mka session`	Displays the MKA session details.

Configuring a Non-standard Ethernet Type Value for EAPOL

The following running configuration example shows how to configure a non-standard Ethernet Type value for an EAPOL on an interface. Replace the <placeholders> with relevant values for your setup.

```
switch# configure terminal
switch(config)# macsec non-standard eapol ethertype 0x8976
switch(config)# macsec policy test
switch(config-macsec-policy)# mka enable non-std-eapol ETYPE-ONLY
switch(config-macsec-policy)# exit
switch(config-if)interface ethernet2/1
switch(config-if)# macsec keychain 1 policy ETYPE-ONLY
switch(config)# exit

switch(config)# show macsec mka session

Interface       Local-TxSCI              # Peers    Status     Key-Server EAPoL Type

--------------- ------------------------ ---------- ---------- ---------- ------------------
Ethernet2/1     0000.0043.0038/0001      0          Pending    Yes        Non Standard
ETYPE-ONLY
Ethernet2/25    0000.0043.0050/0001      0          Pending    Yes        Non Standard
ETYPE-ONLY
--------------- ------------------------ ---------- ---------- ---------- ------------------
```

13.8 Configuring a Non-standard DMAC Address Value for EAPOL

Before you begin

Enable the MKA feature.

SUMMARY STEPS

1. **configure terminal**
2. **[no] macsec non-standard eapol dmac-addr** *dmac-address*
3. **macsec policy** *policy-name*
4. **mka enable non-std-eapol {DMAC-ONLY | ETYPE-AND-DMAC-BOTH | ETYPE-ONLY}**

5. **exit**
6. **interface** *interface-name*
7. **macsec keychain** *keychain-name* **policy** *policy-name*
8. (Optional) **show macsec policy** *policy-name*
9. (Optional) **show macsec mka session**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	**[no] macsec non-standard eapol dmac-addr** *dmac-address* **Example:** `switch(config)# macsec non-standard eapol dmac-addr 11:11:22:22:33:33`	Configures a non-standard DMAC address value for a EAPOL. Use the **no** form of the command to disassociate the Ethernet type for a EAPOL.
Step 3	**macsec policy** *policy-name* **Example:** `switch(config)# macsec policy test`	Configures a MACSec policy and enters MACSec configuration mode.
Step 4	**mka enable non-std-eapol** {**DMAC-ONLY** | **ETYPE-AND-DMAC-BOTH** | **ETYPE-ONLY**} **Example:** `switch(config-macsec-policy)# mka enable non-std-eapol DMAC-ONLY`	Configures the non-standard EAPOL type for the MACSec policy. You can choose either a non-standard DMAC, or Ethernet type, or both.
Step 5	**exit** **Example:** `switch(config-macsec-policy)# exit`	Exits global configuration mode.
Step 6	**interface** *interface-name* **Example:** `switch(config-if)# interface ethernet2/1`	Enters interface configuration mode.
Step 7	**macsec keychain** *keychain-name* **policy** *policy-name* **Example:** `switch(config-if)# macsec keychain 1 policy DMAC-ONLY`	Configure a policy and the policy name for the MACSec keychain and applies it to the interface.
Step 8	(Optional) **show macsec policy** *policy-name* **Example:** `switch(config)# show macsec policy test`	Displays the MACSec policies on the interface.
Step 9	(Optional) **show macsec mka session** **Example:** `switch(config)# show macsec mka session`	Displays the MKA session details.

Configuring a Non-standard DMAC Address Value for EAPOL

The following running configuration example shows how to configure a non-standard DMAC value for an EAPOL on an interface. Replace the <placeholders> with relevant values for your setup.

```
switch# configure terminal
switch(config)# macsec non-standard eapol dmac-addr 11:11:22:22:33:33
switch(config)# macsec policy test
switch(config-macsec-policy)# mka enable non-std-eapol DMAC-ONLY
switch(config-macsec-policy)# exit
switch(config-if)interface ethernet2/1
switch(config-if)# macsec keychain 1 policy DMAC-ONLY
switch(config)# exit
switch(config)# show macsec mka session

Interface       Local-TxSCI              # Peers    Status     Key-Server EAPoL Type

--------------- ------------------------ ---------- ---------- ---------- ------------------
Ethernet2/1     0000.0043.0038/0001      0          Pending    Yes        Non Standard
DMAC-ONLY
Ethernet2/25    0000.0043.0050/0001      0          Pending    Yes        Non Standard
DMAC-ONLY


--------------- ------------------------ ---------- ---------- ---------- ------------------
```

13.9 Displaying MKA Statistics and Capability

Use the following commands to display MKA statistics and capability:

- **show macsec mka statistics** [**interface ethernet** *slot*/*port*]—Displays MKA statistics for the MKA session on an interface or a port channel.**show macsec mka capability interface all**—Displays MKA capability information for a configured interface.

Example: Displaying MKA Statistics

The following example shows how to obtain the MKA statistics for the MKA session on a configured interface:

```
switch# show macsec mka statistics interface ethernet 11/25

Per-CA MKA Statistics for Session on interface (Ethernet11/25) with CKN 0x1
=============================================================================
CA Statistics
   Pairwise CAK Rekeys............0

SA Statistics
   SAKs Generated....................... 0
   SAKs Rekeyed............................ 0
   SAKs Received..........................60
   SAK Responses Received.. 0

MKPDU Statistics
   MKPDUs Transmitted...... 18676
      "Distributed SAK".. 0

   MKPDUs Validated & Rx... 55986
      "Distributed SAK".. 60
MKA Statistics for Session on interface (Ethernet11/25)
========================================================
CA Statistics
   Pairwise CAK Rekeys............0

SA Statistics
```

```
   SAKs Generated.......................... 0
   SAKs Rekeyed............................ 0
   SAKs Received........................... 60
   SAK Responses Received.. 0

MKPDU Statistics
   MKPDUs Transmitted...... 18676
      "Distributed SAK".. 0
   MKPDUs Validated & Rx... 55986


      "Distributed SAK".. 60
MKA IDB Statistics
   MKPDUs Tx Success.......... 19147
   MKPDUs Tx Fail...............................0
   MKPDUS Tx Pkt build fail... 0
   MKPDUS No Tx on intf down.. 0
   MKPDUS No Rx on intf down.. 0
   MKPDUs Rx CA Not found.
   ...................................................................
   0
   MKPDUs Rx Error............................ 0
   MKPDUs Rx Success.......... 55986

MKPDU Failures
   MKPDU Rx Validation.......................................................0
   MKPDU Rx Bad Peer MN...................................................0
   MKPDU Rx Non-recent Peerlist MN.........................0
   MKPDU Rx Drop SAKUSE, KN mismatch...................0
   MKPDU Rx Drop SAKUSE, Rx Not Set...................... 0
   MKPDU Rx Drop SAKUSE, Key MI mismatch.......... 0
   MKPDU Rx Drop SAKUSE, AN Not in Use..............0
   MKPDU Rx Drop SAKUSE, KS Rx/Tx Not Set... 16956
   MKPDU Rx Drop Packet, Ethertype Mismatch. 0

SAK Failures
   SAK Generation............................................. 0
   Hash Key Generation.................................. 0
   SAK Encryption/Wrap.................................. 0
   SAK Decryption/Unwrap............................ 0

CA Failures
   ICK Derivation............................................. 0
   KEK Derivation............................................. 0
   Invalid Peer MACsec Capability... 0

MACsec Failures
   Rx SA Installation....................................12
   Tx SA Installation....................................0
```

Example: Displaying MKA Capability

The following example shows how to obtain MKA capability information for an interface:

```
switch# show macsec mka capability interface all
MKA capability information for interface(s)
--------- ---- ------ --------- --------- ------------ ------- ------- --- ----- ---- ----
Interface SGT  L3-Cap Sec-Pause Clr-Pause Fips-on-Asic MacSec  AES-256 XPN WinSz RxSA TxSA

--------- ---- ------ --------- --------- ------------ ------- ------- --- ----- ---- ----

Eth2/1    Yes   Yes    Yes       Yes       Yes          Yes      Yes     Yes 32    3    3
```

```
Eth2/2    Yes   Yes    Yes       Yes       Yes          Yes      Yes      Yes 32    3    3

Eth2/3    Yes   Yes    Yes       Yes       Yes          Yes      Yes      Yes 32    3    3
.
.
.
Eth2/48   Yes   Yes    Yes       Yes       Yes          Yes      Yes      Yes 32    3    3
```

13.10 Additional References for MKA

This sections provides additional information related to implementing MKA.

Related Documentation

Related Topic	**Document Title**
Inspur INOS licensing	*Inspur INOS Licensing Guide*
Command Reference	*Inspur CN12700 Series INOS Security Command Reference*

CHAPTER 14 Configuring IP ACLs

This chapter describes how to configure IP access control lists (ACLs) on Inspur INOS devices. Unless otherwise specified, the term IP ACL refers to IPv4 and IPv6 ACLs.

This chapter includes the following sections:

- Finding Feature Information
- Information About ACLs
- Licensing Requirements for IP ACLs
- Prerequisites for IP ACLs
- Guidelines and Limitations for IP ACLs
- Default Settings for IP ACLs
- Configuring IP ACLs
- Verifying the IP ACL Configuration
- Monitoring and Clearing IP ACL Statistics
- Configuration Examples for IP ACLs
- Configuring Object Groups
- Verifying the Object-Group Configuration
- Configuring Time Ranges
- Verifying the Time-Range Configuration
- Troubleshooting Flexible ACL TCAM Bank Chaining
- Additional References for IP ACLs
- Feature History for IP ACLs

14.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

14.2 Information About ACLs

An ACL is an ordered set of rules that you can use to filter traffic. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the device determines that an ACL applies to a packet, it tests the packet against the conditions of all rules. The first matching rule determines whether the packet is permitted or denied. If there is no match, the device applies the applicable implicit rule. The device continues processing packets that are permitted and drops packets that are denied.

You can use ACLs to protect networks and specific hosts from unnecessary or unwanted traffic. For example, you could use ACLs to disallow HTTP traffic from a high-security network to the Internet. You could also use ACLs to allow HTTP traffic but only to specific sites, using the IP address of the site to identify it in an IP ACL.

14.2.1 ACL Types and Applications

The device supports the following types of ACLs for security traffic filtering:

IPv4 ACLs

The device applies IPv4 ACLs only to IPv4 traffic.

IPv6 ACLs

The device applies IPv6 ACLs only to IPv6 traffic.

MAC ACLs

The device applies MAC ACLs only to non-IP traffic by default; however, you can configure Layer 2 interfaces to apply MAC ACLs to all traffic.

Security-group ACLs (SGACLs)

The device applies SGACLs to traffic tagged by Inspur TrustSec.

IP and MAC ACLs have the following types of applications:

Port ACL

Filters Layer 2 traffic

Router ACL

Filters Layer 3 traffic

VLAN ACL

Filters VLAN traffic
This table summarizes the applications for security ACLs.

Table 30 : Security ACL Applications

Application	Supported Interfaces	Types of ACLs Supported
Port ACL	• Layer 2 interfaces • Layer 2 Ethernet port-channel interfaces When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on the trunk port.	• IPv4 ACLs • IPv6 ACLs • MAC ACLs
Router ACL	• VLAN interfaces • Physical Layer 3 interfaces • Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Layer 3 Ethernet port-channel subinterfaces • Tunnels • Management interfaces • Starting from Inspur INOS Release 8.4(1), Router ACL is supported on Bridge domain interfaces. **Note** You must enable VLAN interfaces globally before you can configure a VLAN interface. For more information, see the *Inspur CN12700 Series INOS Interfaces Configuration Guide*.	• IPv4 ACLs • IPv6 ACLs **Note** MAC ACLs are supported on Layer 3 interfaces only if you enable MAC packet classification.

Application	Supported Interfaces	Types of ACLs Supported
VLAN ACL	• VLANs	• IPv4 ACLs • IPv6 ACLs • MAC ACLs

Related Topics

MAC Packet Classification
Information About MAC ACLs
Information About VLAN ACLs
SGACLs and SGTs .

14.2.2 Order of ACL Application

When the device processes a packet, it determines the forwarding path of the packet. The path determines which ACLs that the device applies to the traffic. The device applies the ACLs in the following order:

1. Port ACL
2. Ingress VACL
3. Ingress router ACL
4. SGACL
5. Egress router ACL
6. Egress VACL

If the packet is bridged within the ingress VLAN, the device does not apply router ACLs.

Figure 28： Order of ACL Application

The following figure shows the order in which the device applies ACLs.

Figure 29： ACLs and Packet Flow

The following figure shows where the device applies ACLs, depending upon the type of ACL. The red path indicates a packet sent to a destination on a different interface than its source. The blue path indicates a packet that is bridged within its VLAN.

The device applies only the applicable ACLs. For example, if the ingress port is a Layer 2 port and the traffic is on a VLAN that is a VLAN interface, a port ACL and a router ACL both can apply. In addition, if a VACL is applied to the VLAN, the device applies that ACL too.

Related Topics

SGACLs and SGTs .

14.2.3 About Rules

Rules are what you create, modify, and remove when you configure how an ACL filters network traffic. Rules appear in the running configuration. When you apply an ACL to an interface or change a rule within an ACL that is already applied to an interface, the supervisor module creates ACL entries from the rules in the running configuration and sends those ACL entries to the applicable I/O module. Depending upon how you configure the ACL, there may be more ACL entries than rules, especially if you implement policy-based ACLs by using object groups when you configure rules.

You can create rules in access-list configuration mode by using the **permit** or **deny** command. The device allows traffic that matches the criteria in a permit rule and blocks traffic that matches the criteria in a deny rule. You have many options for configuring the criteria that traffic must meet in order to match the rule.

This section describes some of the options that you can use when you configure a rule. For information about every option, see the applicable **permit** and **deny** commands in the *Inspur CN12700 Series INOS Security Command Reference*.

Protocols for IP ACLs

IPv4, IPv6, and MAC ACLs allow you to identify traffic by protocol. For your convenience, you can specify some protocols by name. For example, in an IPv4 or IPv6 ACL, you can specify ICMP by name.

You can specify any protocol by number. In MAC ACLs, you can specify protocols by the EtherType number of the protocol, which is a hexadecimal number. For example, you can use 0x0800 to specify IP traffic in a MAC ACL rule.

In IPv4 and IPv6 ACLs, you can specify protocols by the integer that represents the Internet protocol number. For example, you can use 115 to specify Layer 2 Tunneling Protocol (L2TP) traffic.

For a list of the protocols that each type of ACL supports by name, see the applicable **permit** and **deny** commands in the *Inspur CN12700 Series INOS Security Command Reference*.

Source and Destination

In each rule, you specify the source and the destination of the traffic that matches the rule. You can specify both the source and destination as a specific host, a network or group of hosts, or any host. How you specify the source and destination depends on whether you are configuring IPv4, IPv6, or MAC ACLs. For information about specifying the source and destination, see the applicable **permit** and **deny** commands in the *Inspur CN12700 Series INOS Security Command Reference*.

Implicit Rules for IP and MAC ACLs

IP and MAC ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the device applies them to traffic when no other rules in an ACL match. When you configure the device to maintain per-rule statistics for an ACL, the device does not maintain statistics for implicit rules.

All IPv4 ACLs include the following implicit rule:

```
deny ip any any
```

This implicit rule ensures that the device denies unmatched IP traffic. All IPv6 ACLs include the following implicit rules:

```
permit icmp any any nd-na
permit icmp any any nd-ns
permit icmp any any router-advertisement
permit icmp any any router-solicitation
deny ipv6 any any
```

Unless you configure an IPv6 ACL with a rule that denies ICMPv6 neighbor discovery messages, the first four rules ensure that the device permits neighbor discovery advertisement and solicitation messages. The fifth rule ensures that the device denies unmatched IPv6 traffic.

All MAC ACLs include the following implicit rule:

```
deny any any protocol
```

This implicit rule ensures that the device denies the unmatched traffic, regardless of the protocol specified in the Layer 2 header of the traffic.

Additional Filtering Options

You can identify traffic by using additional options. These options differ by ACL type. The following list includes most but not all additional filtering options:

- IPv4 ACLs support the following additional filtering options:
- Layer 4 protocol
- Authentication Header Protocol
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Encapsulating Security Payload
- General Routing Encapsulation (GRE)
- KA9Q NOS-compatible IP-over-IP tunneling
- Open Shortest Path First (OSPF)
- Payload Compression Protocol
- Protocol-independent multicast (PIM)
- TCP and UDP ports
- ICMP types and codes
- IGMP types
- Precedence level
- Differentiated Services Code Point (DSCP) value
- TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
- Established TCP connections
- Packet length
- IPv6 ACLs support the following additional filtering options:
- Layer 4 protocol
- Authentication Header Protocol
- Encapsulating Security Payload
- Payload Compression Protocol
- Stream Control Transmission Protocol (SCTP)
- SCTP, TCP, and UDP ports
- ICMP types and codes
- IGMP types
- Flow label
- DSCP value
- TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
- Established TCP connections
- Packet length
- MAC ACLs support the following additional filtering options:
- Layer 3 protocol
- VLAN ID
- Class of Service (CoS)

For information about all filtering options available in rules, see the applicable **permit** and **deny** commands in the *Inspur CN12700 Series INOS Security Command Reference*.

Sequence Numbers

The device supports sequence numbers for rules. Every rule that you enter receives a sequence number, either assigned by you or assigned automatically by the device. Sequence numbers simplify the following ACL tasks:

Adding new rules between existing rules

By specifying the sequence number, you specify where in the ACL a new rule should be positioned. For example, if you need to insert a rule between rules numbered 100 and 110, you could assign a sequence number of 105 to the new rule.

Removing a rule

Without using a sequence number, removing a rule requires that you enter the whole rule, as follows:

```
switch(config-acl)# no permit tcp 10.0.0.0/8 any
```

However, if the same rule had a sequence number of 101, removing the rule requires only the following command:

```
switch(config-acl)# no 101
```

Moving a rule

With sequence numbers, if you need to move a rule to a different position within an ACL, you can add a second instance of the rule using the sequence number that positions it correctly, and then you can remove the original instance of the rule. This action allows you to move the rule without disrupting traffic.

If you enter a rule without a sequence number, the device adds the rule to the end of the ACL and assigns a sequence number that is 10 greater than the sequence number of the preceding rule to the rule. For example, if the last rule in an ACL has a sequence number of 225 and you add a rule without a sequence number, the device assigns the sequence number 235 to the new rule.

In addition, Inspur INOS allows you to reassign sequence numbers to rules in an ACL. Resequencing is useful when an ACL has rules numbered contiguously, such as 100 and 101, and you need to insert one or more rules between those rules.

Logical Operators and Logical Operation Units

IP ACL rules for TCP and UDP traffic can use logical operators to filter traffic based on port numbers. The device stores operator-operand couples in registers called logical operator units (LOUs). Inspur CN12700 Series devices support 104 LOUs.

The LOU usage for each type of operator is as follows:

eq

Is never stored in an LOU

gt

Uses 1/2 LOU

lt

Uses 1/2 LOU

Neq

Uses 1/2 LOU

range

Uses 1 LOU

The following guidelines determine when the devices store operator-operand couples in LOUs:

- If the operator or operand differs from other operator-operand couples that are used in other rules, the couple is stored in an LOU.

For example, the operator-operand couples "gt 10" and "gt 11" would be stored separately in half an LOU each. The couples "gt 10" and "lt 10" would also be stored separately.

- Whether the operator-operand couple is applied to a source port or a destination port in the rule affects LOU usage. Identical couples are stored separately when one of the identical couples is applied to a source port and the other couple is applied to a destination port.

For example, if a rule applies the operator-operand couple "gt 10" to a source port and another rule applies a "gt 10" couple to a destination port, both couples would also be stored in half an LOU, resulting in the use of one whole LOU. Any additional rules using a "gt 10" couple would not result in further LOU usage.

Logging

You can enable the device to create an informational log message for packets that match a rule. The log message contains the following information about the packet:

- Protocol
- Status of whether the packet is a TCP, UDP, or ICMP packet, or if the packet is only a numbered packet.
- Source and destination address
- Source and destination port numbers, if applicable

Access Lists with Fragment Control

As non-initial fragments contain only Layer 3 information, these access-list entries containing only Layer 3 information, can now be applied to non-initial fragments also. The fragment has all the information the system requires to filter, so the access-list entry is applied to the fragments of a packet.

This feature adds the optional **fragments** keyword to the following IP access list commands: **deny (IPv4)**, **permit (IPv4)**, **deny (IPv6)**, **permit (IPv6)**. By specifying the **fragments** keyword in an access-list entry, that particular access-list entry applies only to non-initial fragments of packets; the fragment is either permitted or denied accordingly.

The behavior of access-list entries regarding the presence or absence of the **fragments** keyword can be summarized as follows:

If the Access-List Entry has...	Then...
...no **fragments** keyword and all of the access-list entry information matches	For an access-list entry containing only Layer 3 information: • The entry is applied to non-fragmented packets, initial fragments, and non-initial fragments. For an access-list entry containing Layer 3 and Layer 4 information: • The entry is applied to non-fragmented packets and initial fragments. • If the entry matches and is a permit statement, the packet or fragment is permitted. • If the entry matches and is a deny statement, the packet or fragment is denied. • The entry is also applied to non-initial fragments in the following manner. Because non-initial fragments contain only Layer 3 information, only the Layer 3

<table>
<tr><td></td><td>portion of an access-list entry can be applied. If the Layer 3 portion of the access-list entry matches, and
• If the entry is a permit statement, the non-initial fragment is permitted.
• If the entry is a deny statement, the next access-list entry is processed.
Note The deny statements are handled differently for non-initial fragments versus non-fragmented or initial fragments.</td></tr>
<tr><td>...the fragments keyword and all of the access-list entry information matches</td><td>The access-list entry is applied only to non-initial fragments.
Note The fragments keyword cannot be configured for an access-list entry that contains any Layer 4 information.</td></tr>
</table>

You should not add the **fragments** keyword to every access-list entry, because the first fragment of the IP packet is considered a non-fragment and is treated independently of the subsequent fragments. Because an initial fragment will not match an access list permit or deny entry that contains the **fragments** keyword, the packet is compared to the next access list entry until it is either permitted or denied by an access list entry that does not contain the **fragments** keyword. Therefore, you may need two access list entries for every deny entry. The first deny entry of the pair will not include the **fragments** keyword, and applies to the initial fragment. The second deny entry of the pair will include the **fragments** keyword and applies to the subsequent fragments. In the cases where there are multiple deny access list entries for the same host but with different Layer 4 ports, a single deny access-list entry with the **fragments** keyword for that host is all that has to be added. Thus all the fragments of a packet are handled in the same manner by the access list.

Packet fragments of IP datagrams are considered individual packets and each fragment counts individually as a packet in access-list accounting and access-list violation counts.

Policy Routing

14.2.4 Time Ranges

Fragmentation and the fragment control feature affect policy routing if the policy routing is based on the **match ip address** command and the access list had entries that match on Layer 4 through Layer 7 information. It is possible that noninitial fragments pass the access list and are policy routed, even if the first fragment was not policy routed or the reverse.

By using the **fragments** keyword in access-list entries as described earlier, a better match between the action taken for initial and noninitial fragments can be made and it is more likely policy routing will occur as intended.

You can use time ranges to control when an ACL rule is in effect. For example, if the device determines that a particular ACL applies to traffic arriving on an interface, and a rule in the ACL uses a time range that is not in effect, the device does not compare the traffic to that rule. The device evaluates time ranges based on its clock.

When you apply an ACL that uses time ranges, the device updates the affected I/O module whenever a time range referenced in the ACL starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.

IPv4, IPv6, and MAC ACLs support time ranges. When the device applies an ACL to traffic, the rules in effect

are as follows:
- All rules without a time range specified
- Rules with a time range that includes the second when the device applies the ACL to traffic

The device supports named, reusable time ranges, which allows you to configure a time range once and specify it by name when you configure many ACL rules. Time range names have a maximum length of 64 alphanumeric characters.

A time range contains one or more rules. The two types of rules are as follows:

Absolute

A rule with a specific start date and time, specific end date and time, both, or neither. The following items describe how the presence or absence of a start or end date and time affect whether an absolute time range rule is active:
- Start and end date and time both specified—The time range rule is active when the current time is later than the start date and time and earlier than the end date and time.
- Start date and time specified with no end date and time—The time range rule is active when the current time is later than the start date and time.
- No start date and time with end date and time specified—The time range rule is active when the current time is earlier than the end date and time.
- No start or end date and time specified—The time range rule is always active.

For example, you could prepare your network to allow access to a new subnet by specifying a time range that allows access beginning at midnight of the day that you plan to place the subnet online. You can use that time range in ACL rules that apply to the subnet. After the start time and date have passed, the device automatically begins applying the rules that use this time range when it applies the ACLs that contain the rules.

Periodic

A rule that is active one or more times per week. For example, you could use a periodic time range to allow access to a lab subnet only during work hours on weekdays. The device automatically applies ACL rules that use this time range only when the range is active and when it applies the ACLs that contain the rules.

Time ranges also allow you to include remarks, which you can use to insert comments into a time range. Remarks have a maximum length of 100 alphanumeric characters.

The device determines whether a time range is active as follows:
- The time range contains one or more absolute rules—The time range is active if the current time is within one or more absolute rules.
- The time range contains one or more periodic rules—The time range is active if the current time is within one or more periodic rules.
- The time range contains both absolute and periodic rules—The time range is active if the current time is within one or more absolute rules and within one or more periodic rules.

When a time range contains both absolute and periodic rules, the periodic rules can only be active when at least one absolute rule is active.

14.2.5 Policy-Based ACLs

The device supports policy-based ACLs (PBACLs), which allow you to apply access control policies across object groups. An object group is a group of IP addresses or a group of TCP or UDP ports. When you create a rule, you specify the object groups rather than specifying IP addresses or ports.

Using object groups when you configure IPv4 or IPv6 ACLs can help reduce the complexity of updating ACLs when you need to add or remove addresses or ports from the source or destination of rules. For example, if three rules reference the same IP address group object, you can add an IP address to the object instead of changing all three rules.

PBACLs do not reduce the resources required by an ACL when you apply it to an interface. When you apply a PBACL or update a PBACL that is already applied, the device expands each rule that refers to object groups into one ACL entry per object within the group. If a rule specifies the source and destination both with object groups, the number of ACL entries created on the I/O module when you apply the PBACL is equal to the number of objects in the source group multiplied by the number of objects in the destination group.

The following object group types apply to port, router, and VLAN ACLs:

IPv4 address object groups

Can be used with IPv4 ACL rules to specify source or destination addresses. When you use the **permit** or **deny** command to configure a rule, the **addrgroup** keyword allows you to specify an object group for the source or destination.

IPv6 address object groups

Can be used with IPv6 ACL rules to specify source or destination addresses. When you use the **permit** or **deny** command to configure a rule, the **addrgroup** keyword allows you to specify an object group for the source or destination.

Protocol port object groups

Can be used with IPv4 and IPv6 TCP and UDP rules to specify source or destination ports. When you use the **permit** or **deny** command to configure a rule, the **portgroup** keyword allows you to specify an object group for the source or destination.

14.2.6 Statistics and ACLs

The device can maintain global statistics for each rule that you configure in IPv4, IPv6, and MAC ACLs. If an ACL is applied to multiple interfaces, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that ACL is applied.

For each ACL that you configure, you can specify whether the device maintains statistics for that ACL, which allows you to turn ACL statistics on or off as needed to monitor traffic filtered by an ACL or to help troubleshoot the configuration of an ACL.

The device does not maintain statistics for implicit rules in an ACL. For example, the device does not maintain a count of packets that match the implicit **deny ip any any** rule at the end of all IPv4 ACLs. If you want to maintain statistics for implicit rules, you must explicitly configure the ACL with rules that are identical to the implicit rules.

Related Topics

Monitoring and Clearing IP ACL Statistics
Implicit Rules for IP and MAC ACLs.

14.2.7 Atomic ACLUpdates

An atomic ACL update is a hardware operation where both the existing ACL and the updated ACL are programmed in TCAM memory. This is the default mode of operation. The benefit of this update method is that ACL changes are not service impacting. When you make a change to the ACL, the current ACL is already programmed in TCAM. The Inspur CN12700 Series device will then take the current ACL and merge it with the changes to produce ACL prime. ACL prime will also be programmed into TCAM. The Inspur CN12700 Series device will then change the pointer so that ACL prime is associated with the interface. The final step is to delete the old ACL from TCAM. Functionally this means that you can never exceed 50 percent of ACL TCAM resources if you want to use atomic ACL updates. If you exceed 50 percent of ACL resources while atomic ACL update is active, the “ERROR: Tcam will be over used, please turn off atomic update” message is received and the new ACL changes are not applied.

Nonatomic ACL updates are required if you are using more than 50 percent of the ACL TCAM. When this mode is active, the Inspur CN12700 Series device will remove the old ACL from TCAM and replace it with ACL prime as quickly as possible. This allows you to use up to 100 percent of your ACL TCAM but has the disadvantage that it will cause a temporary interruption in service because packets that were permitted by the old ACL will be dropped until ACL prime can be successfully programed into the ACL TCAM.

By default, when a supervisor module of a Inspur CN12700 Series device updates an I/O module with changes to an ACL, it performs an atomic ACL update. An atomic update does not disrupt traffic that the updated ACL applies to; however, an atomic update requires that an I/O module that receives an ACL update has enough available resources to store each updated ACL entry in addition to all pre-existing entries in the affected ACL. After the update occurs, the additional resources used for the update are freed. If the I/O module lacks the required resources, the device generates an error message and the ACL update to the I/O module fails.

If an I/O module lacks the resources required for an atomic update, you can disable atomic updates by using the **no hardware access-list update atomic** command; however, during the brief time required for the device to remove the preexisting ACL and implement the updated ACL, traffic that the ACL applies to is dropped by default.

If you want to permit all traffic that an ACL applies to while it receives a nonatomic update, use the **hardware access-list update default-result permit** command.This example shows how to disable atomic updates to ACLs:

```
switch# config t
switch(config)# no hardware access-list update atomic
```

This example shows how to permit affected traffic during a nonatomic ACL update:

```
switch# config t
switch(config)# hardware access-list update default-result permit
```

This example shows how to revert to the atomic update method:

```
switch# config t
switch(config)# no hardware access-list update default-result permit
switch(config)# hardware access-list update atomic
```

Planning for Atomic ACL Updates

To adequately plan for Atomic ACL updates you need to be aware of how many ACE (Access Control Elements) you are using on all of your ACLs on each module. You also need to know how many ACEs your TCAM can support. You can find out your current usage with the **show hardware access-list resource utilization mod** *module-number* command.

```
show hardware access-list resource
utilization mod 3
 INSTANCE 0x0
 -------------
 ACL Hardware Resource Utilization (Mod 3)
 --------------------------------------------
               Used  Free  Percent
                           Utilization
 -----------------------------------------------------
 Tcam 0, Bank 0  1   16383  0.01
 Tcam 0, Bank 1  2   16382  0.01
 Tcam 1, Bank 0  7   16377  0.04
 Tcam 1, Bank 1  138 16246  0.84
```

For M-series modules, the ACL TCAM is spread across four banks. On non-XL modules, each bank has 16,000 entries for a total of 64K entries. On XL modules each bank has 32,000 entries for a total of 128,000 entries. Under normal circumstances, a single ACL will only use the resources of a single TCAM bank. In order to enable a single ACL to use resources from all of the banks you need to enable bank pooling with the **hardware access-list resource pooling module** *mod-numbe*r command.

You can verify that bank pooling is enabled with the **show hardware access-list resource pooling** command.

14.2.8 ACL TCAM Bank Mapping

ACL ternary control address memory (TCAM) bank mapping allows TCAM banks to accommodate more feature combinations in a more predictable manner. Features are preclassified into feature groups, which are further predefined into feature classes according to which features are allowed to coexist in a TCAM bank. For example, a port ACL (port ACL) feature and a Layer 2 NetFlow feature are defined as one feature class. These classes are allocated to specific banks. An error message appears if you enable or disable a feature class that is not supported on a specific TCAM bank.

ACL TCAM bank mapping allows you to configure a set of features at the same time and reduces multiple results that can accumulate when feature combinations that cannot coexist are configured on the same TCAM banks. By using this feature, you can optimize space and maximize the utilization of TCAM banks.

Beginning with Inspur INOS Release 8.2(3), you can issue the **show hardware access-list** {**input** | **output**} {**interface** | **vlan**} **feature-combo** *features* command to display the bank mapping matrix.

14.2.9 Flexible ACL TCAM Bank Chaining

In releases prior to Inspur INOS Release 8.2(3),the usage of ternary control address memory banks by an ACL were as follows:

- Single ACL using resources of a single TCAM bank.
- Single ACL using resources from all the TCAM banks with bank chaining mode enabled.

With bank chaining mode, you can have only single ACL result type per destination even though the ACL is not large enough to accommodate all the banks. However, the flexible bank chaining feature overcomes this limitation by allowing you to chain two TCAM banks and have two ACLs with two results per packet per direction. This helps you to handle larger ACLs that can be spread across multiple TCAM banks.

Consider the following scenarios with the F3 module; whose scale is 16K entries and each bank has 4K entries:

- Scenario 1–A PACL is configured and has 16K entries.

Solution–In this scenario, you should enable full bank chaining mode to use all the four TCAM banks to accommodate the PACL.

- Scenario 2–A PACL is configured on an L2 port and a RACL on a VLAN. Note that the L2 port is part of the VLAN. Each ACL has less than 8K entries.

Solution–The PACL and RACL combination is not supported by the full bank chaining mode. However, this combination is supported by the flexible TCAM bank chaining feature. PACL accommodates the two banks of first TCAM and RACL accommodates the two banks of second TCAM.

Flexible ACL TCAM Bank Chaining Modes

The flexible ACL TCAM bank chaining feature supports the following modes:

- VLAN-VLAN mode– This mode is used when you want to configure two VLAN features on a destination per direction. For example, when you have QoS and RACL features on a VLAN, use the VLAN-VLAN mode to accommodate the ACLs on the TCAMs.
- PORT-VLAN mode– This mode is used when you want to configure a port feature and a VLAN feature on a destination per direction. For example, when you have a NetFlow feature on a port and BFD on a VLAN, use the PORT-VLAN mode to accommodate the features on the TCAMs. For more examples, see Scenario 2.

You can check the features that are allocated to TCAM banks for VLAN-VLAN and PORT-VLAN modes in the bank mapping table. To display the TCAM bank mapping table, use the following command:

\# **show system internal access-list feature bank-chain map vlan-vlan** {**egress** | **ingress**}|**port-vlan** {**egress**| **{interface ingress| vlan ingress}} [module** *module-number***]**

The output displays the mapping table. You can check whether the feature result types overlap under the same

TCAM in the TCAM bank mapping. If a feature result types overlap, the configuration fails. For more information, see *Troubleshooting Flexible ACL TCAM Bank Chaining*.

You also check whether features can coexist in a TCAM bank. For example, a RACL feature and a Layer 2 NetFlow feature are defined as one feature class. These classes are allocated to specific banks. An error message appears if you enable or disable a feature class that is not supported on a specific TCAM bank. For more information, see *ACL TCAM Bank Mapping*.

Example: Displaying TCAM Bank Mapping

The following example displays the mapping output for VLAN-VLAN TCAM bank chaining mode:

```
switch# show system internal access-list feature bank-chain map vlan-vlan ingress module 3
```

Feature	Rslt Type	T0B0	T0B1	T1B0	T1B1
QoS	Qos	X	X		
RACL	Acl			X	X
PBR	Acl			X	X
VACL	Acl			X	X
DHCP	Acl			X	X
ARP	Acl			X	X
Netflow	Acl			X	X
Netflow (SVI)	Acl			X	X
Netflow Sampler	Acc	X	X		
Netflow Sampler (SVI)	Acc	X	X		
SPM WCCP	Acl			X	X
BFD	Acl			X	X
SPM OTV	Acl			X	X
ACLMGR ERSPAN (source)	Acl			X	X
SPM_VINCI_PROXY	Acl			X	X
SPM_VINCI_ANYCAST	Acl			X	X
SPM_VINCI_FABRIC_VLAN	Acl			X	X
SPM ITD	Acl			X	X
SPM EVPN ARP	Acl			X	X

Features that are displayed under the same TCAM bank, but have different result types, cannot be configured together. The output shows that you cannot configure the following feature combinations on TCAM0:

- QoS and Netflow Sampler
- Qos and Netflow Sampler (SVI)

For TCA, you can configure any feature combinations that does not include QoS, Netflow Sampler, and Netflow Sampler (SVI).

The following example displays the mapping output for PORT-VLAN TCAM bank chaining mode:

```
switch# show system internal access-list feature bank-chain map port-vlan ingress
```

Feature	Rslt Type	T0B0	T0B1	T1B0	T1B1
PACL	Acl	X	X		
RACL	Acl			X	X
DHCP	Acl			X	X
QoS	Qos	X	X		
PBR	Acl			X	X
VACL	Acl			X	X
Netflow	Acl			X	X
Netflow Sampler	Acc	X	X		

SPM WCCP	Acl			X	X
BFD	Acl			X	X
SPM OTV	Acl	X	X		
FEX	Acl	X	X		
SPM CBTS	Acl	X	X		
SPM LISP INST	Acl	X	X		
Openflow	Acl			X	X
SPM ITD	Acl			X	X

Consider the scenario when you configure the QoS feature in the ingress direction. However, if the QoS feature accommodates the TCAM0, then you cannot configure PACL, Netflow Sampler, SPM OTV, FEX, SPM CBTS, and SPM LISP INST features. Also, note that the PACL feature is only applicable at ingress.

The following example displays the mapping output for PORT-VLAN TCAM bank chaining mode for interface:

show system internal access-list feature bank-chain map port-vlan interface ingress

Feature	Rslt Type	T0B0	T0B1	T1B0	T1B1
PACL	Acl	X	X		
RACL	Acl			X	X
DHCP	Acl			X	X
DHCP_FHS	Acl	X	X		
DHCP_LDRA	Acl	X	X		
QoS	Qos	X	X		
PBR	Acl			X	X
Netflow	Acl			X	X
Netflow Sampler	Acc	X	X		
SPM WCCP	Acl			X	X
BFD	Acl			X	X
SPM OTV	Acl	X	X		
FEX	Acl	X	X		
SPM CBTS	Acl	X	X		
SPM LISP INST	Acl	X	X		
UDP RELAY	Acl			X	X
Openflow	Acl			X	X

The following example displays the mapping output for PORT-VLAN TCAM bank chaining mode for VLAN:

show system internal access-list feature bank-chain map port-vlan vlan ingress

```
Feature                 Rslt Type      T0B0       T0B1       T1B0       T1B1

QoS                         Qos                                 X          X
RACL                        Acl                                 X          X
PBR                         Acl                                 X          X
VACL                        Acl                                 X          X
DHCP                        Acl                                 X          X
DHCP_FHS                    Acl                                 X          X
DHCP_LDRA                   Acl                                 X          X
ARP                         Acl                                 X          X
Netflow                     Acl                                 X          X
Netflow (SVI)               Acl                                 X          X
Netflow Sampler             Acc                                 X          X
Netflow Sampler (SVI)       Acc                                 X          X
SPM WCCP                    Acl                                 X          X
BFD                         Acl                                 X          X
SPM OTV                     Acl                                 X          X
ACLMGR ERSPAN (source)      Acl                                 X          X
SPM_VINCI_PROXY             Acl                                 X          X
SPM_VINCI_ANYCAST           Acl                                 X          X
SPM_VINCI_FABRIC_VLAN       Acl                                 X          X
SPM ITD                     Acl                                 X          X
SPM EVPN ARP                Acl                                 X          X
UDP RELAY                   Acl                                 X          X
SPM_VXLAN_OAM               Acl                                 X          X
```

14.2.10 Session Manager Support for IP ACLs

Session Manager supports the configuration of IP and MAC ACLs. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration.

14.2.11 Virtualization Support for IP ACLs

The following information applies to IP and MAC ACLs used in virtual device contexts (VDCs):

- ACLs are unique per VDC. You cannot use an ACL that you created in one VDC in a different VDC.
- Because ACLs are not shared by VDCs, you can reuse ACL names in different VDCs.
- The device does not limit ACLs or rules on a per-VDC basis.
- Configuring atomic ACL updates must be performed in the default VDC but applies to all VDCs.

14.3 Licensing Requirements for IP ACLs

The following table shows the licensing requirements for this feature:

Product	**License Requirement**
Inspur INOS	No license is required to use IP ACLs. However to support up to 128K ACL entries using an XL line card, you must install the scalable services license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

14.4 Prerequisites for IP ACLs

IP ACLs have the following prerequisites:

- You must be familiar with IP addressing and protocols to configure IP ACLs.

- You must be familiar with the interface types that you want to configure with ACLs.

14.5 Guidelines and Limitations for IP ACLs

IP ACLs have the following configuration guidelines and limitations:

- When an access control list (ACL) is applied at the ingress of the original packet, it gets the destination index of the actual egress port and has no knowledge of the Encapsulated Remote Switched Port Analyzer (ERSPAN) session's point of egress at that moment. Because the packet does not go through the ACL engine after rewrite, it cannot be matched on ERSPAN packets.
- We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This is especially useful for ACLs that include more than about 1000 rules. For more information about Session Manager, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.
- In most cases, ACL processing for IP packets occurs on the I/O modules, which use hardware that accelerates ACL processing. In some circumstances, processing occurs on the supervisor module, which can result in slower ACL processing, especially during processing that involves an ACL with a large number of rules. Management interface traffic is always processed on the supervisor module. If IP packets in any of the following categories are exiting a Layer 3 interface, they are sent to the supervisor module for processing:
- Packets that fail the Layer 3 maximum transmission unit check and therefore require fragmenting.
- IPv4 packets that have IP options (additional IP packet header fields following the destination address field).
- IPv6 packets that have extended IPv6 header fields.

Rate limiters prevent redirected packets from overwhelming the supervisor module.

- When you apply an ACL that uses time ranges, the device updates the ACL entries on the affected I/O modules whenever a time range referenced in an ACL entry starts or ends. Updates that are initiated by time ranges occur on a best-effort priority. If the device is especially busy when a time range causes an update, the device may delay the update by up to a few seconds.
- To apply an IP ACL to a VLAN interface, you must have enabled VLAN interfaces globally. For more information about VLAN interfaces, see the *Inspur CN12700 Series INOS Interfaces Configuration Guide*
- The maximum number of supported IP ACL entries is 64,000 for devices without an XL line card and 128,000 for devices with an XL line card.
- If you try to apply too many ACL entries to a non-XL line card, the configuration is rejected.

The VTY ACL feature restricts all traffic for all VTY lines. You cannot specify different traffic restrictions for different VTY lines.

Any router ACL can be configured as a VTY ACL.

- ACLs configured for VTYs do not apply to the mgmt0 interface. Mgmt0 ACLs must be applied specifically to the interface.
- The Inspur CN 2000 Series Fabric Extender supports the full range of ingress ACLs that are available on its parent Inspur CN12700 Series device. For more information about the Fabric Extender, see the *Configuring the Inspur CN 2000 Series Fabric Extender*.
- ACL policies are not supported on the Fabric Extender fabric port channel.
- ACL capture is a hardware-assisted feature and is not supported for the management interface or for control packets originating in the supervisor. It is also not supported for software ACLs such as SNMP community ACLs and VTY ACLs.
- Enabling ACL capture disables ACL logging for all VDCs and the rate limiter for ACL logging.
- Port channels and supervisor in-band ports are not supported as a destination for ACL capture.
- ACL capture session destination interfaces do not support ingress forwarding and ingress MAC learning. If a destination interface is configured with these options, the monitor keeps the ACL capture session down. Use the **show monitor session all** command to see if ingress forwarding and MAC learning are enabled.

- The source port of the packet and the ACL capture destination port cannot be part of the same packet replication ASIC. If both ports belong to the same ASIC, the packet is not captured. The **show monitor session** command lists all the ports that are attached to the same ASIC as the ACL capture destination port.
- Only one ACL capture session can be active at any given time in the system across VDCs.
- If you configure an ACL capture monitor session before configuring the **hardware access-list capture** command, you must shut down the monitor session and bring it back up in order to start the session.
- When you apply an undefined ACL to an interface, the system treats the ACL as empty and permits all traffic.
- An IPv6 atomic policy update can be disruptive. It may cause disruption when there is an addition, deletion, or modification of an IPv6 source or destination address:
- Modifying the Layer 4 fields of the IPv6 ACE is not disruptive.
- Adding an IPv6 address may not always be disruptive, however, it can cause disruption in some cases.
- There may be disruption if you change the prefix length of an existing entry or add/delete the entry with a new prefix length.
- Resource pooling and ACL TCAM bank mapping cannot be enabled at the same time.
- You cannot configure the **mac packet-classify** command on shared interfaces.

Table 31： Protocol Number and Associated Layer 3 Protocol

Protocol Number	Layer 3 Protocol
1	ICMP
2	IGMP
4	IPv4 Encapsulation
6	TCP
17	UDP

- Two banks can be chained within the same TCAM. However, you cannot chain banks across multiple TCAMs.
- The bank chaining and bank mapping features cannot co-exist.
- You cannot configure port ACL features such as PACL, L2 QOS, and L2 Netflow when you enable the VLAN-VLAN mode for configuring the flexible ACL TCAM bank chaining feature.
- Enabling the flexible ACL TCAM bank chaining feature on all the modules is not supported.
- F3 Series Module
- The forwarding engines in an F3 Series module has 16,000 total TCAM entries that are equally split across two banks.
- F3 Series modules supports ACL capture.
- F3 Series modules supports FCoE ACLs.
- For F3 Series modules, the log option in egress ACLs is not supported for multicast packets.
- If an F3 Series module is shared among different VDCs, any egress ACL that is configured on one VDC is pushed to the other VDCs.
- For F3 Series modules, the **mac packet-classify** command enables a MAC ACL for port policies but an IPv4 or IPv6 ACL for VLAN policies.
- Two banks can be chained within the same TCAM. However, you cannot chain banks across multiple TCAMs.
- The bank chaining and bank mapping features cannot co-exist.
- You cannot configure port ACL features such as PACL, L2 QOS, and L2 Netflow when you enable the VLAN-VLAN mode for configuring the flexible ACL TCAM bank chaining feature.
- The flexible ACL TCAM bank chaining feature is supported only on the F3 Series modules. Enabling the flexible ACL TCAM bank chaining feature on all the modules is not supported.

ACLs on VTY lines have the following guidelines and limitations:

- ACLs applied on a VTY line in egress direction filter traffic without any issues. However, ACLs applied on a

VTY line in ingress direction will not filter management traffic. For example, FTP, TFTP, or SFP traffic in the return direction, that is, if the FTP connection is initiated from a switch to an external server, ingress ACL on a VTY line will not be used, if ACLs are configured to block or permit this return traffic. Therefore, ACLs should be applied in the egress direction on VTY lines to block the FTP, TFTP, or SCP traffic from the switch.

- It is recommended to use ACLs on management interface as well to secure access to the switch from secured and permitted sources.

14.6 Default Settings for IP ACLs

This table lists the default settings for IP ACL parameters.

Table 32 : Default IP ACL Parameters

Parameters	**Default**
IP ACLs	No IP ACLs exist by default
ACL rules	Implicit rules apply to all ACLs
Object groups	No object groups exist by default
Time ranges	No time ranges exist by default
ACL TCAM bank mapping	Disabled

Related Topics

Implicit Rules for IP and MAC ACLs.

14.7 Configuring IP ACLs

14.7.1 Creating an IP ACL

You can create an IPv4 ACL or IPv6 ACL on the device and add rules to it.

Before you begin

We recommend that you perform the ACL configuration using the Session Manager. This feature allows you to verify the ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules. For more information about Session Manager, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **ip access-list *name***
   - **ipv6 access-list** *name*
3. (Optional) **fragments** {**permit-all** | **deny-all**}
4. [*sequence-number*] {**permit** | **deny**} *protocol source destination*
5. (Optional) **statistics per-entry**
6. (Optional) Enter one of the following commands:
   - **show ip access-lists** *name*

• **show ipv6 access-lists** ***name***

7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	• **ip access-list** *name*
	• **ipv6 access-list** *name*
	Example:
	`switch(config)# ip access-list acl-01`
	`switch(config-acl)#`
		Creates the IP ACL and enters IP ACL configuration mode. The *name* argument can be up to 64 characters.

Step 3	(Optional) **fragments** {**permit-all** | **deny-all**} **Example:** `switch(config-acl)# fragments permit-all`	Optimizes fragment handling for noninitial fragments. When a device applies to traffic an ACL that contains the **fragments** command, the **fragments** command only matches noninitial fragments that do not match any explicit **permit** or **deny** commands in the ACL.
Step 4	[*sequence-number*] {**permit** | **deny**} *protocol source destination* **Example:** `switch(config-acl)# permit ip 192.168.2.0/24 any`	Creates a rule in the IP ACL. You can create many rules. The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Inspur CN12700 Series INOS Security Command Reference*.
Step 5	(Optional) **statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 6	(Optional) Enter one of the following commands: • **show ip access-lists** *name* • **show ipv6 access-lists** *name* **Example:** `switch(config-acl)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

14.7.2 Changing an IP ACL

You can add and remove rules in an existing IPv4 or IPv6 ACL, but you cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers.

Before you begin

We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules. For more information about Session Manager, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **ip access-list** *name*
   - **ipv6 access-list** *name*
3. (Optional) [*sequence-number*] {**permit** | **deny**} *protocol source destination*
4. (Optional) [**no**] **fragments** {**permit-all** | **deny-all**}
5. (Optional) **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*}
6. (Optional) [**no**] **statistics per-entry**
7. (Optional) Enter one of the following commands:
   - **show ip access-lists** *name*
   - **show ipv6 access-lists** *name*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	• **ip access-list** *name*
	• **ipv6 access-list** *name*
	Example:
	`switch(config)# ip access-list acl-01`
	`switch(config-acl)#`
		Enters IP ACL configuration mode for the ACL that you specify by name.

Step 3	(Optional) [*sequence-number*] {**permit** | **deny**} *protocol source destination* **Example:** `switch(config-acl)# 100 permit ip 192.168.2.0/24 any`	Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number between 1 and 4294967295.

	Command or Action	Purpose
		The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.
Step 4	(Optional) [**no**] **fragments** {**permit-all** | **deny-all**} **Example:** `switch(config-acl)# fragments permit-all`	Optimizes fragment handling for noninitial fragments. When a device applies to traffic an ACL that contains the **fragments** command, the **fragments** command only matches noninitial fragments that do not match any explicit **permit** or **deny** commands in the ACL. The **no** option removes fragment-handling optimization.
Step 5	(Optional) **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination*} **Example:** `switch(config-acl)# no 80`	Removes the rule that you specified from the IP ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Inspur CN12700 Series INOS Security Command Reference*.
Step 6	(Optional) [**no**] **statistics per-entry** **Example:** `switch(config-acl)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 7	(Optional) Enter one of the following commands: • **show ip access-lists** *name* • **show ipv6 access-lists** *name* **Example:** `switch(config-acl)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Changing Sequence Numbers in an IP ACL.

14.7.3 Changing Sequence Numbers in an IP ACL

You can change all the sequence numbers assigned to the rules in an IP ACL.

Before you begin

We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules. For more information about Session Manager, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **resequence** {**ip** | **ipv6**} **access-list** *name starting-sequence-number increment*
3. (Optional) **show ip access-lists** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**resequence** {**ip** | **ipv6**} **access-list** *name starting-sequence-number increment* **Example:** `switch(config)# resequence access-list ip acl-01 100 10`	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The *starting-sequence-number* argument and the *increment* argument can be a whole number between 1 and 4294967295.
Step 3	(Optional) **show ip access-lists** *name* **Example:** `switch(config)# show ip access-lists acl-01`	Displays the IP ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

14.7.4 Removing an IP ACL

You can remove an IP ACL from the device.

Before you begin

Ensure that you know whether the ACL is applied to an interface. The device allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the device considers the removed ACL to be empty. Use the **show ip access-lists** command or the **show ipv6 access-lists** command with the summary keyword to find the interfaces that an IP ACL is configured on.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:

   • **no ip access-list** *name*

- **no ipv6 access-list** *name*

3. (Optional) Enter one of the following commands:
   - **show ip access-lists** *name* **summary**
   - **show ipv6 access-lists** *name* **summary**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands: • **no ip access-list** *name* • **no ipv6 access-list** *name* **Example:** `switch(config)# no ip access-list acl-01`	Removes the IP ACL that you specified by name from the running configuration.
Step 3	(Optional) Enter one of the following commands: • **show ip access-lists** *name* **summary** • **show ipv6 access-lists** *name* **summary** **Example:** `switch(config)# show ip access-lists acl-01 summary`	Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

14.7.5 Applying an IP ACL as a Router ACL

You can apply an IPv4 or IPv6 ACL to any of the following types of interfaces:

- Physical Layer 3 interfaces and subinterfaces
- Layer 3 Ethernet port-channel interfaces and subinterfaces
- VLAN interfaces
- Tunnels
- Management interfaces
- Bridge domain interfaces

ACLs applied to these interface types are considered router ACLs.

Before you begin

Ensure that the ACL you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. switch# **configure terminal**
2. Enter one of the following commands:
- switch(config)# **interface ethernet** *slot*/*port*[**.** *number*]
- switch(config)# **interface port-channel** *channel-number*[**.** *number*]
- switch(config)# **interface tunnel** *tunnel-number*
- switch(config)# **interface vlan** *vlan-ID*
- switch(config)# **interface mgmt** *port*
- switch(config)# **interface bdi** *number*

3. Enter one of the following commands:
- switch(config-if)# **ip access-group** *access-list* {**in** | **out**}
- switch(config-if)# **ipv6 traffic-filter** *access-list* {**in** | **out**}

4. (Optional) switch(config-if)# **show running-config aclmgr**
5. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	Enter one of the following commands: • switch(config)# **interface ethernet** *slot*/*port*[**.** *number*] • switch(config)# **interface port-channel** *channel-number*[**.** *number*] • switch(config)# **interface tunnel** *tunnel-number* • switch(config)# **interface vlan** *vlan-ID* • switch(config)# **interface mgmt** *port* • switch(config)# **interface bdi** *number*	Enters configuration mode for the interface type that you specified.
Step 3	Enter one of the following commands: • switch(config-if)# **ip access-group** *access-list* {**in** | **out**} • switch(config-if)# **ipv6 traffic-filter** *access-list* {**in** | **out**}	Applies an IPv4 or IPv6 ACL to the Layer 3 interface for traffic flowing in the direction specified. You can apply one router ACL per direction.
Step 4	(Optional) switch(config-if)# **show running-config aclmgr**	Displays the ACL configuration.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Related Topics

Creating an IP ACL.

14.7.6 Applying an IP ACL as a Port ACL

You can apply an IPv4 or IPv6 ACL to a Layer 2 interface, which can be a physical port or a port channel. ACLs applied to these interface types are considered port ACLs.

Before you begin

Ensure that the ACL you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *slot*/*port*
   - **interface port-channel** *channel-number*
3. Enter one of the following commands:
   - **ip port access-group** *access-list* **in**
   - **ipv6 port traffic-filter** *access-list* **in**
4. (Optional) **show running-config aclmgr**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		Enters configuration mode for the interface type that you specified.

Step 3	Enter one of the following commands: • **ip port access-group** *access-list* **in** • **ipv6 port traffic-filter** *access-list* **in**	Applies an IPv4 or IPv6 ACL to the interface or port channel. Only inbound filtering is supported with port ACLs. You can apply one port ACL to an interface.
	Example: `switch(config-if)# ip port access-group acl-l2-marketing-group in`	
Step 4	(Optional) **show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	Displays the ACL configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config`	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	startup-config	

Related Topics

Creating an IP ACL.
Enabling or Disabling MAC Packet Classification.

14.7.7 Applying an IP ACL as a VACL

You can apply an IP ACL as a VACL.

Related Topics

Configuring VACLs.

14.7.8 Configuring ACL TCAM Bank Mapping

You can configure the device to allow ACL TCAM bank mapping. This feature allows TCAM banks to accommodate feature combinations in a more predictable manner.

Before you begin

Ensure that you are in the default VDC (or use the **switchto** command).

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **hardware access-list resource feature bank-mapping**
3. **show hardware access-list** {**input** | **output**} {**interface** | **vlan** } **feature-combo** *features*
4. (Optional) **show system internal access-list feature bank-class map** {**ingress** | **egress**} [**module** *module*]
5. **copy running-config startup-config**

DETAILED STEPS

Step 1 **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 [**no**] **hardware access-list resource feature bank-mapping**

Example:

```
switch(config)# hardware access-list resource feature bank-mapping
```

Enables ACL TCAM bank mapping for feature groups and classes.

Note This command is available only in the default VDC but applies to all VDCs.

Step 3 **show hardware access-list** {**input** | **output**} {**interface** | **vlan** } **feature-combo** *features*

Example:

```
switch# show hardware access-list input vlan feature-combo pacl
```

Feature	Rslt Type	T0B0	T0B1	T1B0	T1B1
PACL	Acl	X			

```
QoS                                Qos                    X
```

Displays the bank mapping matrix.

Step 4 (Optional) **show system internal access-list feature bank-class map** {**ingress** | **egress**} [**module** *module*]

Example:

```
switch(config)# show system internal access-list feature bank-class map ingress module 4

Feature Class Definition:
1. CLASS_QOS : QoS,
2. CLASS_INBAND :
Tunnel Decap, SPM LISP, SPM ERSPAN (termination),
3. CLASS_PACL    :    PACL,
Netflow,
4. CLASS_DHCP :
DHCP, Netflow, ARP, VACL,
5. CLASS_RACL :
RACL, RACL_STAT, Netflow (SVI), ARP,
6. CLASS_VACL :
VACL, VACL_STAT, ARP, FEX, Netflow,
7. CLASS_RV_ACL :
RACL, PBR, BFD, ARP, SPM WCCP, VACL, SPM OTV, FEX, CTS implicit Tunnel
```

Displays the feature group and class combination tables.

Step 5 **copy running-config startup-config**

Example:

```
switch# copy running-config startup-config
```

Copies the running configuration to the startup configuration.

14.7.9 Configuring Flexible ACL TCAM Bank Chaining

Use this task to configure the flexible ACL TCAM bank chaining feature.

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Enable the flexible TCAM bank chaining feature:

switch(config)# **hardware access-list resource pooling {vlan-vlan|port-vlan} module** *module-number*

Step 3 Exit global configuration mode:
switch(config)# **exit**

Step 4 Required: Display the flexible TCAM bank chaining mode:
switch# **show system internal access-list globals**

Step 5 (Optional) Display the flexible TCAM bank mapping table:

switch# **show system internal access-list feature bank-chain map vlan-vlan** {**egress | ingress**}|**port-vlan** {**egress**| {**interface ingress**| **vlan ingress**}} [**module** ***module-number***]

Configuring Flexible ACL TCAM Bank Chaining

The following running configuration shows how to configure flexible ACL TCAM bank chaining feature with VLAN-VLAN mode for module 3. Replace the placeholders with relevant values for your setup.

```
configure terminal
  hardware access-list resource pooling <vlan-vlan> module <3>
 exit
```

The following example shows how to check the TCAM bank chaining mode:

```
switch# show system internal access-list globals
slot  3
=======
  Atomic Update : ENABLED
  Default ACL   : DENY
  Bank Chaining : VLAN-VLAN
  Seq Feat Model : NO_DENY_ACE_SUPPORT
  This pltfm supports seq feat model
  Bank Class Model : DISABLED
  This pltfm supports bank class model
  Fabric path DNL : DISABLED
  Seq Feat Model : NO_DENY_ACE_SUPPORT
  This pltfm supports seq feat model

  L4 proto CAM extend : DISABLED
  This pltfm supports L4 proto CAM extend
  MPLS Topmost As Pipe Mode : DISABLED

  This pltfm supports mpls topmost as pipe mode
  LOU Threshold Value : 5
```

The following example displays the mapping output for the VLAN-VLAN mode:

```
switch# show system internal access-list feature bank-chain map vlan-vlan egress
```

Feature	Rslt Type	T0B0	T0B1	T1B0	T1B1
QoS	Qos	X	X		
RACL	Acl			X	X
VACL	Acl			X	X
Tunnel Decap	Acl	X	X		
Netflow	Acl			X	X
Netflow Sampler	Acc	X	X		
Rbacl	Acl	X	X		
CTS implicit Tunnel	Acl	X	X		
SPM WCCP	Acl			X	X
SPM OTV	Acl	X	X		
SPM LISP	Acl	X	X		
SPM ERSPAN (termination)	Acl	X	X		
OTV25 DECAP	Acl	X	X		
SPM NVE	Acl			X	X
SPM NVE RDT	Acl			X	X
SPM ITD	Acl			X	X

14.8 Verifying the IP ACL Configuration

To display IP ACL configuration information, perform one of the following tasks. For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show ip access-lists	Displays the IPv4 ACL configuration.

Command	Purpose
show ipv6 access-lists	Displays the IPv6 ACL configuration.
show system internal access-list feature bank-class map {**ingress** | **egress**} [**module** *module*]	Displays the feature group and class combination tables.
show running-config aclmgr [**all**]	Displays the ACL running configuration, including the IP ACL configuration and the interfaces to which IP ACLs are applied.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration.

14.9 Monitoring and Clearing IP ACL Statistics

To monitor or clear IP ACL statistics, use one of the commands in this table. For detailed information about these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show ip access-lists	Displays the IPv4 ACL configuration. If the IPv4 ACL includes the **statistics per-entry** command, the **show ip access-lists** command output includes the number of packets that have matched each rule.
show ipv6 access-lists	Displays IPv6 ACL configuration. If the IPv6 ACL includes the **statistics per-entry** command, then the **show ipv6 access-lists** command output includes the number of packets that have matched each rule.
clear ip access-list counters	Clears statistics for all IPv4 ACLs or for a specific IPv4 ACL.
clear ipv6 access-list counters	Clears statistics for all IPv6 ACLs or for a specific IPv6 ACL.

14.10 Configuration Examples for IPACLs

The following example shows how to create an IPv4 ACL named acl-01 and apply it as a port ACL to Ethernet interface 2/1, which is a Layer 2 interface:

```
ip access-list acl-01
  permit ip 192.168.2.0/24 any
interface ethernet 2/1
  ip port access-group acl-01 in
```

The following example shows how to create an IPv6 ACL named acl-120 and apply it as a router ACL to Ethernet interface 2/3, which is a Layer 3 interface:

```
ipv6 access-list acl-120
  permit tcp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
  permit udp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
  permit tcp 2001:0db8:69f2::/48 2001:0db8:be03:2112::/64
  permit udp 2001:0db8:69f2::/48 2001:0db8:be03:2112::/64
interface ethernet 2/3
  ipv6 traffic-filter acl-120 in
```

14.11 ConfiguringObjectGroups

You can use object groups to specify source and destination addresses and protocol ports in IPv4 ACL and IPv6 ACL rules.

14.11.1 Session Manager Support for Object Groups

Session Manager supports the configuration of object groups. This feature allows you to create a configuration session and verify your object group configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.

14.11.2 Creating and Changing an IPv4 Address Object Group

You can create and change an IPv4 address group object.

SUMMARY STEPS

1. **configure terminal**
2. **object-group ip address** *name*
3. Enter one of the following commands:

- [*sequence-number*] **host** *IPv4-address*
- [*sequence-number*] *IPv4-address network-wildcard*
- [*sequence-number*] *IPv4-address/prefix-len*

4. Enter one of the following commands:

- **no** [*sequence-number*]
- **no host** *IPv4-address*
- **no** *IPv4-address network-wildcard*
- **no** *IPv4-address/prefix-len*

5. (Optional) **show object-group** *name*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**object-group ip address** *name*
	Example:
	`switch(config)# object-group ip address ipv4-addr-group-13`
	`switch(config-ipaddr-ogroup)#`
		Creates the IPv4 address object group and enters IPv4 address object-group configuration mode.

Step 3	Enter one of the following commands: • [*sequence-number*] **host** *IPv4-address* • [*sequence-number*] *IPv4-address network-wildcard* • [*sequence-number*] *IPv4-address/prefix-len*	Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host or omit the **host** command to specify a network of hosts.

	Command or Action	Purpose
	Example: `switch(config-ipaddr-ogroup)# host 10.99.32.6`	
Step 4	Enter one of the following commands: • **no** [*sequence-number*] • **no host** *IPv4-address* • **no** *IPv4-address network-wildcard* • **no** *IPv4-address/prefix-len* **Example:** `switch(config-ipaddr-ogroup)# no host 10.99.32.6`	Removes an entry in the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.
Step 5	(Optional) **show object-group** *name* **Example:** `switch(config-ipaddr-ogroup)# show object-group ipv4-addr-group-13`	Displays the object group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-ipaddr-ogroup)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

14.11.3 Creating and Changing an IPv6 Address Object Group

You can create and change an IPv6 address group object.

SUMMARY STEPS

1. **config t**
2. **object-group ipv6 address** *name*
3. Enter one of the following commands:

- [*sequence-number*] **host** *IPv6-address*
- [*sequence-number*] *IPv6-address/prefix-len*

4. Enter one of the following commands:

- **no** *sequence-number*
- **no host** *IPv6-address*
- **no** *IPv6-address/prefix-len*

5. (Optional) **show object-group** *name*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**object-group ipv6 address** *name*
	Example:
	`switch(config)# object-group ipv6 address ipv6-addr-group-A7`
	`switch(config-ipv6addr-ogroup)#`
		Creates the IPv6 address object group and enters IPv6 address object-group configuration mode.

Step 3	Enter one of the following commands: • [*sequence-number*] **host** *IPv6-address* • [*sequence-number*] *IPv6-address/prefix-len* **Example:** `switch(config-ipv6addr-ogroup)# host 2001:db8:0:3ab0::1`	Creates an entry in the object group. For each entry that you want to create, use the **host** command and specify a single host or omit the **host** command specify a network of hosts.
Step 4	Enter one of the following commands: • **no** *sequence-number* • **no host** *IPv6-address* • **no** *IPv6-address/prefix-len* **Example:** `switch(config-ipv6addr-ogroup)# no host 2001:db8:0:3ab0::1`	Removes an entry from the object group. For each entry that you want to remove from the object group, use the **no** form of the **host** command.
Step 5	(Optional) **show object-group** *name* **Example:** `switch(config-ipv6addr-ogroup)# show object-group ipv6-addr-group-A7`	Displays the object group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-ipv6addr-ogroup)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

14.11.4 Creating and Changing a Protocol Port Object Group

You can create and change a protocol port object group.

SUMMARY STEPS

1. **configure terminal**
2. **object-group ip port** *name*
3. [*sequence-number*] *operator port-number* [*port-number*]
4. **no** {*sequence-number* | *operator port-number* [*port-number*]}
5. (Optional) **show object-group** *name*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	

Step 2	**object-group ip port** *name*
	Example:
	`switch(config)# object-group ip port NYC-datacenter-ports`
	`switch(config-port-ogroup)#`
		Creates the protocol port object group and enters port object-group configuration mode.

Step 3	[*sequence-number*] *operator port-number* [*port-number*] **Example:** `switch(config-port-ogroup)# eq 80`	Creates an entry in the object group. For each entry that you want to create, use one of the following operator commands: • **eq**—Matches the port number that you specify only. • **gt**—Matches port numbers that are greater than (and not equal to) the port number that you specify. • **lt**—Matches port numbers that are less than (and not equal to) the port number that you specify. • **neq**—Matches all port numbers except for the port number that you specify. • **range**—Matches the range of port number between and including the two port numbers that you specify. **Note** The **range** command is the only operator command that requires two *port-number* arguments.
Step 4	**no** {*sequence-number* | *operator port-number* [*port-number*]} **Example:** `switch(config-port-ogroup)# no eq 80`	Removes an entry from the object group. For each entry that you want to remove, use the **no** form of the applicable operator command.
Step 5	(Optional) **show object-group** *name* **Example:** `switch(config-port-ogroup)# show object-group NYC-datacenter-ports`	Displays the object group configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-port-ogroup)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

14.11.5 Removing an Object Group

You can remove an IPv4 address object group, an IPv6 address object group, or a protocol port object group.

SUMMARY STEPS

1. **configure terminal**
2. **no object-group** {**ip address** | **ipv6 address** | **ip port**} *name*
3. (Optional) **show object-group**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no object-group** {**ip address** | **ipv6 address** | **ip port**} *name* **Example:** `switch(config)# no object-group ip address ipv4-addr-group-A7`	Removes the object group that you specified.
Step 3	(Optional) **show object-group** **Example:** `switch(config)# show object-group`	Displays all object groups. The removed object group should not appear.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

14.12 Verifying the Object-Group Configuration

To display object-group configuration information, perform one of the following tasks:

Command	Purpose
show object-group	Displays the object-group configuration.
show running-config aclmgr	Displays ACL configuration, including object groups.

For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

14.13 Configuring Time Ranges

14.13.1 Session Manager Support for Time Ranges

Session Manager supports the configuration of time ranges. This feature allows you to create a configuration session and verify your time-range configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.

14.13.2 Creating a Time Range

You can create a time range on the device and add rules to it.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **configure terminal**
2. **time-range** *name*
3. (Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*
4. (Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time*
5. (Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*]
6. (Optional) [*sequence-number*] **absolute** [start *time date*] **end** *time date*
7. (Optional) **show time-range** *name*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**time-range** *name*
	Example:
	`switch(config)# time-range workday-daytime`
	`switch(config-time-range)#`
		Creates the time range and enters time-range configuration mode.

Step 3	(Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time* **Example:** `switch(config-time-range)# periodic monday 00:00:00 to friday 23:59:59`	Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times.
Step 4	(Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time* **Example:** `switch(config-time-range)# periodic weekdays 06:00:00 to 20:00:00`	Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument: • **daily** —All days of the week. • **weekdays** —Monday through Friday. • **weekend** —Saturday through Sunday.
Step 5	(Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*]	Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you omit the **end** keyword, the rule is always in effect after the start time and date have

	Command or Action	Purpose
	Example: `switch(config-time-range)# absolute start 1:00 15 march 2008`	passed.
Step 6	(Optional) [*sequence-number*] **absolute** [start *time date*] **end** *time date* **Example:** `switch(config-time-range)# absolute end 23:59:59 31 december 2008`	Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed.
Step 7	(Optional) **show time-range** *name* **Example:** `switch(config-time-range)# show time-range workday-daytime`	Displays the time-range configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config-time-range)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

14.13.3 Changing a Time Range

You can add and remove rules in an existing time range. You cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **configure terminal**
2. **time-range** *name*
3. (Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time*
4. (Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time*
5. (Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*]
6. (Optional) [*sequence-number*] **absolute** [**start** *time date*] **end** *time date*
7. (Optional) **no** {*sequence-number* | **periodic** *arguments . . .* | **absolute** *arguments. . .*}
8. (Optional) **show time-range** *name*
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	

Step 2	**time-range** *name*
	Example:
	`switch(config)# time-range workday-daytime`
	`switch(config-time-range)#`
		Enters time-range configuration mode for the specified time range.

Step 3	(Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time* **Example:** `switch(config-time-range)# periodic monday 00:00:00 to friday 23:59:59`	Creates a periodic rule that is in effect for one or more contiguous days between and including the specified start and end days and times.
Step 4	(Optional) [*sequence-number*] **periodic** *list-of-weekdays time* **to** *time* **Example:** `switch(config-time-range)# 100 periodic weekdays 05:00:00 to 22:00:00`	Creates a periodic rule that is in effect on the days specified by the *list-of-weekdays* argument between and including the specified start and end times. The following keywords are also valid values for the *list-of-weekdays* argument: • **daily** —All days of the week. • **weekdays** —Monday through Friday. • **weekend** —Saturday through Sunday.
Step 5	(Optional) [*sequence-number*] **absolute start** *time date* [**end** *time date*] **Example:** `switch(config-time-range)# absolute start 1:00 15 march 2008`	Creates an absolute rule that is in effect beginning at the time and date specified after the **start** keyword. If you omit the **end** keyword, the rule is always in effect after the start time and date have passed.
Step 6	(Optional) [*sequence-number*] **absolute** [**start** *time date*] **end** *time date* **Example:** `switch(config-time-range)# absolute end 23:59:59 31 december 2008`	Creates an absolute rule that is in effect until the time and date specified after the **end** keyword. If you omit the **start** keyword, the rule is always in effect until the end time and date have passed.
Step 7	(Optional) **no** {*sequence-number* | **periodic** *arguments . . .* | **absolute** *arguments. . .*} **Example:** `switch(config-time-range)# no 80`	Removes the specified rule from the time range.
Step 8	(Optional) **show time-range** *name* **Example:** `switch(config-time-range)# show time-range workday-daytime`	Displays the time-range configuration.
Step 9	(Optional) **copy running-config startup-config**	Copies the running configuration to the

	Command or Action	Purpose
	Example: `switch(config-time-range)# copy running-config startup-config`	startup configuration.

Related Topics

Changing Sequence Numbers in a Time Range.

14.13.4 Removing a Time Range

You can remove a time range from the device.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

Ensure that you know whether the time range is used in any ACL rules. The device allows you to remove time ranges that are used in ACL rules. Removing a time range that is in use in an ACL rule does not affect the configuration of interfaces where you have applied the ACL. Instead, the device considers the ACL rule using the removed time range to be empty.

SUMMARY STEPS

1. **configure terminal**
2. **no time-range** *name*
3. (Optional) **show time-range**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no time-range** *name* **Example:** `switch(config)# no time-range daily-workhours`	Removes the time range that you specified by name.
Step 3	(Optional) **show time-range** **Example:** `switch(config-time-range)# show time-range`	Displays the configuration for all time ranges. The removed time range should not appear.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

14.13.5 Changing Sequence Numbers in a Time Range

You can change all the sequence numbers assigned to rules in a time range.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Because ACL names can be repeated in different VDCs, we recommend that you confirm which VDC you are working in.

SUMMARY STEPS

1. **configure terminal**
2. **resequence time-range** *name starting-sequence-number increment*
3. (Optional) **show time-range** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**resequence time-range** *name starting-sequence-number increment* **Example:** `switch(config)# resequence time-range daily-workhours 100 10 switch(config)#`	Assigns sequence numbers to the rules contained in the time range, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify.
Step 3	(Optional) **show time-range** *name* **Example:** `switch(config)# show time-range daily-workhours`	Displays the time-range configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

14.14 Verifying the Time-Range Configuration

To display time-range configuration information, perform one of the following tasks. For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show time-range	Displays the time-range configuration.
show running-config aclmgr	Displays ACL configuration, including all time ranges.

14.15 Troubleshooting Flexible ACL TCAM Bank Chaining

Problem: The configuration of a feature on a VLAN or a port fails.

Scenario: The flexible ACL TCAM bank chaining feature is configured with the VLAN-VLAN mode on module 2. The QoS feature on the destination VLAN is configured. Additionally, the role-based access control list

(RBACL) should be configured on the same VLAN. In this case, the configuration of the RBACL feature fails.

Solution: Check whether the feature result types overlap under the same TCAM in the TCAM bank mapping table, as follows:

```
switch# show system internal access-list feature bank-chain map vlan-vlan egress module 2

Feature                 Rslt Type      T0B0      T0B1      T1B0      T1B1

QoS                         Qos          X         X
RACL                        Acl                              X         X
VACL                        Acl                              X         X
Tunnel Decap                Acl          X         X
Netflow                     Acl                              X         X
Netflow Sampler             Acc          X         X
Rbacl                       Acl          X         X
CTS implicit Tunnel         Acl          X         X
SPM WCCP                    Acl                              X         X
SPM OTV                     Acl          X         X
SPM LISP                    Acl          X         X

SPM ERSPAN (termination)    Acl          X         X
OTV25 DECAP                 Acl          X         X
SPM NVE                     Acl                              X         X
SPM NVE RDT                 Acl                              X         X
SPM ITD                     Acl                              X         X
```

Check whether features with different result types overlap under the same TCAM. In this scenario, the QoS and RBACL features have different result types and are displayed under the same TCAM: T0B0 and T0B1. Features that are displayed under the same TCAM bank, but have different result types, cannot be configured together.

14.16 Additional References for IP ACLs

Related Documents

Related Topic	Document Title
IP ACL commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Inspur CN12700 Series INOS Security Command Reference*
Object group commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Inspur CN12700 Series INOS Security Command Reference*
Time range commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Inspur CN12700 Series INOS Security Command Reference*
SNMP	*Inspur CN12700 Series INOS System Management Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

14.17 Feature History for IP ACLs

This table lists the release history for this feature.

Table 33： Feature History for IP ACLs

Feature Name	Releases	Feature Information
Router ACL on Bridge domain interfaces	8.2(3)	Router ACL is now supported on Bridge domain interfaces.
Flexible ACL TCAM Bank Chaining	8.2(3)	Added the support for Inspur CN series modules for the flexible ACL TCAM bank chaining feature.
Configuring ACLs over F3 modules	8.2(3)	Support for F3 modules is introduced.
Flexible ACL TCAM Bank Chaining	8.2(3)	Added the support for the flexible ACL TCAM bank chaining feature.
ACL TCAM bank mapping	8.2(3)	Added a command to display the bank-mapping matrix.
IP ACLs	8.2(3)	Added support for ACL TCAM bank mapping.
IP ACLs	8.2(3)	Updated for F3modules.
IP ACLs	8.2(3)	Updated for F3Series modules.
FCoE ACLs	8.2(3)	Added support for FCoE ACLs on F3Series modules.
IP ACLs	8.2(3)	Added support for ACL capture on F3 modules.
IP ACLs	8.2(3)	Changed the **show running-config aclmgr** and **show startup-config aclmgr** commands to display only the user-configured ACLs (and not also the default CoPP-configured ACLs) in the running and startup configurations.
VTY ACLs	8.2(3)	Added support to control access to traffic received over a VTY line.
IP ACLs	8.2(3)	Added support for up to 128K ACL entries when using an XL line card, provided a scalable services license is installed.
ACL logging	8.2(3)	Added support for logging of packets sent to the supervisor module for ACL processing.
IP ACLs	8.2(3)	Added support for MAC packet classification on Layer 2 interfaces.

CHAPTER 15 Configuring MAC ACLs

This chapter describes how to configure MAC access lists (ACLs) on Inspur INOS devices. This chapter contains the following sections:

- Finding Feature Information
- Information About MAC ACLs
- Licensing Requirements for MAC ACLs
- Prerequisites for MAC ACLs
- Guidelines and Limitations for MAC ACLs
- Default Settings for MAC ACLs
- Configuring MAC ACLs
- Verifying the MAC ACL Configuration
- Monitoring and Clearing MAC ACL Statistics
- Configuration Example for MAC ACLs
- Additional References for MAC ACLs
- Feature History for MAC ACLs

15.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

15.2 Information About MAC ACLs

MAC ACLs are ACLs that use information in the Layer 2 header of packets to filter traffic. MAC ACLs share many fundamental concepts with IP ACLs, including support for virtualization.

Related Topics

Information About ACLs.

15.2.1 MAC Packet Classification

MAC packet classification allows you to control whether a MAC ACL that is on a Layer 2 interface applies to all traffic entering the interface, including IP traffic, or to non-IP traffic only.

MAC packet classification does not work on the Layer 3 control plane protocols such as HSRP, VRRP, OSPF, and so on. If you enable MAC packet classification on the VLANs, the basic functionalities will break on these protocols.

MAC Packet Classification State	Effect on Interface
Enabled	• A MAC ACL that is on the interface applies to all traffic entering the interface, including IP traffic. • You cannot apply an IP port ACL on the interface.
Disabled	• A MAC ACL that is on the interface applies only to non-IP traffic entering the interface. • You can apply an IP port ACL on the interface.

Related Topics

Enabling or Disabling MAC Packet Classification.

15.3 Licensing Requirements for MAC ACLs

This table shows the licensing requirements for this feature.

Product	License Requirement
Inspur INOS	MAC ACLs require no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

15.4 Prerequisites for MAC ACLs

There are no prerequisites for configuring MAC ACLs.

15.5 Guidelines and Limitations for MAC ACLs

MAC ACLs have the following configuration guidelines and limitations:
- MAC ACLs apply to ingress traffic only.
- ACL statistics are not supported if the DHCP snooping feature is enabled.

15.6 Default Settings for MAC ACLs

This table lists the default settings for MAC ACL parameters.

Table 34： Default MAC ACLs Parameters

Parameters	Default
MAC ACLs	No MAC ACLs exist by default
ACL rules	Implicit rules apply to all ACLs

15.7 Configuring MAC ACLs

15.7.1 Creating a MAC ACL

You can create a MAC ACL and add rules to it.

SUMMARY STEPS

1. **configure terminal**
2. **mac access-list** *name*
3. {**permit** | **deny**} *source destination protocol*
4. (Optional) **statistics per-entry**
5. (Optional) **show mac access-lists** *name*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**mac access-list** *name*
	Example:
	`switch(config)# mac access-list acl-mac-01`
	`switch(config-mac-acl)#`
		Creates the MAC ACL and enters ACL configuration mode.

Step 3	{**permit** | **deny**} *source destination protocol* **Example:** `switch(config-mac-acl)# permit 00c0.4f00.0000 0000.00ff.ffff any`	Creates a rule in the MAC ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Inspur CN12700 Series INOS Security Command Reference*.
Step 4	(Optional) **statistics per-entry** **Example:** `switch(config-mac-acl)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	(Optional) **show mac access-lists** *name* **Example:** `switch(config-mac-acl)# show mac access-lists acl-mac-01`	Displays the MAC ACL configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-mac-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

15.7.2 Changing a MAC ACL

You can remove a MAC ACL from the device.

Before you begin

Use the **show mac access-lists** command with the summary keyword to find the interfaces that a MAC ACL is configured on.

SUMMARY STEPS

1. **configure terminal**
2. **mac access-list** *name*
3. (Optional) [*sequence-number*] {**permit** | **deny**} *source destination protocol*
4. (Optional) **no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*}
5. (Optional) [**no**] **statistics per-entry**
6. (Optional) **show mac access-lists** *name*
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**mac access-list** *name*
	Example:
	`switch(config)# mac access-list acl-mac-01`
	`switch(config-mac-acl)#`
		Enters ACL configuration mode for the ACL that you specify by name.

Step 3	(Optional) [*sequence-number*] {**permit** | **deny**} *source destination protocol* **Example:** `switch(config-mac-acl)# 100 permit mac 00c0.4f00.00 0000.00ff.ffff any`	Creates a rule in the MAC ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Inspur CN12700 Series INOS Security Command Reference*.
Step 4	(Optional) **no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*} **Example:** `switch(config-mac-acl)# no 80`	Removes the rule that you specify from the MAC ACL. The **permit** and **deny** commands support many ways of identifying traffic. For more information, see the *Inspur CN12700 Series INOS Security Command Reference*.
Step 5	(Optional) [**no**] **statistics per-entry** **Example:** `switch(config-mac-acl)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 6	(Optional) **show mac access-lists** *name* **Example:** `switch(config-mac-acl)# show mac access-lists acl-mac-01`	Displays the MAC ACL configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-mac-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

15.7.3 Changing Sequence Numbers in a MAC ACL

You can change all the sequence numbers assigned to rules in a MAC ACL. Resequencing is useful when you need to insert rules into an ACL and there are not enough available sequence numbers.

SUMMARY STEPS

1. **configure terminal**
2. **resequence mac access-list** *name starting-sequence-number increment*

3. (Optional) **show mac access-lists** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**resequence mac access-list** *name starting-sequence-number increment* **Example:** `switch(config)# resequence mac access-list acl-mac-01 100 10`	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the number specified by the starting-sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment number that you specify.
Step 3	(Optional) **show mac access-lists** *name* **Example:** `switch(config)# show mac access-lists acl-mac-01`	Displays the MAC ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

15.7.4 Removing a MAC ACL

You can remove a MAC ACL from the device.

SUMMARY STEPS

1. **configure terminal**
2. **no mac access-list** *name*
3. (Optional) **show mac access-lists** *name* **summary**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no mac access-list** *name*
	Example:
	`switch(config)# no mac access-list acl-mac-01`
	`switch(config)#`
		Removes the MAC ACL that you specify by name from the running configuration.

	Command or Action	Purpose
Step 3	(Optional) **show mac access-lists** *name* **summary** **Example:** `switch(config)# show mac access-lists acl-mac-01 summary`	Displays the MAC ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

15.7.5 Applying a MAC ACL as a Port ACL

You can apply a MAC ACL as a port ACL to any of the following interface types:

- Layer 2 or Layer 3 Ethernet interfaces
- Layer 2 or Layer 3 port-channel interfaces

Before you begin

Ensure that the ACL that you want to apply exists and is configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *slot*/*port*
   - **interface port-channel** *channel-number*
3. **mac port access-group** *access-list*
4. (Optional) **show running-config aclmgr**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
	Example:
	`switch(config)# interface port-channel 5`
	`switch(config-if)#`
		• Enters interface configuration mode for a Layer 2 or Layer 3 interface.
		• Enters interface configuration mode for a Layer 2 or Layer 3 port-channel interface.

	Command or Action	Purpose
Step 3	**mac port access-group** *access-list* **Example:** `switch(config-if)# mac port access-group acl-01`	Applies a MAC ACL to the interface.
Step 4	(Optional) **show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	Displays ACL configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

15.7.6 Applying a MAC ACL as a VACL

You can apply a MAC ACL as a VACL.

Related Topics

Configuring VACLs.

15.7.7 Enabling or Disabling MAC Packet Classification

You can enable or disable MAC packet classification on a Layer 2 interface.

Before you begin

The interface must be configured as a Layer 2 interface. when the MAC packet classification feature is enabled on the interface.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *slot*/*port*
   - **interface port-channel** *channel-number*
3. **[no] mac packet-classify**
4. (Optional) Enter one of the following commands:
   - show running-config interface ethernet *slot*/*port*
   - **show running-config interface port-channel** *channel-number*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot/port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
	Example:
	`switch(config)# interface port-channel 5`
	`switch(config-if)#`
		• Enters interface configuration mode for a Ethernet interface.
		• Enters interface configuration mode for a port-channel interface.

Step 3	[**no**] **mac packet-classify** **Example:** `switch(config-if)# mac packet-classify`	Enables MAC packet classification on the interface. The **no** option disables MAC packet classification on the interface.

Step 4	(Optional) Enter one of the following commands:
	• **show running-config interface ethernet** *slot/port*
	• **show running-config interface port-channel** *channel-number*
	Example:
	`switch(config-if)# show running-config interface ethernet 2/1`
	Example:
	`switch(config-if)# show running-config interface port-channel 5`
		• Displays the running configuration of the Ethernet interface.
		• Displays the running configuration of the port-channel interface.

Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

MAC Packet Classification.

15.8 Verifying the MAC ACL Configuration

To display MAC ACL configuration information, perform one of the following tasks. For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration.
show running-config aclmgr [**all**]	Displays the ACL configuration, including MAC ACLs and the interfaces to which MAC ACLs are applied.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration.

15.9 Monitoring and Clearing MAC ACL Statistics

Use the **show mac access-lists** command to monitor statistics about a MAC ACL, including the number of packets that have matched each rule.

To monitor or clear MAC ACL statistics, use one of the commands in this table. For detailed information about these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration. If the MAC ACL includes the statistics per-entry command, the **show mac access-lists** command output includes the number of packets that have matched each rule.
clear mac access-list counters	Clears statistics for all MAC ACLs or for a specific MAC ACL.

15.10 Configuration Example for MAC ACLs

The following example shows how to create a MAC ACL named acl-mac-01 and apply it to Ethernet interface 2/1, which is a Layer 2 interface in this example:

```
mac access-list acl-mac-01
  permit 00c0.4f00.0000 0000.00ff.ffff any
interface ethernet 2/1
  mac port access-group acl-mac-01
```

15.11 Additional References for MAC ACLs

Related Documents

Related Topic	Document Title
MAC ACL commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Inspur CN12700 Series INOS Security Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

15.12 Feature History for MAC ACLs

This table lists the release history for this feature.

Table 35： Feature History for MAC ACLs

Feature Name	Releases	Feature Information
MAC ACLs	8.2(3)	Updated for Series modules.
MAC ACLs	8.2(3)	Updated for F3Series modules.
MAC ACLs	8.2(3)	Changed the **show running-config aclmgr** and **show startup-config aclmgr**

		commands to display only the user-configured ACLs (and not also the default CoPP-configured ACLs) in the running and startup configurations.
MAC ACLs	8.2(3)	Support was added for up to 128,000 ACL entries when using an XL line card, provided a scalable services license is installed.
MAC ACLs	8.2(3)	Support was added for MAC packet classification.

CHAPTER 16 Configuring VLAN ACLs

This chapter describes how to configure VLAN access lists (ACLs) on Inspur INOS devices. This chapter includes the following sections:

- Finding Feature Information
- Information About VLAN ACLs
- Licensing Requirements for VACLs
- Prerequisites for VACLs
- Guidelines and Limitations for VACLs
- Default Settings for VACLs
- Configuring VACLs, on page 516
- Verifying the VACL Configuration
- Monitoring and Clearing VACL Statistics
- Configuration Example for VACLs
- Additional References for VACLs
- Feature History for VLAN ACLs

16.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

16.2 Information About VLAN ACLs

A VLAN ACL (VACL) is one application of an IP ACL or a MAC ACL. You can configure VACLs to apply to all packets that are routed into or out of a VLAN or are bridged within a VLAN. VACLs are strictly for security packet filtering and for redirecting traffic to specific physical interfaces. VACLs are not defined by direction (ingress or egress).

Related Topics

Information About ACLs.

16.2.1 VLAN Access Maps and Entries

VACLs use access maps to contain an ordered list of one or more map entries. Each map entry associates IP or MAC ACLs to an action. Each entry has a sequence number, which allows you to control the precedence of entries.

When the device applies a VACL to a packet, it applies the action that is configured in the first access map entry that contains an ACL that permits the packet.

16.2.2 VACLs and Actions

In access map configuration mode, you use the **action** command to specify one of the following actions:

Forward

Sends the traffic to the destination determined by the normal operation of the switch.

Redirect

Redirects the traffic to one or more specified interfaces.

Drop

Drops the traffic. If you specify drop as the action, you can also specify that the device logs the dropped packets.

16.2.3 VACL Statistics

The device can maintain global statistics for each rule in a VACL. If a VACL is applied to multiple VLANs, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that VACL is applied.

For each VLAN access map that you configure, you can specify whether the device maintains statistics for that VACL. This feature allows you to turn VACL statistics on or off as needed to monitor traffic filtered by a VACL or to help troubleshoot VLAN access-map configuration.

Related Topics

Monitoring and Clearing VACL Statistics.

16.2.4 Session Manager Support for VACLs

Session Manager supports the configuration of VACLs. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. For more information about Session Manager, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.

16.2.5 Virtualization Support for VACLs

The following information applies to VACLs used in virtual device contexts (VDCs):

- ACLs are unique per VDC. You cannot use an ACL that you created in one VDC in a different VDC.
- Because ACLs are not shared by VDCs, you can reuse ACL names in different VDCs.
- The device does not limit ACLs or rules on a per-VDC basis.

16.3 Licensing Requirements for VACLs

This table shows the licensing requirements for this feature.

Product	License Requirement
Inspur INOS	VACLs require no license. However to support up to 128,000 ACL entries using an XL line card, you must install the scalable services license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

16.4 Prerequisites for VACLs

VACLs have the following prerequisite:

- Ensure that the IP ACL or MAC ACL that you want to use in the VACL exists and is configured to filter traffic in the manner that you need for this application.

16.5 Guidelines and Limitations for VACLs

VACLs have the following configuration guidelines:

- We recommend that you perform ACL configurations using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources required by the configuration are available prior to committing them to the running configuration. For more information about Session Manager, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.

- ACL statistics are not supported if the DHCP snooping feature is enabled. However, ACL statistics are supported on F3 and F3 Series modules if the DHCP snooping feature is enabled.
- Each of the 16 forwarding engines in an F3Series module supports up to 250 IPv6 addresses across multiple ACLs.
- Each of the 12 forwarding engines in an F3Series module has 16,000 total TCAM entries, equally split across two banks. 168 default entries are reserved. Each forwarding engine also has 512 IPv6 compression TCAM entries.
- Each of the 12 forwarding engines in an F3 Series module has 16,000 total TCAM entries, equally split across four TCAM banks, that is, T0B0, T0B1, T1B0, and T1B1.
- Each of the 6 forwarding engines in an F3 Series module has 128,000 total TCAM entries, equally split across four TCAM banks, that is, T0B0, T0B1, T1B0, and T1B1.
- VACL redirects to SPAN destination ports are not supported.
- Statistics for deny ACE support are supported only for the terminating sequence for the following sequence-based features: VACL, policy-based routing (PBR), and quality of service (QoS). This guideline is also applicable for F3 Series module.

16.6 Default Settings forVACLs

This table lists the default settings for VACL parameters.

Table 36：Default VACL Parameters

Parameters	**Default**
VACLs	No IP ACLs exist by default
ACL rules	Implicit rules apply to all ACLs
Deny ACE support	Disabled

16.7 Configuring VACLs

16.7.1 Creating a VACL or Adding a VACL Entry

You can create a VACL or add entries to an existing VACL. In both cases, you create a VACL entry, which is a VLAN access-map entry that associates one or more ACLs with an action to be applied to the matching traffic.

Before you begin

Ensure that the ACLs that you want to use in the VACL exists and are configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. **configure terminal**
2. **vlan access-map** *map-name* [*sequence-number*]
3. Enter one of the following commands:
   - **match** {**ip** | **ipv6**} **address** *ip-access-list*
   - **match mac address** *mac-access-list*
4. **action {drop | forward | redirect}**
5. (Optional) [**no**] **statistics per-entry**

6. (Optional) **show running-config aclmgr**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**vlan access-map** *map-name* [*sequence-number*]
	Example:
	`switch(config)# vlan access-map acl-mac-map`
	`switch(config-access-map)#`
		Enters VLAN access-map configuration mode for the VLAN access map specified. If the VLAN access map does not exist, the device creates it.
		If you do not specify a sequence number, the device creates a new entry whose sequence number is 10 greater than the last sequence number in the access map.

Step 3	Enter one of the following commands:
	• **match** {**ip** | **ipv6**} **address** *ip-access-list*
	• **match mac address** *mac-access-list*
	Example:
	`switch(config-access-map)# match mac address acl-ip-lab`
	Example:
	`switch(config-access-map)# match mac address acl-mac-01`
		Specifies an ACL for the access-map entry.

Step 4	**action** {**drop** | **forward** | **redirect**} **Example:** `switch(config-access-map)# action forward`	Specifies the action that the device applies to traffic that matches the ACL. The **action** command supports many options. For more information, see the *Inspur CN12700 Series INOS Security Command Reference*.
Step 5	(Optional) [**no**] **statistics per-entry** **Example:** `switch(config-access-map)# statistics per-entry`	Specifies that the device maintains global statistics for packets that match the rules in the VACL. The **no** option stops the device from maintaining global statistics for the VACL.
Step 6	(Optional) **show running-config aclmgr** **Example:** `switch(config-access-map)# show running-config aclmgr`	Displays the ACL configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-access-map)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

16.7.2 Removing a VACL or a VACL Entry

You can remove a VACL, which means that you will delete the VLAN access map. You can also remove a single VLAN access-map entry from a VACL.

Before you begin

Ensure that you know whether the VACL is applied to a VLAN. The device allows you to remove VACLs that are currently applied. Removing a VACL does not affect the configuration of VLANs where you have applied the VACL. Instead, the device considers the removed VACL to be empty.

SUMMARY STEPS

1. **configure terminal**
2. **no vlan access-map** *map-name* [*sequence-number*]
3. (Optional) **show running-config aclmgr**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no vlan access-map** *map-name* [*sequence-number*] **Example:** `switch(config)# no vlan access-map acl-mac-map 10`	Removes the VLAN access map configuration for the specified access map. If you specify the *sequence-number* argument and the VACL contains more than one entry, the command removes only the entry specified.
Step 3	(Optional) **show running-config aclmgr** **Example:** `switch(config)# show running-config aclmgr`	Displays the ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

16.7.3 Applying a VACL to a VLAN

You can apply a VACL to a VLAN.

Before you begin

If you are applying a VACL, ensure that the VACL exists and is configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **vlan filter** *map-name* **vlan-list** *list*
3. (Optional) **show running-config aclmgr**

4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **vlan filter** *map-name* **vlan-list** *list*
	Example:
	`switch(config)# vlan filter acl-mac-map vlan-list 1-20,26-30`
	`switch(config)#`
		Applies the VACL to the VLANs by the list that you specified. The **no** option unapplies the VACL.

Step 3	(Optional) **show running-config aclmgr** **Example:** `switch(config)# show running-config aclmgr`	Displays the ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

16.7.4 Configuring Deny ACE Support

You can configure the device to support deny access control entries (ACEs) in a sequence for the following sequence-based features: VACL, policy-based routing (PBR), and QoS. When deny ACEs are enabled, the traffic that matches a **deny** ACE (an ACL rule with the deny keyword) in a class-map-acl is recursively matched against subsequent class-map-acls until it hits a permit ACE.

Before you begin

Ensure that you are in the default or admin VDC.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **hardware access-list allow deny ace**
3. (Optional) **show running-config aclmgr**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **hardware access-list allow deny ace** **Example:** `switch(config)# hardware access-list allow`	Enables support for deny ACEs in a sequence.

	`deny ace`	
Step 3	(Optional) **show running-config aclmgr** **Example:** `switch(config)# show running-config aclmgr`	Displays the ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

16.8 Verifying the VACL Configuration

To display VACL configuration information, perform one of the following tasks. For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show running-config aclmgr [**all**]	Displays the ACL configuration, including the VACL-related configuration.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration.
show vlan filter	Displays information about VACLs that are applied to a VLAN.
show vlan access-map	Displays information about VLAN access maps.

16.9 Monitoring and Clearing VACL Statistics

To monitor or clear VACL statistics, use one of the commands in this table. For detailed information about these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show vlan access-list	Displays the VACL configuration. If the VLAN access-map includes the **statistics per-entry** command, the **show vlan access-list** command output includes the number of packets that have matched each rule.
clear vlan access-list counters	Clears statistics for all VACLs or for a specific VACL.

16.10 Configuration Example for VACLs

The following example shows how to configure a VACL to forward traffic permitted by a MAC ACL named acl-mac-01 and how to apply the VACL to VLANs 50 through 82.

```
conf t
vlan access-map acl-mac-map
  match mac address acl-mac-01
  action forward
vlan filter acl-mac-map vlan-list 50-82
```

16.11 Additional References for VACLs

Related Documents

Related Topic	Document Title
VACL commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Inspur CN12700 Series INOS Security Command Reference*
Policy-based routing (PBR) configuration	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*
QoS configuration	*Inspur CN12700 Series INOS Quality of Service Configuration Guide*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

16.12 Feature History for VLAN ACLs

This table lists the release history for this feature.

Table 37 : Feature History for VLAN ACLs

Feature Name	Releases	Feature Information	
VLAN ACLs	8.2(3)	Added support for deny ACEs in a sequence.	
VLAN ACLs	8.2(3)	Updated for F3Series modules.	
VLAN access maps	8.2(3)	No change from Release 8.2(3).	

CHAPTER 17 Configuring Port Security

This chapter describes how to configure port security on Inspur INOS devices. This chapter includes the following sections:

- Finding Feature Information
- Information About Port Security
- Licensing Requirements for Port Security
- Prerequisites for Port Security
- Default Settings for Port Security
- Guidelines and Limitations for Port Security
- Configuring Port Security
- Verifying the Port Security Configuration
- Displaying Secure MAC Addresses
- Configuration Example for Port Security
- Feature History for Port Security

17.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

17.2 Information About Port Security

Port security allows you to configure Layer 2 physical interfaces and Layer 2 port-channel interfaces to allow inbound traffic from only a restricted set of MAC addresses. The MAC addresses in the restricted set are called secure MAC addresses. In addition, the device does not allow traffic from these MAC addresses on another interface within the same VLAN. The number of MAC addresses that the device can secure is configurable per interface.

17.2.1 Secure MAC Address Learning

The process of securing a MAC address is called learning. A MAC address can be a secure MAC address on one interface only. For each interface that you enable port security on, the device can learn a limited number of MAC addresses by the static, dynamic, or sticky methods. The way that the device stores secure MAC addresses varies depending upon how the device learned the secure MAC address.

Related Topics

Secure MAC Address Maximums.

Static Method

The static learning method allows you to manually add or remove secure MAC addresses to the running configuration of an interface. If you copy the running configuration to the startup configuration, static secure MAC addresses are unaffected if the device restarts.

A static secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- You explicitly remove the address from the configuration.
- You configure the interface to act as a Layer 3 interface.

Adding secure addresses by the static method is not affected by whether dynamic or sticky address learning is enabled.

Related Topics

Removing a Static Secure MAC Address on an Interface
Port Type Changes.

Dynamic Method

By default, when you enable port security on an interface, you enable the dynamic learning method. With this method, the device secures MAC addresses as ingress traffic passes through the interface. If the address is not yet secured and the device has not reached any applicable maximum, it secures the address and allows the traffic.

The device stores dynamic secure MAC addresses in memory. A dynamic secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- The device restarts.
- The interface restarts.
- The address reaches the age limit that you configured for the interface.
- You explicitly remove the address.
- You configure the interface to act as a Layer 3 interface.

Related Topics

Dynamic Address Aging.
Removing a Dynamic Secure MAC Address.

Sticky Method

If you enable the sticky method, the device secures MAC addresses in the same manner as dynamic address learning, but the device stores addresses learned by this method in nonvolatile RAM (NVRAM). As a result, addresses learned by the sticky method persist through a device restart. Sticky secure MAC addresses do not appear in the running configuration of an interface.

Dynamic and sticky address learning are mutually exclusive. When you enable sticky learning on an interface, the device stops dynamic learning and performs sticky learning instead. If you disable sticky learning, the device resumes dynamic learning.

A sticky secure MAC address entry remains in the configuration of an interface until one of the following events occurs:

- You explicitly remove the address.
- You configure the interface to act as a Layer 3 interface.

Related Topics

Removing a Sticky Secure MAC Address.

17.2.2 Dynamic Address Aging

The device ages MAC addresses learned by the dynamic method and drops them after the age limit is reached. You can configure the age limit on each interface. The range is from 1 to 1440 minutes. The default aging time is 0, which disables aging.

The method that the device uses to determine that the MAC address age is also configurable. The two methods of determining address age are as follows:

Inactivity

The length of time after the device last received a packet from the address on the applicable interface.

Absolute

The length of time after the device learned the address. This is the default aging method; however, the default aging time is 0 minutes, which disables aging.

17.2.3 Secure MAC Address Maximums

By default, an interface can have only one secure MAC address. You can configure the maximum number of MAC addresses permitted per interface or per VLAN on an interface. Maximums apply to secure MAC addresses learned by any method: dynamic, sticky, or static.

Tip	To ensure that an attached device has the full bandwidth of the port, set the maximum number of addresses to one and configure the MAC address of the attached device.

The following three limits can determine how many secure MAC addresses are permitted on an interface:

System maximum

The device has a nonconfigurable limit of 8192 secure MAC addresses. If learning a new address would violate the device maximum, the device does not permit the new address to be learned, even if the interface or VLAN maximum has not been reached.

Interface maximum

You can configure a maximum number of 1025 secure MAC addresses for each interface protected by port security. The default interface maximum is one address. Sum of all interface maximums on a switch cannot exceed the system maximum.

VLAN maximum

You can configure the maximum number of secure MAC addresses per VLAN for each interface protected by port security. The sum of all VLAN maximums under an interface cannot exceed the configured interface maximum. VLAN maximums are useful only for trunk ports. There are no default VLAN maximums.

You can configure VLAN and interface maximums per interface, as needed; however, when the new limit is less than the applicable number of secure addresses, you must reduce the number of secure MAC addresses first. Otherwise, the configuration of new limit is rejected.

Related Topics

Security Violations and Actions.
Removing a Dynamic Secure MAC Address.
Removing a Sticky Secure MAC Address.
Removing a Static Secure MAC Address on an Interface.

17.2.4 Security Violations and Actions

Port security triggers security violations when either of the two following events occur:

MAX Count Violation

Ingress traffic arrives at an interface from a nonsecure MAC address and learning the address would exceed the applicable maximum number of secure MAC addresses.

When an interface has both a VLAN maximum and an interface maximum configured, a violation occurs when either maximum is exceeded. For example, consider the following on a single interface configured with port security:

- VLAN 1 has a maximum of 5 addresses
- The interface has a maximum of 10 addresses

The interface has a maximum of 20 addresses

The device detects a violation when any of the following occurs:

- The device has learned five addresses for VLAN 1 and inbound traffic from a sixth address arrives at the interface in VLAN 1.
- The device has learned 10 addresses on the interface and inbound traffic from an 11th address arrives at the interface.

MAC Move Violation

Ingress traffic from a secure MAC address arrives at a different secured interface in the same VLAN as the interface on which the address is secured.

When a security violation occurs, the device increments the security violation counter for the interface and takes the action specified by the port security configuration of the interface. If a violation occurs because ingress traffic from a secure MAC address arrives at a different interface than the interface on which the address is secure, the device applies the action on the interface that received the traffic.

The violation modes and the possible actions that a device can take are as follows:

Shutdown violation mode

Error disables the interface that received the packet triggering the violation and the port shuts down. The security violation count is set to 1. This action is the default. After you reenable the interface, it retains its port security configuration, including its static and sticky secure MAC addresses. However, the dynamic MAC addresses are not retained and have to relearned.

You can use the **errdisable recovery cause pscecure-violation** global configuration command to configure the device to reenable the interface automatically if a shutdown occurs, or you can manually reenable the interface by entering the **shutdown** and **no shut down** interface configuration commands. For detailed information about the commands, see the Security Command Reference for your platform.

Restrict violation mode

Drops ingress traffic from any nonsecure MAC addresses.

The device keeps a count of the number of unique source MAC addresses of dropped packets, which is called the security violation count.

Violation is triggered for each unique nonsecure source MAC address and security violation count increments till 10, which is the maximum value. The maximum value of 10 is fixed and not configurable.

Address learning continues until the maximum security violations (10 counts) have occurred on the interface. Traffic from addresses learned after the first security violation are added as BLOCKED entries in the MAC table and dropped. These BLOCKED MAC address age out after 5 minutes. The BLOCKED MAC address age out time of 5 minutes is fixed and not configurable.

Depending on the violation type, RESTRICT mode action varies as follows:

- In case of MAX count violation, after the maximum number of MAX count violations (10) is reached, the device stops learning new MAC addresses. Interface remains up.
- In case of MAC move violation, when the maximum security violations have occurred on the interface, the interface is error Disabled.

Protect violation mode

Prevents further violations from occurring. The address that triggered the security violation is learned but any traffic from the address is dropped. Security violation counter is set to 1, which is the maximum value. Further address learning stops. Interface remains up.

Note that the security violation is reset to 0 after the interface is recovered from violation through one of the following events:

- Dynamic secure MAC addresses age out
- Interface flap, link down, or link up events
- Port-security disable and re-enable on the interface
- Changing violation mode of the interface

17.2.5 Port Security and Port Types

You can configure port security only on Layer 2 interfaces. Details about port security and different types of interfaces or ports are as follows:

Access ports

You can configure port security on interfaces that you have configured as Layer 2 access ports. On an access port, port security applies only to the access VLAN. VLAN maximums are not useful for access ports.

Trunk ports

You can configure port security on interfaces that you have configured as Layer 2 trunk ports. The device allows VLAN maximums only for VLANs associated with the trunk port.

SPAN ports

You can configure port security on SPAN source ports but not on SPAN destination ports.

Ethernet port channels

You can configure port security on Layer 2 Ethernet port channels in either access mode or trunk mode.

Fabric Extender (FEX) ports

Port security is supported on GEM and FEX ports.

Private VLAN Enabled Ports

Port Security is supported on ports that are enabled as Private VLAN ports.

PVLAN Host (physical interfaces only)

You can configure Private VLANs (PVLANs) to provide traffic separation and security at the Layer 2 level. A PVLAN is one or more pairs of a primary VLAN and a secondary VLAN, all with the same primary VLAN.

PVLAN Promiscuous (physical interfaces only)

You can configure a Layer 2 VLAN network interface, or switched virtual interface (SVI), on the PVLAN promiscuous port, which provides routing functionality to the primary PVLAN. This is supported on physical interfaces only.

PVLAN trunk secondary/promiscuous

You can configure PVLAN trunk secondary/promiscuous in the of switchport mode. This is supported for both physical interface and portchannel.

17.2.6 Port Security and Port-Channel Interfaces

Port security is supported on Layer 2 port-channel interfaces. Port security operates on port-channel interfaces in the same manner as on physical interfaces, except as described in this section.

General guidelines

Port security on a port-channel interface operates in either access mode or trunk mode. In trunk mode, the MAC address restrictions enforced by port security apply to all member ports on a per-VLAN basis.

Enabling port security on a port-channel interface does not affect port-channel load balancing.

Port security does not apply to port-channel control traffic passing through the port-channel interface. Port security allows port-channel control packets to pass without causing security violations. Port-channel control traffic includes the following protocols:

- Port Aggregation Protocol (PAgP)
- Link Aggregation Control Protocol (LACP)
- Inter-Switch Link (ISL)
- IEEE 802.1Q

Configuring secure member ports

The port security configuration of a port-channel interface has no effect on the port security configuration of member ports.

Adding a member port

If you add a secure interface as a member port of a port-channel interface, the device discards all dynamic secure addresses learned on the member port but retains all other port-security configuration of the member port in the running configuration. Sticky and static secure MAC addresses learned on the secure member port are also stored in the running configuration rather than NVRAM.

If port security is enabled on the member port and not enabled on the port-channel interface, the device warns you when you attempt to add the member port to the port-channel interface. You can use the **force** keyword with the **channel-group** command to forcibly add a secure member port to a nonsecure port-channel interface.

While a port is a member of a port-channel interface, you cannot configure port security on the member port. To do so, you must first remove the member port from the port-channel interface.

Removing a member port

If you remove a member port from a port-channel interface, the device restores the port security configuration of the member port. Static and sticky secure MAC addresses that were learned on the port before you added it to the port-channel interface are restored to NVRAM and removed from the running configuration.

Removing a port-channel interface

If you remove a secure port-channel interface, the following occurs:

- The device discards all secure MAC addresses learned for the port-channel interface, including static and sticky secure MAC addresses learned on the port-channel interface.
- The device restores the port-security configuration of each member port. The static and sticky secure MAC addresses that were learned on member ports before you added them to the port-channel interface are restored to NVRAM and removed from the running configuration. If a member port did not have port security enabled prior to joining the port-channel interface, port security is not enabled on the member port after the port-channel interface is removed.

Disabling port security

If port security is enabled on any member port, the device does not allow you to disable port security on the port-channel interface. To do so, remove all secure member ports from the port-channel interface first. After disabling port security on a member port, you can add it to the port-channel interface again, as needed.

17.2.7 Port Type Changes

When you have configured port security on a Layer 2 interface and you change the port type of the interface, the device behaves as follows:

Access port to trunk port

When you change a Layer 2 interface from an access port to a trunk port, the device deletes all secure addresses learned by the dynamic method. The device moves the addresses learned by the static method to the native trunk VLAN. The sticky MAC addresses remain in same VLAN if the VLAN exists.

Otherwise, the MAC addresses move to the native VLAN of the trunk port.

Trunk port to access port

When you change a Layer 2 interface from a trunk port to an access port, the device drops all secure addresses learned by the dynamic method. It also moves all addresses learned by the sticky method on the native trunk VLAN to the access VLAN. The device drops secure addresses learned by the sticky method if they are not on the native trunk VLAN.

Switched port to routed port

When you change an interface from a Layer 2 interface to a Layer 3 interface, the device disables port security on the interface and discards all port security configuration for the interface. The device also discards all secure MAC addresses for the interface, regardless of the method used to learn the address.

Routed port to switched port

When you change an interface from a Layer 3 interface to a Layer 2 interface, the device has no port security configuration for the interface.

The static secure addresses that are configured per access or trunk VLAN on an interface are not retained during the following events:

- Changing global VLAN mode of the active VLANs on an interface between classical Ethernet and fabric path interfaces
- Changing switchport mode access or trunk to private VLAN or vice versa

802.1X and Port Security

You can configure port security and 802.1X on the same interfaces. Port security secures the MAC addresses that 802.1X authenticates. 802.1X processes packets before port security processes them, so when you enable both on an interface, 802.1X is already preventing inbound traffic on the interface from unknown MAC addresses.

When you enable 802.1X and port security on the same interface, port security continues to learn MAC addresses by the sticky or dynamic method, as configured. Additionally, depending on whether you enable 802.1X in single-host mode or multiple-host mode, one of the following occurs:

Single host mode

Port security learns the MAC address of the authenticated host.

Multiple host mode

Port security drops any MAC addresses learned for this interface by the dynamic method and learns the MAC address of the first host authenticated by 802.1X.

If a MAC address that 802.1X passes to port security would violate the applicable maximum number of secure MAC addresses, the device sends an authentication failure message to the host.

The device treats MAC addresses authenticated by 802.1X as though they were learned by the dynamic method, even if port security previously learned the address by the sticky or static methods. If you attempt to delete a secure MAC address that has been authenticated by 802.1X, the address remains secure.

If the MAC address of an authenticated host is secured by the sticky or static method, the device treats the address as if it were learned by the dynamic method, and you cannot delete the MAC address manually.

Port security integrates with 802.1X to reauthenticate hosts when the authenticated and secure MAC address of the host reaches its port security age limit. The device behaves differently depending upon the type of aging, as follows:

Absolute

Port security notifies 802.1X and the device attempts to reauthenticate the host. The result of reauthentication determines whether the address remains secure. If reauthentication succeeds, the device restarts the aging timer on the secure address; otherwise, the device drops the address from the list of secure addressees for the interface.

Inactivity

Port security drops the secure address from the list of secure addresses for the interface and notifies 802.1X. The device attempts to reauthenticate the host. If reauthentication succeeds, port security secures the address again.

17.2.8 Virtualization Support for Port Security

Port security supports VDCs as follows:

- Port security is local to each VDC. You enable and configure port security on a per-VDC basis.
- Each VDC maintains secure MAC addresses separately.
- The device cannot issue a security violation when a secured MAC address in one VDC is seen on a protected interface in another VDC.

17.3 Licensing Requirements for Port Security

The following table shows the licensing requirements for this feature:

Product	License Requirement
INOS	Port security requires no license. Any feature not included in a license package is bundled with the INOS ce images and is provided at no extra charge to you.

17.4 Prerequisites for Port Security

Port security has the following prerequisites:

- You must globally enable port security for the device that you want to protect with port security.

17.5 Default Settings for Port Security

This table lists the default settings for port security parameters.

Table 38： Default Port Security Parameters

Parameters	Default
Port security enablement globally	Disabled
Port security enablement per interface	Disabled
MAC address learning method	Dynamic
Interface maximum number of secure MAC addresses	1
Security violation action	Shutdown
Aging type	Absolute
Aging time	0

17.6 Guidelines and Limitations for Port Security

When configuring port security, follow these guidelines:

- Port security is supported on PVLAN ports.
- Port security does not support switched port analyzer (SPAN) destination ports.
- Port security does not depend upon other features.
- If any member link in a port-channel is in the pre-provisioned state, that is, the module is offline, then the port security feature cannot be disabled on the port-channel.
- Port security is not supported on vPC ports.
- Port security operates with 802.1X on Layer 2 Ethernet interfaces.

Related Topics

802.1X and Port Security.

17.7 Configuring Port Security

17.7.1 Enabling or Disabling Port Security Globally

You can enable or disable port security globally on a device. By default, port security is disabled globally.

When you disable port security, all port security configuration on the interface is ineffective. When you disable port security globally, all port security configuration is lost.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **feature port-security**
3. **show port-security**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature port-security** **Example:** `switch(config)# feature port-security`	Enables port security globally. The **no** option disables port security globally.
Step 3	**show port-security** **Example:** `switch(config)# show port-security`	Displays the status of port security.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

17.7.2 Enabling or Disabling Port Security on a Layer 2 Interface

You can enable or disable port security on a Layer 2 interface. By default, port security is disabled on all interfaces.

When you disable port security on an interface, all switchport port security configuration for the interface is lost.

You can enable port-security on a port-channel in the following ways:

- Bundle member links into a port-channel by using the **channel-group** command and then enable port-security on the port-channel.
- Create port-channel and configure port security. Configure port security on member links and then bundle member links by using the **channel-group** command. In case of pre-provisioned member links, you can bundle them to the port-channel after the module is online.

Before you begin

You must have enabled port security globally.

If a Layer 2 Ethernet interface is a member of a port-channel interface, you cannot enable or disable port security on the Layer 2 Ethernet interface.

If any member port of a secure Layer 2 port-channel interface has port security enabled, you cannot disable port security for the port-channel interface unless you first remove all secure member ports from the port-channel interface.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *slot*/*port*
   - **interface port-channel** *channel-number*
3. **switchport**
4. [**no**] **switchport port-security**
5. **show running-config port-security**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the Ethernet or port-channel interface that you want to configure with port security.

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures the interface as a Layer 2 interface.
Step 4	[**no**] **switchport port-security**	Enables port security on the interface. The **no** option disables port security on the interface.

	Command or Action	Purpose
	Example: `switch(config-if)# switchport port-security`	
Step 5	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Secure MAC Address Learning.
Enabling or Disabling Sticky MAC Address Learning.

17.7.3 Enabling or Disabling Sticky MAC Address Learning

You can disable or enable sticky MAC address learning on an interface. If you disable sticky learning, the device returns to dynamic MAC address learning on the interface, which is the default learning method.

By default, sticky MAC address learning is disabled.

Before you begin

You must have enabled port security globally.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *slot*/*port*
   - **interface port-channel** *channel-number*
3. **switchport**
4. [**no**] **switchport port-security mac-address sticky**
5. **show running-config port-security**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands: • **interface ethernet** *slot*/*port* • **interface port-channel** *channel-number*	Enters interface configuration mode for the interface that you want to configure with sticky MAC address learning.

	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`	

Step 3	**switchport** **Example:** `switch(config-if)# switchport`	Configures the interface as a Layer 2 interface.
Step 4	[**no**] **switchport port-security mac-address sticky** **Example:** `switch(config-if)# switchport port-security mac-address sticky`	Enables sticky MAC address learning on the interface. The **no** option disables sticky MAC address learning.
Step 5	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

17.7.4 Adding a Static Secure MAC Address on an Interface

You can add a static secure MAC address on a Layer 2 interface.

By default, no static secure MAC addresses are configured on an interface.

Before you begin

You must have enabled port security globally.

Verify that the interface maximum has not been reached for secure MAC addresses. If needed, you can remove a secure MAC address or you can change the maximum number of addresses on the interface.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *slot*/*port*
   - **interface port-channel** *channel-number*
3. [**no**] **switchport port-security mac-address** *address* [**vlan** *vlan-ID*]
4. **show running-config port-security**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:	Enters interface configuration mode for the interface that you specify.

	Command or Action	Purpose

	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`	

Step 3	[**no**] **switchport port-security mac-address** *address* [**vlan** *vlan-ID*] **Example:** `switch(config-if)# switchport port-security mac-address 0019.D2D0.00AE`	Configures a static MAC address for port security on the current interface. Use the **vlan** keyword if you want to specify the VLAN that traffic from the address is allowed on.
Step 4	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Verifying the Port Security Configuration
Configuring a Maximum Number of MAC Addresses
Removing a Dynamic Secure MAC Address.
Removing a Static Secure MAC Address on an Interface.

17.7.5 Removing a Static Secure MAC Address on an Interface

You can remove a static secure MAC address on a Layer 2 interface.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *slot*/*port*
   - **interface port-channel** *channel-number*
3. **no switchport port-security mac-address** *address*
4. **show running-config port-security**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface from which you want to remove a static secure MAC address.

Step 3	**no switchport port-security mac-address** *address* **Example:** `switch(config-if)# no switchport port-security mac-address 0019.D2D0.00AE`	Removes the static secure MAC address from port security on the current interface.
Step 4	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

17.7.6 Removing a Sticky Secure MAC Address

You can remove a sticky secure MAC addresses, which requires that you temporarily disable sticky address learning on the interface that has the address that you want to remove.

Before you begin

You must have enabled port security globally.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
- **interface ethernet** *slot*/*port*
- **interface port-channel** *channel-number*
3. **no switchport port-security mac-address sticky**
4. **clear port-security dynamic address** *address*
5. (Optional) **show port-security address interface** {**ethernet** *slot*/*port* | **port-channel** *channel-number*}
6. (Optional) **switchport port-security mac-address sticky**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands: • **interface ethernet** *slot*/*port*	Enters interface configuration mode for the interface from which you want to remove a sticky secure MAC address.

	Command or Action	Purpose

	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`	

Step 3	**no switchport port-security mac-address sticky** **Example:** `switch(config-if)# no switchport port-security mac-address sticky`	Disables sticky MAC address learning on the interface, which converts any sticky secure MAC addresses on the interface to dynamic secure MAC addresses.
Step 4	**clear port-security dynamic address** *address* **Example:** `switch(config-if)# clear port-security dynamic address 0019.D2D0.02GD`	Removes the dynamic secure MAC address that you specify.
Step 5	(Optional) **show port-security address interface** {**ethernet** *slot/port* | **port-channel** *channel-number*} **Example:** `switch(config)# show port-security address`	Displays secure MAC addresses. The address that you removed should not appear.
Step 6	(Optional) **switchport port-security mac-address sticky** **Example:** `switch(config-if)# switchport port-security mac-address sticky`	Enables sticky MAC address learning again on the interface.

17.7.7 Removing a Dynamic Secure MAC Address

You can remove dynamically learned, secure MAC addresses.

Before you begin

You must have enabled port security globally.

SUMMARY STEPS

1. **configure terminal**
2. **clear port-security dynamic** {**interface ethernet** *slot/port* | **address** *address*} [**vlan** *vlan-ID*]
3. **show port-security address**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**clear port-security dynamic** {**interface ethernet** *slot/port* | **address** *address*} [**vlan** *vlan-ID*]	Removes dynamically learned, secure MAC addresses, as specified.

	Command or Action	Purpose
	Example: `switch(config)# clear port-security dynamic interface ethernet 2/1`	If you use the **interface** keyword, you remove all dynamically learned addresses on the interface that you specify. If you use the **address** keyword, you remove the single, dynamically learned address that you specify. Use the **vlan** keyword if you want to further limit the command to removing an address or addresses on a particular VLAN.
Step 3	**show port-security address** **Example:** `switch(config)# show port-security address`	Displays secure MAC addresses.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

17.7.8 Configuring a Maximum Number of MAC Addresses

You can configure the maximum number of MAC addresses that can be learned or statically configured on a Layer 2 interface. You can also configure a maximum number of MAC addresses per VLAN on a Layer 2 interface. The largest maximum number of addresses that you can configure on an interface is 1025 addresses. The system maximum number of address is 8192.

By default, an interface has a maximum of one secure MAC address. VLANs have no default maximum number of secure MAC addresses.

Before you begin

You must have enabled port security globally.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *slot*/*port*
   - **interface port-channel** *channel-number*
3. [**no**] **switchport port-security maximum** *number* [**vlan** *vlan-ID*]
4. **show running-config port-security**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot/port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode, where *slot* is the interface that you want to configure with the maximum number of MAC addresses.

Step 3	[**no**] **switchport port-security maximum** *number* [**vlan** *vlan-ID*] **Example:** `switch(config-if)# switchport port-security maximum 425`	Configures the maximum number of MAC addresses that can be learned or statically configured for the current interface. The highest valid *number* is 1025. The **no** option resets the maximum number of MAC addresses to the default, which is 1. If you want to specify the VLAN that the maximum applies to, use the **vlan** keyword.
Step 4	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Removing a Dynamic Secure MAC Address.
Removing a Static Secure MAC Address on an Interface.

17.7.9 Configuring an Address Aging Type and Time

You can configure the MAC address aging type and the length of time that the device uses to determine when MAC addresses learned by the dynamic method have reached their age limit.

Absolute aging is the default aging type.

By default, the aging time is 0 minutes, which disables aging.

Before you begin

You must have enabled port security globally.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
   - **interface ethernet** *slot/port*
   - **interface port-channel** *channel-number*
3. **[no] switchport port-security aging type {absolute | inactivity}**
4. [**no**] **switchport port-security aging time** *minutes*
5. **show running-config port-security**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you want to configure with the MAC aging type and time.

Step 3	[**no**] **switchport port-security aging type** {**absolute** | **inactivity**} **Example:** `switch(config-if)# switchport port-security aging type inactivity`	Configures the type of aging that the device applies to dynamically learned MAC addresses. The **no** option resets the aging type to the default, which is absolute aging. **Note** F3series modules do not support the **inactivity** aging type.
Step 4	[**no**] **switchport port-security aging time** *minutes* **Example:** `switch(config-if)# switchport port-security aging time 120`	Configures the number of minutes that a dynamically learned MAC address must age before the device drops the address. The maximum valid *minutes* is 1440. The **no** option resets the aging time to the default, which is 0 minutes (no aging).
Step 5	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

17.7.10 Configuring a Security Violation Action

You can configure the action that the device takes if a security violation occurs. The violation action is configurable on each interface that you enable with port security.

The default security action is to shut down the port on which the security violation occurs.

Before you begin

You must have enabled port security globally.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:

• **interface ethernet** *slot*/*port*

- **interface port-channel** *channel-number*

3. **[no] switchport port-security violation {protect | restrict | shutdown}**
4. **show running-config port-security**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode for the interface that you want to configure with a security violation action.

Step 3	[**no**] **switchport port-security violation** {**protect** | **restrict** | **shutdown**} **Example:** `switch(config-if)# switchport port-security violation restrict`	Configures the security violation action for port security on the current interface. The **no** option resets the violation action to the default, which is to shut down the interface.
Step 4	**show running-config port-security** **Example:** `switch(config-if)# show running-config port-security`	Displays the port security configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

17.8 Verifying the Port Security Configuration

To display the port security configuration information, perform one of the following tasks. For detailed information about the fields in the output from this command, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show running-config port-security	Displays the port security configuration.
show port-security	Displays the port security status of the device.
show port-security interface	Displays the port security status of a specific interface.
show port-security address	Displays secure MAC addresses.

17.9 Displaying Secure MAC Addresses

Use the **show port-security address** command to display secure MAC addresses. For detailed information about the fields in the output from this command, see the *Inspur CN12700 Series INOS Security Command Reference*

17.10 Configuration Example for Port Security

The following example shows a port security configuration for the Ethernet 2/1 interface with VLAN and interface maximums for secure addresses. In this example, the interface is a trunk port. Additionally, the violation action is set to Restrict.

```
feature port-security
interface Ethernet 2/1
  switchport
  switchport  port-security
  switchport port-security maximum 10
  switchport port-security maximum 7 vlan 10
  switchport port-security maximum 3 vlan 20
  switchport port-security violation restrict
```

17.11 Feature History for Port Security

This table lists the release history for this feature.

Table 39： Feature History for Port Security

Feature Name	Releases	Feature Information	
Port security	8.2(3)	No change from Release 8.2(3).	
Port security	8.2(3)	Support for Layer 2 port-channel interfaces was added.	

CHAPTER 18 Configuring DHCP

This chapter describes how to configure the Dynamic Host Configuration Protocol (DHCP) on a Inspur INOS device.

This chapter includes the following sections:

- Finding Feature Information
- Information About DHCP Snooping
- Information About the DHCP Relay Agent
- Information About the DHCPv6 Relay Agent
- Information About DHCP Response Redirect
- Virtualization Support for DHCP
- Licensing Requirements for DHCP
- Prerequisites for DHCP
- Guidelines and Limitations for DHCP
- Default Settings for DHCP
- Configuring DHCP
- Configuring DHCPv6
- Configuring DHCP Response Redirect
- Verifying the DHCP Configuration
- Displaying DHCP Bindings
- Clearing the DHCP Snooping Binding Database
- Clearing DHCP Relay Statistics
- Clearing DHCPv6 Relay Statistics
- Monitoring DHCP
- Additional References for DHCP
- Feature History for DHCP

18.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

18.2 Information About DHCP Snooping

DHCP snooping acts like a firewall between untrusted hosts and trusted DHCP servers. DHCP snooping performs the following activities:

- Validates DHCP messages received from untrusted sources and filters out invalid messages.
- Builds and maintains the DHCP snooping binding database, which contains information about untrusted hosts with leased IP addresses.
- Uses the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

DHCP snooping can be enabled globally and on a per-VLAN basis. By default, the feature is disabled globally and on all VLANs. You can enable the feature on a single VLAN or a range of VLANs.

18.2.1 Trusted and Untrusted Sources

You can configure whether DHCP snooping trusts traffic sources. An untrusted source may initiate traffic attacks or other hostile actions. To prevent such attacks, DHCP snooping filters messages from untrusted sources.

In an enterprise network, a trusted source is a device that is under your administrative control. These devices include the switches, routers, and servers in the network. Any device beyond the firewall or outside the network is an untrusted source. Generally, host ports are treated as untrusted sources.

In a service provider environment, any device that is not in the service provider network is an untrusted source (such as a customer switch). Host ports are untrusted sources.

In the Inspur INOS device, you indicate that a source is trusted by configuring the trust state of its connecting interface.

The default trust state of all interfaces is untrusted. You must configure DHCP server interfaces as trusted. You can also configure other interfaces as trusted if they connect to devices (such as switches or routers) inside your network. You usually do not configure host port interfaces as trusted.

18.2.2 DHCP Snooping Binding Database

Using information extracted from intercepted DHCP messages, DHCP snooping dynamically builds and maintains a database. The database contains an entry for each untrusted host with a leased IP address if the host is associated with a VLAN that has DHCP snooping enabled. The database does not contain entries for hosts connected through trusted interfaces.

DHCP snooping updates the database when the device receives specific DHCP messages. For example, the feature adds an entry to the database when the device receives a DHCPACK message from the server. The feature removes the entry in the database when the IP address lease expires or the device receives a DHCPRELEASE message from the host.

Each entry in the DHCP snooping binding database includes the MAC address of the host, the leased IP address, the lease time, the binding type, and the VLAN number and interface information associated with the host.

Dynamic ARP inspection (DAI) and IP Source Guard also use information stored in the DHCP snooping binding database.

You can remove entries from the binding database by using the **clear ip dhcp snooping binding** command.

Related Topics

Clearing the DHCP Snooping Binding Database.

18.2.3 Packet Validation

The device validates DHCP packets received on the untrusted interfaces of VLANs that have DHCP snooping enabled. The device forwards the DHCP packet unless any of the following conditions occur (in which case, the packet is dropped):

- The device receives a DHCP response packet (such as a DHCPACK, DHCPNAK, or DHCPOFFER packet) on an untrusted interface.
- The device receives a packet on an untrusted interface, and the source MAC address and the DHCP client hardware address do not match. This check is performed only if the DHCP snooping MAC address verification option is turned on.
- The device receives a DHCPRELEASE or DHCPDECLINE message from an untrusted host with an entry in the DHCP snooping binding table, and the interface information in the binding table does not match the interface on which the message was received.

In addition, you can enable strict validation of DHCP packets, which checks the options field of DHCP packets, including the “magic cookie” value in the first four bytes of the options field. By default, strict validation is disabled. When you enable it, by using the **ip dhcp packet strict-validation** command, if DHCP snooping processes a packet that has an invalid options field, it drops the packet.

18.2.4 DHCP Snooping Option 82 Data Insertion

DHCP can centrally manage the IP address assignments for a large number of subscribers. When you enable

Option 82, the device identifies a subscriber device that connects to the network (in addition to its MAC address). Multiple hosts on the subscriber LAN can connect to the same port on the access device and are uniquely identified.

When you enable Option 82 on the Inspur INOS device, the following sequence of events occurs:

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.
2. When the Inspur INOS device receives the DHCP request, it adds the Option 82 information in the packet. The Option 82 information contains the device MAC address (the remote ID suboption) and the port identifier, vlan-mod-port, from which the packet is received (the circuit ID suboption). For hosts behind the port channel, the circuit ID is filled with the if_index of the port channel.
3. The device forwards the DHCP request that includes the Option 82 field to the DHCP server.
4. The DHCP server receives the packet. If the server is Option 82 capable, it can use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server echoes the Option 82 field in the DHCP reply.
5. The DHCP server sends the reply to the Inspur INOS device. The Inspur INOS device verifies that it originally inserted the Option 82 data by inspecting the remote ID and possibly the circuit ID fields. The Inspur INOS device removes the Option 82 field and forwards the packet to the interface that connects to the DHCP client that sent the DHCP request.

If the previously described sequence of events occurs, the following values do not change:

- Circuit ID suboption fields
- Suboption type
- Length of the suboption type
- Circuit ID type
- Length of the circuit ID type
- Remote ID suboption fields
- Suboption type
- Length of the suboption type
- Remote ID type
- Length of the circuit ID type

Figure 30：Suboption Packet Formats

This figure shows the packet formats for the remote ID suboption and the circuit ID suboption. The Inspur INOS device uses the packet formats when you globally enable DHCP snooping and when you enable Option 82 data insertion and removal. For the circuit ID suboption, the module field is the slot number of the module.

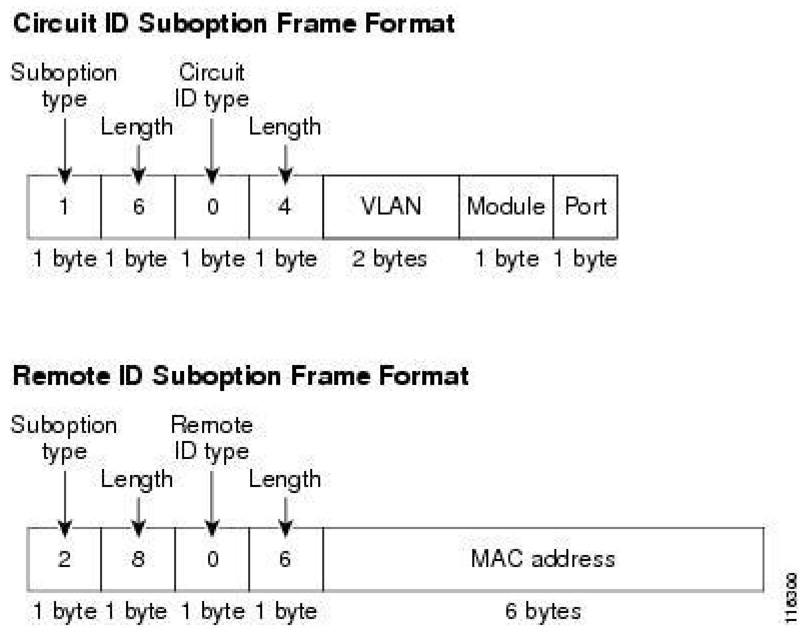
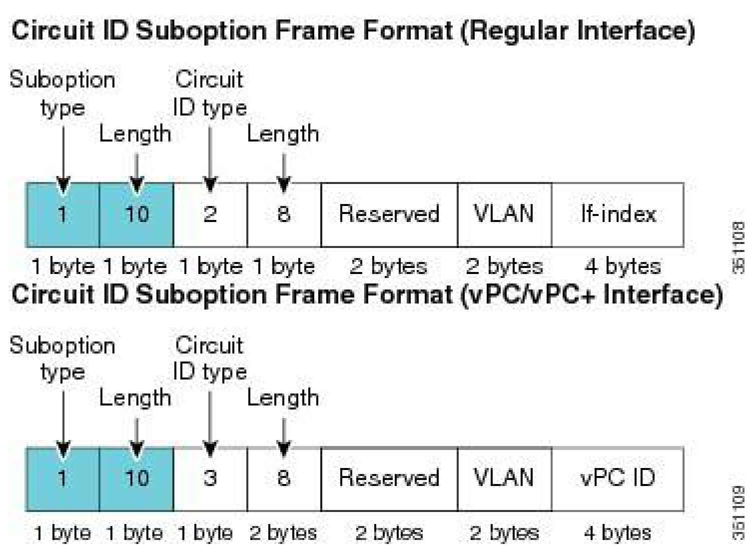


Figure 31： Circuit ID Suboption Frame Format for Regular and vPC Interfaces

Beginning with Inspur INOS Release 8.2(3), a new circuit ID format is used when Option 82 is enabled in DHCP snooping. The new circuit ID format is used by default and cannot be disabled. However, you might need to configure the DHCP server for the new circuit ID format if it was using the old Option 82 format for IP address allocation. These figures show the new default circuit ID format that is used for regular interfaces and vPC interfaces when Option 82 is enabled for DHCP snooping.

The enhanced Option 82 format improves DHCP packet processing. For vPC and vPC+ interfaces, the new format assigns vPC peers a unique circuit ID in case some are configured with different port channel numbers.



18.3 Information About the DHCP Relay Agent

18.3.1 DHCP Relay Agent

You can configure the device to run a DHCP relay agent, which forwards DHCP packets between clients and servers. This feature is useful when clients and servers are not on the same physical subnet. Relay agents receive DHCP messages and then generate a new DHCP message to send out on another interface. The relay agent sets the gateway address (giaddr field of the DHCP packet) and, if configured, adds the relay agent information option (Option 82) in the packet and forwards it to the DHCP server. The reply from the server is forwarded back to the client after removing Option 82.

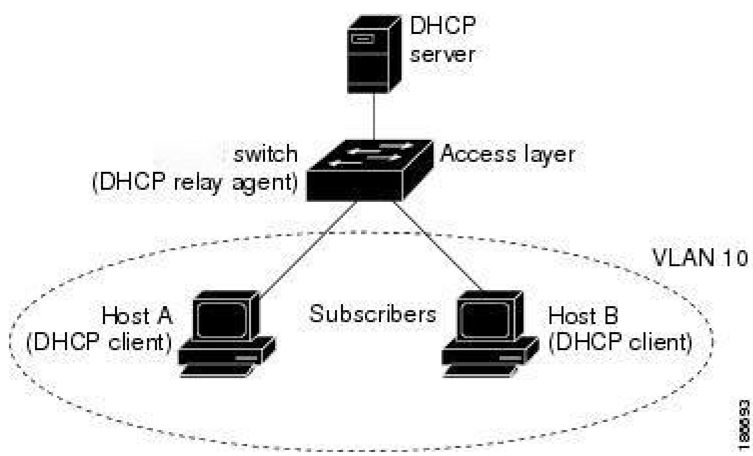
After you enable Option 82, the device uses the binary ifindex format by default. If needed, you can change the Option 82 setting to use an encoded string format instead.

18.3.2 DHCP Relay Agent Option 82

You can enable the device to insert and remove Option 82 information on DHCP packets that are forwarded by the relay agent.

Figure 32： DHCP Relay Agent in a Metropolitan Ethernet Network

This figure shows an example of a metropolitan Ethernet network in which a centralized DHCP server assigns IP addresses to subscribers connected to the device at the access layer. Because the DHCP clients and their associated DHCP server do not reside on the same IP network or subnet, a DHCP relay agent is configured with a helper address to enable broadcast forwarding and to transfer DHCP messages between the clients and the server.



When you enable Option 82 for the DHCP relay agent on the Inspur INOS device, the following sequence of events occurs:

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.
2. When the Inspur INOS device receives the DHCP request, it adds the Option 82 information in the packet. The Option 82 information contains the device MAC address (the remote ID suboption) and the port identifier, vlan-mod-port, from which the packet is received (the circuit ID suboption). In DHCP relay, the circuit ID is filled with the if_index of the SVI or Layer 3 interface on which DHCP relay is configured.
3. The device adds the IP address of the relay agent to the DHCP packet.
4. The device forwards the DHCP request that includes the Option 82 field to the DHCP server.
5. The DHCP server receives the packet. If the server is Option 82 capable, it can use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server echoes the Option 82 field in the DHCP reply.

6. The DHCP server unicasts the reply to the Inspur INOS device if the request was relayed to the server by the device. The Inspur INOS device verifies that it originally inserted the Option 82 data by inspecting the remote ID and possibly the circuit ID fields. The Inspur INOS device removes the Option 82 field and forwards the packet to the interface that connects to the DHCP client that sent the DHCP request.

18.4 Information About the DHCPv6 Relay Agent

18.4.1 DHCPv6 RelayAgent

You can configure the device to run a DHCPv6 relay agent, which forwards DHCPv6 packets between clients and servers. This feature is useful when clients and servers are not on the same physical subnet. Relay agents receive DHCPv6 messages and then generate a new DHCPv6 message to send out on another interface. The relay agent sets the gateway address (giaddr field of the DHCPv6 packet) and forwards it to the DHCPv6 server.

18.4.2 VRF Support for the DHCPv6 Relay Agent

You can configure the DHCPv6 relay agent to forward DHCPv6 broadcast messages from clients in a virtual routing and forwarding (VRF) instance to DHCPv6 servers in a different VRF. By using a single DHCPv6 server to provide DHCP support to clients in multiple VRFs, you can conserve IP addresses by using a single IP address pool rather than one for each VRF. For general information about VRFs, see the *Inspur CN12700 Series INOS Unicast Routing Configuration Guide*.

18.5 Information About DHCP Response Redirect

In a secured fabric network, a DHCP server is deployed as a shared service in a network, which is different from the fabric end points. Every fabric edge is configured as a DHCP relay agent to relay the DHCP traffic between the fabric end points and the DHCP server. A border node uses a fabric border as the packet forwarder to communicate with the DHCP server. Also, a any-cast address is configured across all the fabric edge nodes.

When a DHCP relay agent intercepts a DISCOVER packet, the DHCP relay agent sets a any-cast address as the gateway address (giaddr) and inserts the Option-82 information in the packet, which includes the circuit ID and remote ID suboptions. The DHCP server sends the OFFER packet with the destination as giaddr.

However, forwarding the OFFER packet to the correct switch is difficult because the any-cast address is the same on the edge network.

From Inspur INOS Release 8.2(1), you can use the **ip dhcp redirect-response** command on a DHCP server-facing interface to redirect packets to the correct switch. When you run this command, the border node processes the SERVER REPLY packets. When the DHCP server sends the OFFER packets, the border node uses the information from the remote ID option to create a VXLAN header that includes the source locator set as the outer destination address, and the VXLAN Network Identifier of the client segment. This helps the border node send the OFFER packet to the correct switch.

18.6 Virtualization Support for DHCP

The following information applies to DHCP used in virtual device contexts (VDCs):

- DHCP snooping binding databases are unique per VDC. Bindings in one VDC do not affect DHCP snooping in other VDCs.
- The system does not limit the binding database size on a per-VDC basis.
- The DHCP smart relay agent can be configured independently in default and nondefault VDCs.

18.7 Licensing Requirements for DHCP

This table shows the licensing requirements for DHCP.

Product	License Requirement
Inspur INOS	DHCP requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

18.8 Prerequisites for DHCP

DHCP has the following prerequisite:

- You should be familiar with DHCP before you configure DHCP snooping or the DHCP relay agent.

18.9 Guidelines and Limitations for DHCP

DHCP has the following configuration guidelines and limitations:

- If you are using both the Unicast reverse Packeting Forwarding (uRFP) strict mode in your client vPC VLANs and the First Hop Redundancy Protocol (FHRP) with the DHCP relay feature, the DHCP requests are sourced from the physical egress IP address interface (not the FHRP VIP) by default. Consequently, if your DHCP server is not on a directly connected subnet and you have multiple ECMP routes back to your vPC pair, some packets might land on the neighbor switch instead of the originating switch and be dropped by RFP. This behavior is expected. To avoid this scenario, perform one of the following workarounds:
- Use the uRFP loose mode, not uRFP strict.
- Configure static routes for the interface address on the affected FHRP interfaces and redistribute the static routes into IGP.
- Using the **ip dhcp relay source-interface** *interface-name*command, you can configure a different interface as the source interface.
- For Inspur INOS Release 8.2(3) and later releases, you must enable the insertion of Option 82 information for DHCP packets to support the highest DHCP snooping scale.
- After System Switchover, DHCP Global stats show incorrect values as they are not stored in PSS and get erased. Updating stats in PSS during packet path will affect scale.
- If you use DHCP relay where DHCP clients and servers are in different VRF instances, use only one DHCP server within a VRF.
- Before globally enabling DHCP snooping on the device, make sure that the devices acting as the DHCP server and the DHCP relay agent are configured and enabled.
- DHCP snooping does not work with DHCP relay configured on the same CN device.
- If a VLAN ACL (VACL) is configured on a VLAN that you are configuring with DHCP snooping, ensure that the VACL permits DHCP traffic between DHCP servers and DHCP hosts. When both DHCP snooping and DHCP relay are enabled on a VLAN and the SVI of that VLAN, DHCP relay takes precedence.
- If an ingress router ACL is configured on a Layer 3 interface that you are configuring with a DHCP server address, ensure that the router ACL permits DHCP traffic between DHCP servers and DHCP hosts.
- Access-control list (ACL) statistics are not supported if the DHCP snooping feature is enabled.
- Before using POAP, make sure that DHCP snooping is enabled and firewall rules are set to block unintended or malicious DHCP servers.
- When you configure DHCPv6 server addresses on an interface, a destination interface cannot be used with global IPv6 addresses.
- The following guidelines and limitations are applicable for the DHCP redirect reponse feature:
- Supported on the L3 or SVI interfaces.
- This feature is also supported on a Secure Fabric configured with VRF leaking.

18.10 Default Settings for DHCP

This table lists the default settings for DHCP parameters.

Table 40： Default DHCP Parameters

Parameters	Default
DHCP feature	Disabled
DHCP snooping	Disabled
DHCP snooping on VLANs	Disabled
DHCP snooping MAC address verification	Enabled
DHCP snooping Option 82 support	Disabled
DHCP snooping trust	Untrusted
DHCP relay agent	Enabled
DHCPv6 relay agent	Enabled
Lightweight DHCPv6 Relay Agent	Disabled
UDP Relay feature	Disabled
VRF support for the DHCP relay agent	Disabled
VRF support for the DHCPv6 relay agent	Disabled
DHCP relay sub-option type Inspur	Disabled
DHCPv6 relay option type Inspur	Disabled
DHCP Option 82 for relay agent	Disabled
DHCP server IP address	None

18.11 Configuring DHCP

18.11.1 Minimum DHCP Configuration

Step 1 Enable the DHCP feature.

When the DHCP feature is disabled, you cannot configure DHCP snooping.

Step 2 Enable DHCP snooping globally.

Step 3 Enable DHCP snooping on at least one VLAN.

By default, DHCP snooping is disabled on all VLANs.

Step 4 Ensure that the DHCP server is connected to the device using a trusted interface.

Step 5 (Optional) Configure an interface with the IP address of the DHCP server.

Related Topics

Enabling or Disabling the DHCP Feature
Enabling or Disabling DHCP Snooping Globally.
Enabling or Disabling DHCP Snooping on a VLAN

Configuring an Interface as Trusted or Untrusted.
Enabling or Disabling the DHCP Relay Agent.
Enabling or Disabling Option 82 for the DHCP Relay Agent
Configuring DHCP Server Addresses on an Interface.

18.11.2 Enabling or Disabling the DHCP Feature

You can enable or disable the DHCP feature on the device. By default, DHCP is disabled.

When the DHCP feature is disabled, you cannot configure DHCP snooping, the DHCP relay agent, or any of the features that depend on DHCP, such as dynamic ARP inspection and IP Source Guard. In addition, all DHCP, dynamic ARP inspection, and IP Source Guard configuration is removed from the device.

SUMMARY STEPS

1. **config t**
2. [**no**] **feature dhcp**
3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **feature dhcp** **Example:** `switch(config)# feature dhcp`	Enables the DHCP feature. The **no** option disables the DHCP feature and erases all DHCP configuration.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling or Disabling DHCP Snooping Globally.

18.11.3 Enabling or Disabling DHCP Snooping Globally

You can enable or disable DHCP snooping globally on the device.

Before you begin

Ensure that you have enabled the DHCP feature.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp snooping**

3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping** **Example:** `switch(config)# ip dhcp snooping`	Enables DHCP snooping globally. The **no** option disables DHCP snooping.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling or Disabling the DHCP Feature.

18.11.4 Enabling or Disabling DHCP Snooping on a VLAN

You can enable or disable DHCP snooping on one or more VLANs. By default, DHCP snooping is disabled on all VLANs.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. config t
2. [**no**] **ip dhcp snooping vlan** *vlan-list*
3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping vlan** *vlan-list* **Example:**	Enables DHCP snooping on the VLANs specified by *vlan-list*. The **no** option disables DHCP

	Command or Action	Purpose
	`switch(config)# ip dhcp snooping vlan 100,200,250-252`	snooping on the VLANs specified.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling or Disabling the DHCP Feature.

18.11.5 Enabling or Disabling DHCP Snooping MAC Address Verification

You can enable or disable DHCP snooping MAC address verification. If the device receives a packet on an untrusted interface and the source MAC address and the DHCP client hardware address do not match, address verification causes the device to drop the packet. MAC address verification is enabled by default.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp snooping verify mac-address**
3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping verify mac-address** **Example:** `switch(config)# ip dhcp snooping verify mac-address`	Enables DHCP snooping MAC address verification. The **no** option disables MAC address verification.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling or Disabling the DHCP Feature.

18.11.6 Enabling or Disabling Option 82 Data Insertion and Removal

You can enable or disable the insertion and removal of Option 82 information for DHCP packets forwarded without the use of the DHCP relay agent. By default, the device does not include Option 82 information in DHCP packets.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp snooping information option**
3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp snooping information option** **Example:** `switch(config)# ip dhcp snooping information option`	Enables the insertion and removal of Option 82 information for DHCP packets. The **no** option disables the insertion and removal of Option 82 information.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling or Disabling the DHCP Feature
Enabling or Disabling Option 82 for the DHCP Relay Agent

18.11.7 Configuring an Interface as Trusted or Untrusted

You can configure whether an interface is a trusted or untrusted source of DHCP messages. By default, all interfaces are untrusted. You can configure DHCP trust on the following types of interfaces:

- Layer 2 Ethernet interfaces
- Layer 2 port-channel interfaces

Before you begin

Ensure that the DHCP feature is enabled.
Ensure that the interface is configured as a Layer 2 interface.

SUMMARY STEPS

1. **config t**
2. Do one of the following options:
   - **interface ethernet** *slot*/*port*
   - **interface port-channel** *channel-number*
3. **[no] ip dhcp snooping trust**
4. (Optional) **show running-config dhcp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Do one of the following options:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-number*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		• Enters interface configuration mode, where *slot*/*port* is the Layer 2 Ethernet interface that you want to configure as trusted or untrusted for DHCP snooping.
		• Enters interface configuration mode, where *slot*/*port* is the Layer 2 port-channel interface that you want to configure as trusted or untrusted for DHCP snooping.

Step 3	[**no**] **ip dhcp snooping trust** **Example:** `switch(config-if)# ip dhcp snooping trust`	Configures the interface as a trusted interface for DHCP snooping. The **no** option configures the port as an untrusted interface.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling or Disabling the DHCP Feature.

18.11.8 Enabling or Disabling DHCP Relay Trusted Port Functionality

You can enable or disable the DHCP relay trusted port functionality. By default, if the gateway address is set to all zeros in the DHCP packet and the relay information option is already present in the packet, the DHCP relay agent will not discard the packet. If the **ip dhcp relay information option trust** command is configured globally, the DHCP relay agent will discard the packet if the gateway address is set to all zeros.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp relay information option trust**
3. (Optional) **show ip dhcp relay**
4. (Optional) **show ip dhcp relay information trusted-sources**
5. (Optional) **show running-config dhcp**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp relay information option trust** **Example:** `switch(config)# ip dhcp relay information option trust`	Enables the DHCP relay trusted port functionality. The **no** option disables this functionality.
Step 3	(Optional) **show ip dhcp relay** **Example:** `switch(config)# show ip dhcp relay`	Displays the DHCP relay configuration.
Step 4	(Optional) **show ip dhcp relay information trusted-sources** **Example:** `switch(config)# show ip dhcp relay information trusted-sources`	Displays the DHCP relay trusted ports configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

18.11.9 Configuring an Interface as a DHCP Relay Trusted or Untrusted Port

You can configure whether a Layer 3 interface is a DHCP relay trusted or untrusted interface. By default, all interfaces are untrusted. You can configure DHCP relay trust on the following types of interfaces:

- Layer 3 Ethernet interfaces and sub-interfaces
- Layer 3 port-channel interfaces
- Interface VLAN

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **config t**
2. Do one of the following options:
   - **interface ethernet** *slot*/*port*.[*number*]
   - **interface port-channel** *channel-number*.[*subchannel-id*]
   - **interface vlan** *vlan-id*
3. **[no] ip dhcp relay information trusted**
4. **show ip dhcp relay information trusted-sources**
5. (Optional) **show running-config dhcp**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Do one of the following options:
	• **interface ethernet** *slot*/*port*.[*number*]
	• **interface port-channel** *channel-number*.[*subchannel-id*]
	• **interface vlan** *vlan-id*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		• Enters interface configuration mode, where *slot*/*port* is the Layer 3 Ethernet interface that you want to configure as trusted or untrusted.
		• Enters interface configuration mode, where *channel-number* is the Layer 3 port-channel interface that you want to configure as trusted or untrusted.
		• Enters interface configuration mode, where *vlan-id* is the VLAN interface that you want to configure as trusted or untrusted.

Step 3	[**no**] **ip dhcp relay information trusted** **Example:** `switch(config-if)# ip dhcp relay information trusted`	Configures the interface as a trusted interface for DHCP relay agent information. The **no** option configures the port as an untrusted interface. **Note** For any L3 interface, if the interface is configured as trusted either through global command or interface-level command, the interface is considered as a trusted interface. Hence, when the trusted-port command is enabled at Global level, any L3 interface cannot be considered as untrusted irrespective of the interface-level configuration.
Step 4	**show ip dhcp relay information trusted-sources** **Example:**	Displays the DHCP relay trusted ports configuration.

	Command or Action	Purpose
	`switch(config-if)# show ip dhcp relay information trusted-sources`	
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

18.11.10 Configuring all Interfaces as Trusted or Untrusted

You can configure all Layer 3 interfaces as DHCP relay trusted or untrusted interfaces. By default, all interfaces are untrusted. You can configure DHCP relay trust on the following types of interfaces:

- Layer 3 Ethernet interfaces and sub-interfaces
- Layer 3 port-channel interfaces
- Interface VLAN

When you enable the **ip dhcp relay information trust-all** command, any Layer 3 interface cannot be considered as untrusted irrespective of the interface-level configuration.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp relay information trust-all**
3. **show ip dhcp relay information trusted-sources**
4. (Optional) **show running-config dhcp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp relay information trust-all** **Example:** `switch(config)# ip dhcp relay information trust-all`	Configures the interfaces as trusted sources of DHCP messages. The **no** option configures the ports as untrusted interfaces.
Step 3	**show ip dhcp relay information trusted-sources** **Example:** `switch(config)# show ip dhcp relay information trusted-sources`	Displays the DHCP relay trusted ports configuration.

	Command or Action	Purpose
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling or Disabling the DHCP Relay Agent

You can enable or disable the DHCP relay agent. By default, the DHCP relay agent is enabled.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **config t**
2. [**no**] **ip dhcp relay**
3. (Optional) **show ip dhcp relay**
4. (Optional) **show running-config dhcp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp relay** **Example:** `switch(config)# ip dhcp relay`	Enables the DHCP relay agent. The **no** option disables the relay agent.
Step 3	(Optional) **show ip dhcp relay** **Example:** `switch(config)# show ip dhcp relay`	Displays the DHCP relay configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling or Disabling the DHCP Feature.

18.11.11 Enabling or Disabling the DHCP Relay Source Interface

You can enable or disable the DHCP relay source interface. You can configure a different interface as the source of the DHCP relay agent.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip dhcp relay source-interface** *interface-name*
3. (Optional) **show ip dhcp relay source-interface**
4. (Optional) **show running-config dhcp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp relay source-interface** *interface-name* **Example:** `switch(config)# ip dhcp relay source-interface Ethernet1/1`	Enables the DHCP relay source interface. You can configure a different interface as the source of the DHCP relay agent, The **no** option disables the relay source interface.
Step 3	(Optional) **show ip dhcp relay source-interface** **Example:** `switch(config)# show ip dhcp relay source-interface`	Displays the DHCP relay source-interface configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

18.11.12 Enabling or Disabling Option 82 for the DHCP Relay Agent

You can enable or disable the device to insert and remove Option 82 information on DHCP packets forwarded by the relay agent.

By default, the DHCP relay agent does not include Option 82 information in DHCP packets.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip dhcp relay**
3. [**no**] **ip dhcp relay information option**

4. (Optional) **show ip dhcp relay**
5. (Optional) **show running-config dhcp**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip dhcp relay** **Example:** `switch(config)# ip dhcp relay`	Enables the DHCP relay feature. The **no** option disables this behavior.
Step 3	[**no**] **ip dhcp relay information option** **Example:** `switch(config)# ip dhcp relay information option`	Enables the DHCP relay agent to insert and remove Option 82 information on the packets that it forwards. The Option 82 information is in binary ifindex format by default. The **no** option disables this behavior.
Step 4	(Optional) **show ip dhcp relay** **Example:** `switch(config)# show ip dhcp relay`	Displays the DHCP relay configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

18.11.13 Configuring DHCP Server Addresses on an Interface

You can configure DHCP server IP addresses on an interface. When an inbound DHCP BOOTREQUEST packet arrives on the interface, the relay agent forwards the packet to all DHCP server IP addresses specified. The relay agent forwards replies from all DHCP servers to the host that sent the request.

Before you begin

Ensure that the DHCP feature is enabled.
Ensure that the DHCP server is correctly configured.
Determine the IP address for each DHCP server that you want to configure on the interface.
If the DHCP server is in a different VRF instance than the interface, ensure that you have enabled VRF support.

SUMMARY STEPS

1. **config t**
2. Do one of the following options:
- **interface ethernet** *slot*/*port*[**.** *number*]
- **interface vlan** *vlan-id*

- **interface port-channel** *channel-id*[.*subchannel-id*]

3. **ip dhcp relay address** *IP-address*
4. (Optional) **show ip dhcp relay address**
5. (Optional) **show running-config dhcp**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Do one of the following options:
	• **interface ethernet** *slot*/*port*[**.** *number*]
	• **interface vlan** *vlan-id*
	• **interface port-channel** *channel-id*[.*subchannel-id*]
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		• Enters interface configuration mode, where *slot*/*port* is the physical Ethernet interface that you want to configure with a DHCP server IP address. If you want to configure a subinterface, include the *number* argument to specify the subinterface number.
		• Enters interface configuration mode, where *vlan-id* is the ID of the VLAN that you want to configure with a DHCP server IP address.
		• Enters interface configuration mode, where *channel-id* is the ID of the port channel that you want to configure with a DHCP server IP address. If you want to configure a subchannel, include the *subchannel-id* argument to specify the subchannel ID.

Step 3	**ip dhcp relay address** *IP-address* **Example:** `switch(config-if)# ip dhcp relay address 10.132.7.120`	Configures an IP address for a DHCP server to which the relay agent forwards BOOTREQUEST packets received on this interface. To configure more than one IP address, use the **ip dhcp relay address** command once per address.
Step 4	(Optional) **show ip dhcp relay address** **Example:** `switch(config-if)# show ip dhcp relay address`	Displays all the configured DHCP server addresses.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config`	Copies the running configuration to the startup configuration.

	Command or Action	Purpose
	startup-config	

Related Topics

Enabling or Disabling the DHCP Feature.

18.12 Configuring DHCPv6

18.12.1 Enabling or Disabling the DHCPv6 Relay Agent

You can enable or disable the DHCPv6 relay agent. By default, the DHCPv6 relay agent is enabled.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ipv6 dhcp relay**
3. (Optional) **show ipv6 dhcp relay** [**interface** *interface*]
4. (Optional) **show running-config dhcp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ipv6 dhcp relay** **Example:** `switch(config)# ipv6 dhcp relay`	Enables the DHCPv6 relay agent. The **no** option disables the relay agent.
Step 3	(Optional) **show ipv6 dhcp relay** [**interface** *interface*] **Example:** `switch(config)# show ipv6 dhcp relay`	Displays the DHCPv6 relay configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

18.12.2 Enabling or Disabling VRF Support for the DHCPv6 Relay Agent

You can configure the device to support the relaying of DHCPv6 requests that arrive on an interface in one VRF to a DHCPv6 server in a different VRF.

Before you begin

Ensure that the DHCP feature is enabled.
Ensure that the DHCPv6 relay agent is enabled.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ipv6 dhcp relay option vpn**
3. [**no**] **ipv6 dhcp relay option type Inspur**
4. (Optional) **show ipv6 dhcp relay** [**interface** *interface*]
5. (Optional) **show running-config dhcp**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ipv6 dhcp relay option vpn** **Example:** `switch(config)# ipv6 dhcp relay option vpn`	Enables VRF support for the DHCPv6 relay agent. The **no** option disables this behavior.
Step 3	[**no**] **ipv6 dhcp relay option type Inspur** **Example:** `switch(config)# ipv6 dhcp relay option type Inspur`	Causes the DHCPv6 relay agent to insert virtual subnet selection (VSS) details as part of the vendor-specific option. The **no** option causes the DHCPv6 relay agent to insert VSS details as part of the VSS option (68), which is defined in RFC-6607. This command is useful when you want to use DHCPv6 servers that do not support RFC-6607 but allocate IPv6 addresses based on the client VRF name.
Step 4	(Optional) **show ipv6 dhcp relay** [**interface** *interface*] **Example:** `switch(config)# show ipv6 dhcp relay`	Displays the DHCPv6 relay configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

18.12.3 Configuring DHCPv6 Server Addresses on an Interface

You can configure DHCPv6 server IP addresses on an interface. When an inbound DHCP BOOTREQUEST packet arrives on the interface, the relay agent forwards the packet to all DHCPv6 server IP addresses specified. The relay agent forwards replies from all DHCPv6 servers to the host that sent the request.

Before you begin

Ensure that the DHCP feature is enabled.
Ensure that the DHCPv6 server is correctly configured.
Determine the IP address for each DHCPv6 server that you want to configure on the interface.
If the DHCPv6 server is in a different VRF than the interface, ensure that you have enabled VRF support.

SUMMARY STEPS

1. config t
2. Do one of the following options:
   - **interface ethernet** *slot*/*port*[**.** *number*]
   - **interface port-channel** *channel-id*[.*subchannel-id*]
3. [**no**] **ipv6 dhcp relay address** *IPv6-address*
4. (Optional) **show running-config dhcp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Do one of the following options:
	• **interface ethernet** *slot*/*port*[**.** *number*]
	• **interface port-channel** *channel-id*[.*subchannel-id*]
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		• Enters interface configuration mode, where *slot*/*port* is the physical Ethernet interface that you want to configure with a DHCPv6 server IP address. If you want to configure a subinterface, include the *number* argument to specify the subinterface number.
		• Enters interface configuration mode, where *channel-id* is the ID of the port channel that you want to configure with a DHCPv6 server IP address. If you want to configure a subchannel, include the *subchannel-id* argument to specify the subchannel ID.

Step 3	[**no**] **ipv6 dhcp relay address** *IPv6-address* **Example:** `switch(config-if)# ipv6 dhcp relay address FF02:1::FF0E:8C6C`	Configures an IP address for a DHCPv6 server to which the relay agent forwards BOOTREQUEST packets received on this interface. Use the **use-vrf** option to specify the VRF name of the server if it is in a different VRF and the other argument interface is used to specify the output interface for the

	Command or Action	Purpose
		destination. The server address can either be a link-scoped unicast or multicast address or a global or site-local unicast or multicast address. The **interface** option is mandatory for a link-scoped server address and multicast address. It is not allowed for a global or site-scoped server address. To configure more than one IP address, use the **ipv6 dhcp relay address** command once per address.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Displays the DHCPv6 configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

18.12.4 Configuring the DHCPv6 Relay Source Interface

You can configure the source interface for the DHCPv6 relay agent. By default, the DHCPv6 relay agent uses the relay agent address as the source address of the outgoing packet. Configuring the source interface enables you to use a more stable address (such as the loopback interface address) as the source address of relayed messages.

Before you begin

Ensure that the DHCP feature is enabled. Ensure that the DHCPv6 relay agent is enabled.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ipv6 dhcp relay source-interface** *interface*
3. (Optional) **show ipv6 dhcp relay** [**interface** *interface*]
4. (Optional) **show running-config dhcp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ipv6 dhcp relay source-interface** *interface* **Example:** `switch(config)# ipv6 dhcp relay source-interface loopback 2`	Configures the source interface for the DHCPv6 relay agent. **Note** The DHCPv6 relay source interface can be configured globally, per interface, or

	Command or Action	Purpose
		both. When both the global and interface levels are configured, the interface-level configuration overrides the global configuration.
Step 3	(Optional) **show ipv6 dhcp relay** [**interface** *interface*] **Example:** `switch(config)# show ipv6 dhcp relay`	Displays the DHCPv6 relay configuration.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

18.13 Configuring DHCP Response Redirect

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Enable the DHCP feature:
switch(config)# **feature dhcp**

Step 3 Specify the DHCP server-facing interface:
switch(config)# **interface ethernet** *slot/ethernet*

Step 4 Configure DHCP response redirect:
switch(config-if)# **[no] ip dhcp redirect-response**

Step 5 Exit the interface and global configuration modes:
switch(config-if)# **end**

Step 6 (Optional) Display the DHCP configuration:
switch# **show running-config dhcp**

Example: Configuring DHCP Response Redirect

The following running configuration example shows how to configure DHCP response redirect on a DHCP server-facing interface. Replace the <*placeholders*> with relevant values for your setup.

```
configure terminal
 interface Ethernet <2/1>
 ip dhcp redirect-response
 end
```

The following example shows the DHCP response redirect configuration details:

```
switch# show running-config dhcp

 !Command: show running-config dhcp
 !Time: Fri Dec 11 09:36:15 2016
```

```
version 8.2(0)SK(1)
feature dhcp

service  dhcp
ip dhcp relay
ipv6 dhcp relay


interface Ethernet2/1
ip dhcp redirect-response
```

18.14 Verifying the DHCP Configuration

To display DHCP configuration information, perform one of the following tasks. For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show running-config dhcp [**all**]	Displays the DHCP configuration in the running configuration.
show ip dhcp relay	Displays the DHCP relay configuration.
show ipv6 dhcp relay [**interface** *interface*]	Displays the DHCPv6 relay global or interface-level configuration.
show ip dhcp relay address	Displays all the DHCP server addresses configured on the device.
show ip dhcp snooping	Displays general information about DHCP snooping.
show startup-config dhcp [**all**]	Displays the DHCP configuration in the startup configuration.

18.15 Displaying DHCP Bindings

Use the **show ip dhcp snooping binding** command to display the DHCP binding table. For detailed information about the fields in the output from this command, see the *Inspur CN12700 Series INOS Security Command Reference*.

18.16 Clearing the DHCP Snooping Binding Database

You can remove entries from the DHCP snooping binding database, including a single entry, all entries associated with an interface, or all entries in the database.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. (Optional) **clear ip dhcp snooping binding**
2. (Optional) **clear ip dhcp snooping binding interface ethernet** *slot*/*port*[.*subinterface-number*]
3. (Optional) **clear ip dhcp snooping binding interface port-channel** *channel-number*[.*subchannel-number*]
4. (Optional) **clear ip dhcp snooping binding vlan** *vlan-id* **mac** *mac-address* **ip** *ip-address* **interface** {**ethernet** *slot*/*port*[.*subinterface-number* | **port-channel** *channel-number*[.*subchannel-number*] }

5. (Optional) **show ip dhcp snooping binding**

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **clear ip dhcp snooping binding** **Example:** `switch# clear ip dhcp snooping binding`	Clears all entries from the DHCP snooping binding database.
Step 2	(Optional) **clear ip dhcp snooping binding interface ethernet** *slot*/*port*[.*subinterface-number*] **Example:** `switch# clear ip dhcp snooping binding interface ethernet 1/4`	Clears entries associated with a specific Ethernet interface from the DHCP snooping binding database.
Step 3	(Optional) **clear ip dhcp snooping binding interface port-channel** *channel-number*[.*subchannel-number*] **Example:** `switch# clear ip dhcp snooping binding interface port-channel 72`	Clears entries associated with a specific port-channel interface from the DHCP snooping binding database.
Step 4	(Optional) **clear ip dhcp snooping binding vlan** *vlan-id* **mac** *mac-address* **ip** *ip-address* **interface** {**ethernet** *slot*/*port*[.*subinterface-number* | **port-channel** *channel-number*[.*subchannel-number*] } **Example:** `switch# clear ip dhcp snooping binding vlan 23 mac 0060.3aeb.54f0 ip 10.34.54.9 interface ethernet 2/11`	Clears a single, specific entry from the DHCP snooping binding database.
Step 5	(Optional) **show ip dhcp snooping binding** **Example:** `switch# show ip dhcp snooping binding`	Displays the DHCP snooping binding database.

Related Topics

Enabling or Disabling the DHCP Feature.

18.17 Clearing DHCP Relay Statistics

Use the **clear ip dhcp relay statistics** command to clear the global DHCP relay statistics.

Use the **clear ip dhcp relay statistics interface** *interface* command to clear the DHCP relay statistics for a particular interface.

18.18 Clearing DHCPv6 Relay Statistics

Use the **clear ipv6 dhcp relay statistics** command to clear the global DHCPv6 relay statistics.

Use the **clear ipv6 dhcp relay statistics interface** *interface* command to clear the DHCPv6 relay statistics for a particular interface.

18.19 Monitoring DHCP

Use the **show ip dhcp snooping statistics** command to monitor DHCP snooping.

Use the **show ip dhcp relay statistics** [**interface** *interface*] command to monitor DHCP relay statistics at the global or interface level.

Use the (Optional) **show ip dhcp snooping statistics vlan** [*vlan-id*] **interface** [**ethernet**|*port-channel*][*id*] command to know the exact statistics about snooping statistics per interface under a vlan.

Use the **show ipv6 dhcp relay statistics** [**interface** *interface*] command to monitor DHCPv6 relay statistics at the global or interface level.

18.20 Additional References for DHCP

Related Documents

Related Topic	Document Title
DHCP commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Inspur CN12700 Series INOS Security Command Reference*
VRFs and Layer 3 virtualization	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*
	Inspur CN12700 Series INOS Interfaces Configuration Guide

Standards

Standards	Title
RFC-2131	Dynamic Host Configuration Protocol
RFC-3046	DHCP Relay Agent Information Option
RFC-6607	Virtual Subnet Selection Options for DHCPv4 and DHCPv6

18.21 Feature History for DHCP

This table lists the release history for this feature.

Table 41： Feature History for DHCP

Feature Name	Releases	Feature Information	
IP DHCP Relay Source Interface	8.2(3)	Added support for the DHCP relay source interface.	
DHCP	8.2(3)	Added support for the DHCP redirect response feature.	
DHCP	8.2(3)	Added support for the DHCPv6 relay agent.	
DHCP	8.2(3)	Added a new default circuit ID format that is used when Option 82 is enabled for DHCP snooping.	
DHCP	8.2(3)	No change from Release 8.2(3)	

DHCP	8.2(3)	Deprecated the **service dhcp** command and replaced it with the **ip dhcp relay** command.	

CHAPTER 19 Configuring IPv6 First-Hop Security

This chapter describes the IPv6 First-Hop Security features. This chapter includes the following sections:

- Finding Feature Information
- Introduction to First-Hop Security
- RA Guard
- DHCPv6 Guard
- IPv6 Snooping
- How to Configure IPv6 FHS
- Configuration Examples
- Additional References for IPv6 First-Hop Security
- Feature History for IPv6 First-Hop Security

19.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

19.2 Introduction to First-Hop Security

The Layer 2 and Layer 3 switches operate in the Layer 2 domains with technologies such as server virtualization, Overlay Transport Virtualization (OTV), and Layer 2 mobility. These devices are sometimes referred to as "first hops", specifically when they are facing end nodes. The First-Hop Security feature provides end node protection and optimizes link operations on IPv6 or dual-stack networks.

First-Hop Security (FHS) is a set of features to optimize IPv6 link operation, as well as help with scale in large L2 domains. These features provide protection from a wide host of rogue or mis-configured users, and this can be extended with additional features for different deployment scenarios, or attack vectors.

Starting with Inspur INOS Release 8.2(3), the following FHS features are supported:

- IPv6 RA Guard
- DHCPv6 Guard
- IPv6 Snooping

19.2.1 IPv6 Global Policies

IPv6 global policies provide storage and access policy database services. IPv6 snooping and IPv6 RA guard are IPv6 global policies features. Every time IPv6 snooping or RA guard is configured globally, the policy attributes are stored in the software policy database. The policy is then applied to an interface, and the software policy database entry is updated to include this interface to which the policy is applied.

19.2.2 IPv6 First-Hop Security Binding Table

The IPv6 First-Hop Security Binding Table recovery mechanism feature enables the binding table to recover in the event of a device reboot. A database table of IPv6 neighbors connected to the device is created from information sources such as IPv6 snooping. This database, or binding, table is used by various IPv6 guard features to validate the link-layer address (LLA), the IPv4 or IPv6 address, and prefix binding of the neighbors to prevent spoofing and redirect attacks.

This mechanism enables the binding table to recover in the event of a device reboot. The recovery mechanism will block any data traffic sourced from an unknown source; that is, a source not already specified in the binding table and previously learned through snooping or DHCP gleaning. This feature recovers the missing binding table entries when the resolution for a destination address fails in the destination guard. When a failure occurs, a binding table entry is recovered by querying the DHCP server or the destination host, depending on the configuration.

19.3 RA Guard

19.3.1 Overview of IPv6 RA Guard

The IPv6 RA Guard feature provides support for allowing the network administrator to block or reject unwanted or rogue RA guard messages that arrive at the network device platform. RAs are used by devices to announce themselves on the link. The IPv6 RA Guard feature analyzes these RAs and filters out RAs that are sent by unauthorized devices. In host mode, all RA and router redirect messages are disallowed on the port. The RA guard feature compares configuration information on the Layer 2 (L2) device with the information found in the received RA frame. Once the L2 device has validated the content of the RA frame and router redirect frame against the configuration, it forwards the RA to its unicast or multicast destination. If the RA frame content is not validated, the RA is dropped.

19.3.2 Guidelines and Limitations of IPv6 RA Guard

The guidelines and limitations of IPv6 RA Guard are as follows:

- The IPv6 RA Guard feature does not offer protection in environments where IPv6 traffic is tunneled.
- This feature is supported only in hardware when the ternary content addressable memory (TCAM) is programmed.
- This feature can be configured on a switch port interface in the ingress direction.
- This feature supports host mode and router mode.
- This feature is supported only in the ingress direction; it is not supported in the egress direction.
- This feature is not supported on EtherChannel and EtherChannel port members.
- This feature is not supported on trunk ports with merge mode.
- This feature is supported on auxiliary VLANs and private VLANs (PVLANs). In the case of PVLANs, primary VLAN features are inherited and merged with port features.
- Packets dropped by the IPv6 RA Guard feature can be spanned.
- If the **platform ipv6 acl icmp optimize neighbor-discovery** command is configured, the IPv6 RA Guard feature cannot be configured and an error message will be displayed. This command adds default global Internet Control Message Protocol (ICMP) entries that will override the RA guard ICMP entries.

19.4 DHCPv6 Guard

19.4.1 Overview of DHCP—DHCPv6 Guard

The DHCPv6 Guard feature blocks DHCP reply and advertisement messages that originate from unauthorized DHCP servers and relay agents that forward DHCP packets from servers to clients. Client messages or messages sent by relay agents from clients to servers are not blocked. The filtering decision is determined by the device role assigned to the receiving switch port, trunk, or VLAN. In addition, to provide a finer level of filter granularity, messages can be filtered based on the address of the sending server or relay agent, or by the prefixes and addresses ranges listed in the reply message. This functionality helps to prevent traffic redirection or denial of service (DoS).

Packets are classified into one of the three DHCP type messages. All client messages are always switched regardless of device role. DHCP server messages are only processed further if the device role is set to server. Further processing of server messages includes DHCP server advertisements (for source validation and server preference) and DHCP server replies (for permitted prefixes).

If the device is configured as a DHCP server, all the messages need to be switched, regardless of the device role configuration.

19.4.2 Limitation of DHCPv6 Guard

The DHCPv6 guard feature is not supported on Etherchannel ports.

19.5 IPv6 Snooping

19.5.1 Overview of IPv6 Snooping

IPv6 "snooping," feature bundles several Layer 2 IPv6 first-hop security features, which operates at Layer 2, or between Layer 2 and Layer 3, and provides IPv6 features with security and scalability. This feature mitigates some of the inherent vulnerabilities for the neighbor discovery mechanism, such as attacks on duplicate address detection (DAD), address resolution, device discovery, and the neighbor cache.

IPv6 snooping learns and secures bindings for stateless autoconfiguration addresses in Layer 2 neighbor tables and analyzes snooping messages in order to build a trusted binding table. IPv6 snooping messages that do not have valid bindings are dropped. An IPv6 snooping message is considered trustworthy if its IPv6-to-MAC mapping is verifiable.

When IPv6 snooping is configured on a target (which varies depending on platform target support and may include device ports, switch ports, Layer 2 interfaces, Layer 3 interfaces, and VLANs), capture instructions are downloaded to the hardware to redirect the snooping protocol and Dynamic Host Configuration Protocol (DHCP) for IPv6 traffic up to the switch integrated security features (SISF) infrastructure in the routing device. For snooping traffic, messages such as NS, NA, RS, RA, and REDIRECT are directed to SISF. For DHCP, UDP messages sourced from port 546 or 547 are redirected.

IPv6 snooping registers its "capture rules" to the classifier, which aggregates all rules from all features on a given target and installs the corresponding ACL down into the platform-dependent modules. Upon receiving redirected traffic, the classifier calls all entry points from any registered feature (for the target on which the traffic is being received), including the IPv6 snooping entry point. This entry point is the last to be called, so any decision (such as drop) made by another feature supersedes the IPv6 snooping decision.

IPv6 snooping provides IPv6 host liveness tracking so that a neighbor table can be immediately updated when an IPv6 host disappears.

Additionally, IPv6 snooping is the foundation for many other IPv6 features that depend on an accurate binding table. It inspects snooping and DHCP messages on a link to glean addresses, and then populates the binding table with these addresses. This feature also enforces address ownership and limits the number of addresses any given node is allowed to claim.

19.5.2 Restrictions for IPv6 Snooping

The IPv6 snooping feature is not supported on Etherchannel ports.

19.6 How to Configure IPv6 FHS

19.6.1 Configuring the IPv6 RA Guard Policy on the Device

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 nd raguard policy** *policy-name*
3. **device-role {host | router | monitor | switch}**
4. **hop-limit** {**maximum** | **minimum** *limit*}

5. **managed-config-flag {on | off}**
6. **other-config-flag {on | off}**
7. **router-preference maximum {high | low | medium}**
8. **trusted-port**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ipv6 nd raguard policy** *policy-name* **Example:** `Device(config)# ipv6 nd raguard policy policy1`	Defines the RA guard policy name and enters RA guard policy configuration mode.
Step 3	**device-role {host | router | monitor | switch}** **Example:** `Device(config-ra-guard)# device-role router`	Specifies the role of the device attached to the port.
Step 4	**hop-limit {maximum | minimum** *limit*} **Example:** `Device(config-ra-guard)# hop-limit minimum 3`	(Optional) Enables verification of the advertised hop count limit. • If not configured, this check will be bypassed.
Step 5	**managed-config-flag {on | off}** **Example:** `Device(config-ra-guard)# managed-config-flag on`	(Optional) Enables verification that the advertised managed address configuration flag is on. • If not configured, this check will be bypassed.
Step 6	**other-config-flag {on | off}** **Example:** `Device(config-ra-guard)# other-config-flag on`	(Optional) Enables verification of the advertised "other" configuration parameter.
Step 7	**router-preference maximum {high | low | medium}** **Example:** `Device(config-ra-guard)# router-preference maximum high`	(Optional) Enables verification that the advertised default router preference parameter value is lower than or equal to a specified limit.
Step 8	**trusted-port** **Example:** `Device(config-ra-guard)# trusted-port`	(Optional) Specifies that this policy is being applied to trusted ports. • All RA guard policing will be disabled.
Step 9	**exit** **Example:** `Device(config-ra-guard)# exit`	Exits RA guard policy configuration mode and returns to global configuration mode.

19.6.2 Configuring IPv6 RA Guard on an Interface

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*
3. **ipv6 nd raguard attach-policy** [*policy-name*]
4. **exit**
5. **show ipv6 nd raguard policy** [*policy-name*]
6. **debug ipv6 snooping raguard** [*filter* | *interface* | *vlanid*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *type number* **Example:** `Device(config)# interface fastethernet 3/13`	Specifies an interface type and number, and places the device in interface configuration mode.
Step 3	**ipv6 nd raguard attach-policy** [*policy-name*] **Example:** `Device(config-if)# ipv6 nd raguard attach-policy`	Applies the IPv6 RA Guard feature to a specified interface.
Step 4	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

Step 5	**show ipv6 nd raguard policy** [*policy-name*]
	Example:
	`switch# show ipv6 nd raguard policy host`
	`Policy host configuration:`
	`device-role host`
	`Policy applied on the following interfaces:`
	`Et0/0   vlan all`
	`Et1/0   vlan all`
		Displays the RA guard policy on all interfaces configured with the RA guard.

Step 6	**debug ipv6 snooping raguard** [*filter* | *interface* | *vlanid*] **Example:** `Device# debug ipv6 snooping raguard`	Enables debugging for IPv6 RA guard snooping information.

19.6.3 Configuring DHCP—DHCPv6 Guard

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 dhcp guard policy** *policy-name*
3. **device-role {client | server}**
4. **preference min** *limit*

5. **preference max** *limit*
6. **trusted-port**
7. **exit**
8. **interface** *type number*
9. **switchport**
10. **ipv6 dhcp guard** [**attach-policy** *policy-name*]
11. **exit**
12. **vlan configuration** *vlan-id*
13. **ipv6 dhcp guard** [**attach-policy** *policy-name*]
14. **exit**
15. **exit**
16. **show ipv6 dhcp guard policy** [*policy-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ipv6 dhcp guard policy** *policy-name* **Example:** `Device(config)# ipv6 dhcp guard policy pol1`	Defines the DHCPv6 guard policy name and enters DHCP guard configuration mode.
Step 3	**device-role** {**client** | **server**} **Example:** `Device(config-dhcp-guard)# device-role server`	Specifies the device role of the device attached to the target (interface or VLAN).
Step 4	**preference min** *limit* **Example:** `Device(config-dhcp-guard)# preference min 0`	(Optional) Enables verification that the advertised preference (in preference option) is greater than the specified limit. If not specified, this check will be bypassed.
Step 5	**preference max** *limit* **Example:** `Device(config-dhcp-guard)# preference max 255`	(Optional) Enables verification that the advertised preference (in preference option) is less than the specified limit. If not specified, this check will be bypassed.
Step 6	**trusted-port** **Example:** `Device(config-dhcp-guard)# trusted-port`	(Optional) Specifies that this policy is being applied to trusted ports. All DHCP guard policing will be disabled.
Step 7	**exit** **Example:** `Device(config-dhcp-guard)# exit`	Exits DHCP guard configuration mode and returns to global configuration mode.
Step 8	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/2/0`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 9	**switchport** **Example:** `Device(config-if)# switchport`	Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 10	**ipv6 dhcp guard** [**attach-policy** *policy-name*] **Example:** `Device(config-if)# ipv6 dhcp guard attach-policy pol1`	Attaches a DHCPv6 guard policy to an interface.
Step 11	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	**vlan configuration** *vlan-id* **Example:** `Device(config)# vlan configuration 1`	Specifies a VLAN and enters VLAN configuration mode.
Step 13	**ipv6 dhcp guard** [**attach-policy** *policy-name*] **Example:** `Device(config-vlan-config)# ipv6 dhcp guard attach-policy pol1`	Attaches a DHCPv6 guard policy to a VLAN.
Step 14	**exit** **Example:** `Device(config-vlan-config)# exit`	Exits VLAN configuration mode and returns to global configuration mode.
Step 15	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 16	**show ipv6 dhcp guard policy** [*policy-name*] **Example:** `Device# show ipv6 dhcp policy guard pol1`	(Optional) Displays the policy configuration as well as all the interfaces where the policy is applied.

19.6.4 Configuring IPv6Snooping

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 snooping policy** *snooping-policy*
3. **ipv6 snooping attach-policy** *snooping-policy*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**ipv6 snooping policy** *snooping-policy* **Example:** `Device(config)# ipv6 snooping policy policy1`	Configures an IPv6 snooping policy and enters IPv6 snooping configuration mode.
Step 3	**ipv6 snooping attach-policy** *snooping-policy* **Example:** `Device(config-ipv6-snooping)# ipv6 snooping attach-policy policy1`	Attaches the IPv6 snooping policy to a target.

19.6.5 Configuring IPv6 First-Hop Security BindingTable

SUMMARY STEPS

1. **configure terminal**
2. **ipv6 neighbor binding vlan** *vlan-id* {**interface** *type number* | *ipv6-address* | *mac-address*} [**tracking** [**disable** | **enable** | **retry-interval** *value*] | **reachable-lifetime** *value*]
3. **ipv6 neighbor binding max-entries** *entries* [**vlan-limit** *number* | **interface-limit** *number* | **mac-limit** *number*]
4. **ipv6 neighbor binding logging**
5. **ipv6 neighbor tracking retry-interval** *value*
6. **exit**
7. **show ipv6 neighbor binding** [**vlan** *vlan-id* | **interface** *type number* | **ipv6** *ipv6-address* | **mac** *mac-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ipv6 neighbor binding vlan** *vlan-id* {**interface** *type number* | *ipv6-address* | *mac-address*} [**tracking** [**disable** | **enable** | **retry-interval** *value*] | **reachable-lifetime** *value*] **Example:** `Device(config)# ipv6 neighbor binding vlan 100 interface Ethernet 0/0 reachable-lifetime 100`	Adds a static entry to the binding table database.
Step 3	**ipv6 neighbor binding max-entries** *entries* [**vlan-limit** *number* | **interface-limit** *number* | **mac-limit** *number*] **Example:** `Device(config)# ipv6 neighbor binding max-entries 100`	Specifies the maximum number of entries that are allowed to be inserted in the binding table cache.
Step 4	**ipv6 neighbor binding logging**	Enables the logging of binding table main events.

	Command or Action	Purpose
	Example: `Device(config)# ipv6 neighbor binding logging`	
Step 5	**ipv6 neighbor tracking retry-interval** *value* **Example:** `Device(config)# ipv6 neighbor binding retry-interval 8`	Tracks entries in the binding table.
Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 7	**show ipv6 neighbor binding** [**vlan** *vlan-id* | **interface** *type number* | **ipv6** *ipv6-address* | **mac** *mac-address*] **Example:** `Device# show ipv6 neighbor binding`	Displays the contents of a binding table.

19.6.6 Verifying and Troubleshooting IPv6 Snooping

SUMMARY STEPS

1. **show ipv6 snooping capture-policy** [**interface** *type number*]
2. **show ipv6 snooping counter** [**interface** *type number*]
3. **show ipv6 snooping features**
4. **show ipv6 snooping policies** [**interface** *type number*]
5. **debug ipv6 snooping**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ipv6 snooping capture-policy** [**interface** *type number*] **Example:** `Device# show ipv6 snooping capture-policy interface ethernet 0/0`	Displays snooping message capture policies.
Step 2	**show ipv6 snooping counter** [**interface** *type number*] **Example:** `Device# show ipv6 snooping counter interface FastEthernet 4/12`	Displays information about the packets counted by the interface counter.
Step 3	**show ipv6 snooping features** **Example:** `Device# show ipv6 snooping features`	Displays information about snooping features configured on the device.
Step 4	**show ipv6 snooping policies** [**interface** *type number*]	Displays information about the configured policies and the interfaces to which they are attached.

	Command or Action	Purpose
	Example: `Device# show ipv6 snooping policies`	
Step 5	**debug ipv6 snooping** **Example:** `Device# debug ipv6 snooping`	Enables debugging for snooping information in IPv6.

19.7 Configuration Examples

19.7.1 Example: IPv6 RA Guard Configuration

```
Device(config)# interface fastethernet 3/13

Device(config-if)# ipv6 nd raguard attach-policy

Device# show running-config interface fastethernet 3/13

Building  configuration...
Current configuration : 129 bytes
!
interface FastEthernet3/13
 switchport
 switchport access vlan 222
 switchport mode access
 access-group mode prefer port
 ipv6 nd raguard
end
```

19.7.2 Example: Configuring DHCP—DHCPv6 Guard

The following example displays a sample configuration for DHCPv6 Guard:

```
configure terminal
ipv6 dhcp guard policy pol1
 device-role server
 preference min 0
 preference max 255
 trusted-port
interface GigabitEthernet 0/2/0
 switchport
 ipv6 dhcp guard attach-policy pol1
 vlan configuration 1
  ipv6 dhcp guard attach-policy pol1
show ipv6 dhcp guard policy pol1
```

19.7.3 Example: Configuring IPv6 First-Hop Security Binding Table

```
config terminal
 ipv6 neighbor binding vlan 100 2001:db8::1 interface ethernet3/0
 ipv6 neighbor binding max-entries 100
 ipv6 neighbor binding logging
 ipv6 neighbor binding retry-interval 8
 exit
```

```
show ipv6 neighbor binding
```

19.7.4 Example: Configuring IPv6 Snooping

```
switch (config)# ipv6 snooping policy policy1
switch(config-ipv6-snooping)# ipv6 snooping attach-policy policy1
switch(config-ipv6-snooping)# exit
.
.
.
Device# show ipv6 snooping policies policy1
Policy policy1 configuration:
  trusted-port
  device-role node
Policy applied on the following interfaces:
   Et0/0       vlan all
   Et1/0       vlan all
Policy applied on the following vlans:
   vlan 1-100,200,300-400
```

19.8 Additional References for IPv6 First-Hop Security

This section includes additional information related to configuring IPv6 First-Hop Security.

Related Documents

Related Topic	Document Title
Inspur INOS Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*

19.9 Feature History for IPv6 First-Hop Security

This table lists the release history for this feature.

Table 42：Feature History for IPv6 First-Hop Security

Feature Name	Releases	Feature Information
IPv6 First-Hop Security	8.2(3)	Added support for the following IPv6 First-Hop Security features: • IPv6 RA Guard • DHCPv6 Guard • IPv6 Snooping

CHAPTER 20 Configuring Dynamic ARP Inspection

This chapter describes how to configure dynamic Address Resolution Protocol (ARP) inspection (DAI) on a Inspur INOS device.

This chapter includes the following sections:

- Finding Feature Information
- Information About DAI
- Virtualization Support for DAI
- Licensing Requirements for DAI
- Prerequisites for DAI
- Guidelines and Limitations for DAI
- Default Settings for DAI
- Configuring DAI
- Verifying the DAI Configuration
- Monitoring and Clearing DAI Statistics
- Configuration Examples for DAI
- Configuring ARP ACLs
- Verifying the ARP ACL Configuration
- Additional References for DAI
- Feature History for DAI

20.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

20.2 Information About DAI

20.2.1 ARP

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP address to a MAC address. For example, host B wants to send information to host A but does not have the MAC address of host A in its ARP cache. In ARP terms, host B is the sender and host A is the target.

To get the MAC address of host A, host B generates a broadcast message for all hosts within the broadcast domain to obtain the MAC address associated with the IP address of host A. All hosts within the broadcast domain receive the ARP request, and host A responds with its MAC address.

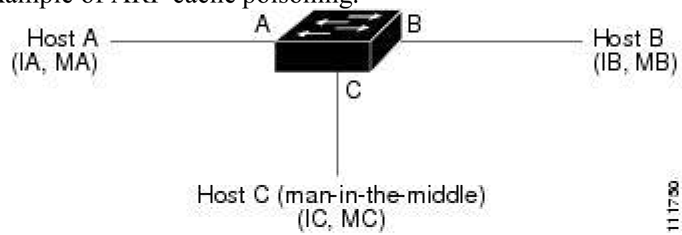
20.2.2 ARP Spoofing Attacks

ARP spoofing attacks and ARP cache poisoning can occur because ARP allows a reply from a host even if an ARP request was not received. After the attack, all traffic from the device under attack flows through the attacker's computer and then to the router, switch, or host.

An ARP spoofing attack can affect hosts, switches, and routers connected to your Layer 2 network by sending false information to the ARP caches of the devices connected to the subnet. Sending false information to an ARP cache is known as ARP cache poisoning. Spoof attacks can also intercept traffic intended for other hosts on the subnet.

Figure 33： ARP Cache Poisoning

This figure shows an example of ARP cache poisoning.



Hosts A, B, and C are connected to the device on interfaces A, B, and C, which are on the same subnet. Their IP and MAC addresses are shown in parentheses; for example, host A uses IP address IA and MAC address MA. When host A needs to send IP data to host B, it broadcasts an ARP request for the MAC address associated with IP address IB. When the device and host B receive the ARP request, they populate their ARP caches with an ARP binding for a host with the IP address IA and a MAC address MA; for example, IP address IA is bound to MAC address MA. When host B responds, the device and host A populate their ARP caches with a binding for a host with the IP address IB and the MAC address MB.

Host C can poison the ARP caches of the device, host A, and host B by broadcasting two forged ARP responses with bindings: one for a host with an IP address of IA and a MAC address of MC and another for a host with the IP address of IB and a MAC address of MC. Host B and the device then use the MAC address MC as the destination MAC address for traffic intended for IA, which means that host C intercepts that traffic. Likewise, host A and the device use the MAC address MC as the destination MAC address for traffic intended for IB.

Because host C knows the true MAC addresses associated with IA and IB, it can forward the intercepted traffic to those hosts by using the correct MAC address as the destination. This topology, in which host C has inserted itself into the traffic stream from host A to host B, is an example of a *man-in-the middle* attack.

20.2.3 DAI and ARP Spoofing Attacks

DAI ensures that only valid ARP requests and responses are relayed. When DAI is enabled and properly configured, a Inspur CN device performs these activities:

- Intercepts all ARP requests and responses on untrusted ports
- Verifies that each of these intercepted packets has a valid IP-to-MAC address binding before updating the local ARP cache or before forwarding the packet to the appropriate destination
- Drops invalid ARP packets

DAI can determine the validity of an ARP packet based on valid IP-to-MAC address bindings stored in a Dynamic Host Configuration Protocol (DHCP) snooping binding database. This database is built by DHCP snooping if DHCP snooping is enabled on the VLANs and on the device. It can also contain static entries that you create. If the ARP packet is received on a trusted interface, the device forwards the packet without any checks. On untrusted interfaces, the device forwards the packet only if it is valid.

DAI can validate ARP packets against user-configured ARP access control lists (ACLs) for hosts with statically configured IP addresses. The device logs dropped packets.

You can configure DAI to drop ARP packets when the IP addresses in the packets are invalid or when the MAC addresses in the body of the ARP packets do not match the addresses specified in the Ethernet header.

Related Topics

Applying ARP ACLs to VLANs for DAI Filtering.
Logging DAI Packets.
Enabling or Disabling Additional Validation.

20.2.4 Interface Trust States and Network Security

DAI associates a trust state with each interface on the device. Packets that arrive on trusted interfaces bypass all

DAI validation checks, and packets that arrive on untrusted interfaces go through the DAI validation process.

In a typical network configuration, the guidelines for configuring the trust state of interfaces are as follows:

Untrusted

Interfaces that are connected to hosts

Trusted

Interfaces that are connected to devices

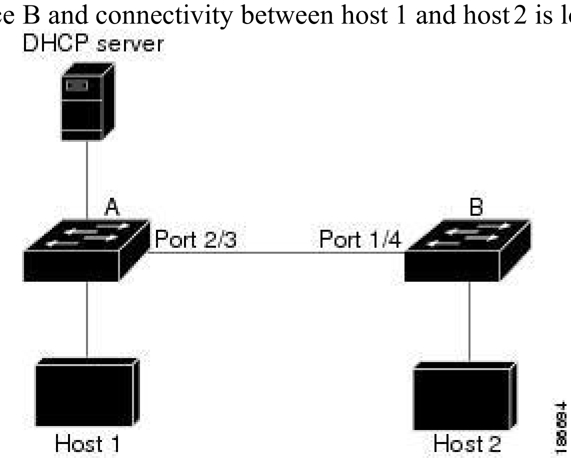
With this configuration, all ARP packets that enter the network from a device bypass the security check. No other validation is needed at any other place in the VLAN or in the network.

Caution	Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity.

Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity.

Figure 34：ARP Packet Validation on a VLAN Enabled for DAI

The following figure, assume that both device A and device B are running DAI on the VLAN that includes host 1 and host 2. If host 1 and host 2 acquire their IP addresses from the DHCP server connected to device A, only device A binds the IP-to-MAC address of host 1. If the interface between device A and device B is untrusted, the ARP packets from host 1 are dropped by device B and connectivity between host 1 and host 2 is lost.



If you configure interfaces as trusted when they should be untrusted, you may open a security hole in a network. If device A is not running DAI, host 1 can easily poison the ARP cache of device B (and host 2, if you configured the link between the devices as trusted). This condition can occur even though device B is running DAI.

DAI ensures that hosts (on untrusted interfaces) connected to a device that runs DAI do not poison the ARP caches of other hosts in the network; however, DAI does not prevent hosts in other portions of the network from poisoning the caches of the hosts that are connected to a device that runs DAI.

If some devices in a VLAN run DAI and other devices do not, the guidelines for configuring the trust state of interfaces on a device that runs DAI becomes the following:

Untrusted

Interfaces that are connected to hosts or to devices that *are not* running DAI

Trusted

Interfaces that are connected to devices that *are* running DAI

To validate the bindings of packets from devices that do not run DAI, configure ARP ACLs on the device that runs DAI. When you cannot determine the bindings, isolate at Layer 3 the devices that run DAI from devices that do not run DAI.

Related Topics

Configuring the DAI Trust State of a Layer 2 Interface
Example 2 One Device Supports DAI.

20.2.5 Prioritizing ARP ACLs and DHCP Snooping Entries

By default, DAI filters DAI traffic by comparing DAI packets to IP-MAC address bindings in the DHCP snooping database.

When you apply an ARP ACL to traffic, the ARP ACLs take precedence over the default filtering behavior. The device first compares ARP packets to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, the device denies the packet regardless of whether a valid IP-MAC binding exists in the DHCP snooping database.

Related Topics

Configuring ARP ACLs.
Applying ARP ACLs to VLANs for DAI Filtering.

20.2.6 Logging DAI Packets

Inspur INOS maintains a buffer of log entries about DAI packets processed. Each log entry contains flow information, such as the receiving VLAN, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

You can also specify the type of packets that are logged. By default, aInspur CN device logs only packets that DAI drops.

If the log buffer overflows, the device overwrites the oldest DAI log entries with newer entries. You can configure the maximum number of entries in the buffer.

Related Topics

Configuring the DAI Logging Buffer Size
Configuring DAI Log Filtering.

20.3 VirtualizationSupportforDAI

The following information applies to DAI used in virtual device contexts (VDCs):

- IP-MAC address bindings are unique per VDC.
- ARP ACLs are unique per VDC. You cannot use an ACL that you created in one VDC in a different VDC.
- Because ACLs are not shared by VDCs, you can reuse ACL names in different VDCs.
- The system does not limit ARP ACLs or rules on a per-VDC basis.

20.4 Licensing Requirements for DAI

This table shows the licensing requirements for DAI.

Product	License Requirement
Inspur INOS	DAI requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

20.5 Prerequisites for DAI

- You must enable the DHCP feature before you can configure DAI.

20.6 Guidelines and Limitations for DAI

DAI has the following configuration guidelines and limitations:

- DAI is an ingress security feature; it does not perform any egress checking.
- DAI is not effective for hosts connected to devices that do not support DAI or that do not have this feature enabled. Because man-in-the-middle attacks are limited to a single Layer 2 broadcast domain, you should separate the domain with DAI from domains without DAI. This separation secures the ARP caches of hosts in the domain with DAI.
- DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. If you want DAI to use static IP-MAC address bindings to determine if ARP packets are valid, DHCP snooping needs only to be enabled. If you want DAI to use dynamic IP-MAC address bindings to determine if ARP packets are valid, you must configure DHCP snooping on the same VLANs on which you configure DAI.
- When you use the **feature dhcp** command to enable the DHCP feature, there is a delay of approximately 30 seconds before the I/O modules receive the DHCP or DAI configuration. This delay occurs regardless of the method that you use to change from a configuration with the DHCP feature disabled to a configuration with the DHCP feature enabled. For example, if you use the Rollback feature to revert to a configuration that enables the DHCP feature, the I/O modules receive the DHCP and DAI configuration approximately 30 seconds after you complete the rollback.
- When DHCP snooping is disabled or used in a non-DHCP environment, you should use ARP ACLs to permit or to deny packets and disable DAI.
- DAI is supported on access ports, trunk ports, port-channel ports, and private VLAN ports.
- The DAI trust configuration of a port channel determines the trust state of all physical ports that you assign to the port channel. For example, if you have configured a physical port as a trusted interface and then you add that physical port to a port channel that is an untrusted interface, the physical port becomes untrusted.
- When you remove a physical port from a port channel, the physical port does not retain the DAI trust state configuration of the port channel.
- When you change the trust state on the port channel, the device configures a new trust state on all the physical ports that comprise the channel.
- If you want DAI to use static IP-MAC address bindings to determine if ARP packets are valid, ensure that DHCP snooping is enabled and that you have configured the static IP-MAC address bindings.
- If you want DAI to use dynamic IP-MAC address bindings to determine if ARP packets are valid, ensure that DHCP snooping is enabled.

20.7 Default Settings for DAI

This table lists the default settings for DAI parameters.

Table 43： Default DAI Parameters

Parameters	Default
DAI	Disabled on all VLANs.
Interface trust state	All interfaces are untrusted.
ARP ACLs for non-DHCP environments	No ARP ACLs are defined.
Validation checks	No checks are performed.

Log buffer	When DAI is enabled, all denied or dropped ARP packets are logged. The number of entries in the log is 32. The number of system messages is limited to 5 per second. The logging-rate interval is 1 second.
Per-VLAN logging	All denied or dropped ARP packets are logged.

20.8 Configuring DAI

20.8.1 Enabling or Disabling DAI on VLANs

You can enable or disable DAI on VLANs. By default, DAI is disabled on all VLANs.

Before you begin

If you are enabling DAI, ensure the following:

- Ensure that the DHCP feature is enabled.
- The VLANs on which you want to enable DAI are configured.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip arp inspection vlan** *list*
3. (Optional) **show ip arp inspection vlan** *list*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip arp inspection vlan** *list* **Example:** `switch(config)# ip arp inspection vlan 13`	Enables DAI for the specified list of VLANs. The **no** option disables DAI for the specified VLANs.
Step 3	(Optional) **show ip arp inspection vlan** *list* **Example:** `switch(config)# show ip arp inspection vlan 13`	Shows the DAI status for the specified list of VLANs.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

20.8.2 Configuring the DAI Trust State of a Layer 2 Interface

You can configure the DAI interface trust state of a Layer 2 interface. By default, all interfaces are untrusted. A device forwards ARP packets that it receives on a trusted Layer 2 interface but does not check them.

On untrusted interfaces, the device intercepts all ARP requests and responses and verifies that the intercepted packets have valid IP-MAC address bindings before updating the local cache and forwarding the packet to the appropriate destination. If the device determines that packets have invalid bindings, it drops the packets and logs them according to the logging configuration.

Before you begin

If you are enabling DAI, ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number* / *slot*
3. **[no] ip arp inspection trust**
4. (Optional) **show ip arp inspection interface** *type number* / *slot*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *type number* / *slot*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Enters interface configuration mode.

Step 3	[**no**] **ip arp inspection trust** **Example:** `switch(config-if)# ip arp inspection trust`	Configures the interface as a trusted ARP interface. The **no** option configures the interface as an untrusted ARP interface.
Step 4	(Optional) **show ip arp inspection interface** *type number* / *slot* **Example:** `switch(config-if)# show ip arp inspection interface ethernet 2/1`	Displays the trust state and the ARP packet rate for the specified interface.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Interface Trust States and Network Security
Configuring DAI Log Filtering.

20.8.3 Applying ARP ACLs to VLANs for DAI Filtering

You can apply an ARP ACL to one or more VLANs. The device permits packets only if the ACL permits them.

By default, no VLANs have an ARP ACL applied.

Before you begin

Ensure that the ARP ACL that you want to apply is correctly configured.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip arp inspection filter** *acl-name* **vlan** *list*
3. (Optional) **show ip arp inspection vlan** *list*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip arp inspection filter** *acl-name* **vlan** *list* **Example:** `switch(config)# ip arp inspection filter arp-acl-01 vlan 100`	Applies the ARP ACL to the list of VLANs, or if you use the **no** option, removes the ARP ACL from the list of VLANs.
Step 3	(Optional) **show ip arp inspection vlan** *list* **Example:** `switch(config)# show ip arp inspection vlan 100`	Shows the DAI status for the specified list of VLANs, including whether an ARP ACL is applied.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring ARP ACLs.

20.8.4 Enabling or Disabling Additional Validation

You can enable or disable additional validation of ARP packets. By default, no additional validation of ARP packets is enabled. When no additional validation is configured, the source MAC address and the source IP address check against the IP-to-MAC binding entry for ARP packets are done by using the Ethernet source MAC address (not the ARP sender MAC address) and the ARP sender IP address.

DAI intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. You can enable additional validation on the destination MAC address, the sender and target IP addresses, and the source MAC address.

You can use the following keywords with the **ip arp inspection validate** command to implement additional validations:

dst-mac

Checks the destination MAC address in the Ethernet header against the target MAC address in the ARP body for ARP responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

ip

Checks the ARP body for invalid and unexpected IP addresses. Addresses include 0.0.0.0, 255.255.255.255, and all IP multicast addresses. Sender IP addresses are checked in all ARP requests and responses, and target IP addresses are checked only in ARP responses.

src-mac

Checks the source MAC address in the Ethernet header against the sender MAC address in the ARP body for ARP requests and responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped.

When enabling additional validation, follow these guidelines:

- You must specify at least one of the keywords. You can specify one, two, or all three keywords.
- Each **ip arp inspection validate** command that you enter replaces the configuration from any previous commands. If you enter an **ip arp inspection validate** command to enable src-mac and dst-mac validations, and a second **ip arp inspection validate** command to enable ip validation, the src-mac and dst-mac validations are disabled when you enter the second command.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]}
3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]} **Example:** `switch(config)# ip arp inspection validate src-mac dst-mac ip`	Enables additional DAI validation, or if you use the **no** option, disables additional DAI validation.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP snooping configuration, including the DAI configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

20.8.5 Configuring the DAI Logging Buffer Size

You can configure the DAI logging buffer size. The default buffer size is 32 messages.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip arp inspection log-buffer entries** *number*
3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip arp inspection log-buffer entries** *number* **Example:** `switch(config)# ip arp inspection log-buffer entries 64`	Configures the DAI logging buffer size. The **no** option reverts to the default buffer size, which is 32 messages. The buffer size can be between 1 and 1024 messages.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP snooping configuration, including the DAI configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

20.8.6 Configuring DAI Log Filtering

You can configure how the device determines whether to log a DAI packet. By default, the device logs DAI packets that are dropped.

SUMMARY STEPS

1. **configure terminal**
2. Enter one of the following commands:
- **ip arp inspection vlan** *vlan-list* **logging dhcp-bindings all**
- **ip arp inspection vlan** *vlan-list* **logging dhcp-bindings none**
- **ip arp inspection vlan** *vlan-list* **logging dhcp-bindingspermit**
- **no ip arp inspection vlan** *vlan-list* **logging dhcp-bindings** {**all** | **none** | **permit**}

3. (Optional) **show running-config dhcp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:	Configures DAI log filtering, as follows. The **no**

	Command or Action	Purpose
	• **ip arp inspection vlan** *vlan-list* **logging dhcp-bindings all** • **ip arp inspection vlan** *vlan-list* **logging dhcp-bindings none** • **ip arp inspection vlan** *vlan-list* **logging dhcp-bindingspermit** • **no ip arp inspection vlan** *vlan-list* **logging dhcp-bindings** {**all** | **none** | **permit**} **Example:** `switch(config)# ip arp inspection vlan 100 dhcp-bindings permit`	option removes DAI log filtering. • Logs all packets that match DHCP bindings. • Does not log packets that match DHCP bindings. • Logs packets permitted by DHCP bindings. • Removes DAI log filtering.
Step 3	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP snooping configuration, including the DAI configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

20.9 Verifying the DAI Configuration

To display the DAI configuration information, perform one of the following tasks. For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show ip arp inspection	Displays the status of DAI.
show ip arp inspection interface ethernet	Displays the trust state.
show ip arp inspection vlan	Displays the DAI configuration for a specific VLAN.
show arp access-lists	Displays ARP ACLs.
how ip arp inspection log	Displays the DAI log configuration.

20.10 Monitoring and Clearing DAI Statistics

To monitor and clear DAI statistics, use the commands in this table. For more information about these commands, see the *Security Command Reference* for your Inspur CN device.

Command	Purpose
show ip arp inspection statistics	Displays DAI statistics.
clear ip arp inspection statistics vlan <id>	Clears DAI statistics.

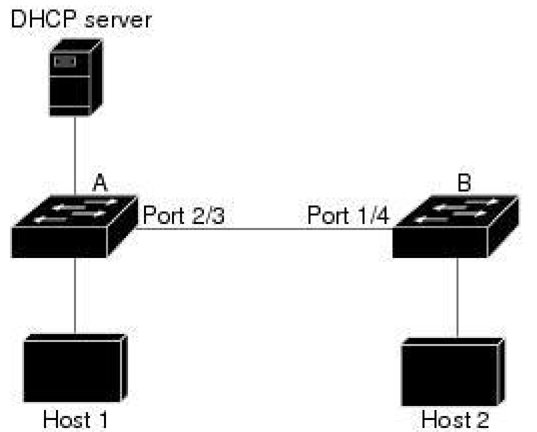
20.11 Configuration Examples for DAI

20.11.1 Example 1-Two Devices Support DAI

These procedures show how to configure DAI when two devices support DAI.

Figure 35：Two Devices Supporting DAI

The following figure shows the network configuration for this example. Host 1 is connected to device A, and Host 2 is connected to device B. Both devices are running DAI on VLAN 1 where the hosts are located. A DHCP server is connected to device A. Both hosts acquire their IP addresses from the same DHCP server. Device A has the bindings for Host 1 and Host 2, and device B has the binding for Host 2. Device A Ethernet interface 2/3 is connected to the device B Ethernet interface 1/4.



DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically-assigned IP addresses.

- This configuration does not work if the DHCP server is moved from device A to a different location.
- To ensure that this configuration does not compromise security, configure Ethernet interface 2/3 on device A and Ethernet interface 1/4 on device B as trusted.

Configuring Device A

To enable DAI and configure Ethernet interface 2/3 on device A as trusted, follow these steps:

Step 1 While logged into device A, verify the connection between device A and device B.

```
switchA# show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device ID              Local Intrfce   Hldtme  Capability  Platform      Port ID
switchB                Ethernet2/3     177     R S I    WS-C2960-24TC Ethernet1/4
switchA#
```

Step 2 Enable DAI on VLAN 1 and verify the configuration.

```
switchA# config t
switchA(config)# ip arp inspection vlan 1
switchA(config)# show ip arp inspection vlan 1
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
Vlan : 1
-----------
Configuration   : Enabled
```

```
Operation State : Active
switchA(config)#
```

Step 3 Configure Ethernet interface 2/3 as trusted.

```
switchA(config)#  interface  ethernet  2/3
switchA(config-if)# ip arp inspection trust
switchA(config-if)# exit
switchA(config)# exit
switchA# show ip arp inspection interface ethernet 2/3
 Interface        Trust State    Rate (pps)    Burst Interval
 -------------    -----------    ----------    --------------
 Ethernet2/3      Trusted          15             5
```

Step 4 Verify the bindings.

```
switchA# show ip dhcp snooping binding
MacAddress         IpAddress        LeaseSec  Type           VLAN  Interface
----------------- --------------- --------    -------------  ----  -------------
00:60:0b:00:12:89  10.0.0.1         0         dhcp-snooping  1     Ethernet2/3
switchA#
```

Step 5 Check the statistics before and after DAI processes any packets.

```
switchA# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded  = 0


ARP Res Forwarded  = 0
ARP Req Dropped    = 0
ARP Res Dropped    = 0
DHCP Drops         = 0
DHCP Permits       = 0
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req   = 0
IP Fails-ARP Res   = 0
switchA#
```

If host 1 sends out two ARP requests with an IP address of 10.0.0.1 and a MAC address of 0002.0002.0002, both requests are permitted, and are shown as follows:

```
switchA# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded = 2
ARP Res Forwarded = 0
ARP  Req  Dropped  =  0
ARP  Res  Dropped  =  0
DHCP Drops         = 0
DHCP  Permits     =  2
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP  Fails-ARP  Req  =  0
IP Fails-ARP Res   = 0
```

If host 1 tries to send an ARP request with an IP address of 10.0.0.3, the packet is dropped and an error message is logged.

```
00:12:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 2 Invalid ARPs (Req) on Ethernet2/3, vlan
1.([0002.0002.0002/10.0.0.3/0000.0000.0000/0.0.0.0/02:42:35 UTC Fri Jul 13 2008])
```

The statistics display as follows:

```
switchA# show ip arp inspection statistics vlan 1
switchA#
Vlan : 1
-----------
ARP Req Forwarded = 2
ARP Res Forwarded = 0
ARP  Req  Dropped  =  2
ARP  Res  Dropped  =  0
DHCP Drops         = 2
DHCP  Permits     =  2
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP  Fails-ARP  Req  =  0
IP  Fails-ARP  Res  =  0
switchA#
```

Configuring Device B

To enable DAI and configure Ethernet interface 1/4 on device B as trusted, follow these steps:

Step 1 While logged into device B, verify the connection between device B and device A.

```
switchB# show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device ID              Local Intrfce   Hldtme  Capability  Platform      Port ID
switchA                Ethernet1/4     120     R S I    WS-C2960-24TC Ethernet2/3
switchB#
```

Step 2 Enable DAI on VLAN 1, and verify the configuration.

```
switchB# config t
switchB(config)# ip arp inspection vlan 1
switchB(config)# show ip arp inspection vlan 1
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
Vlan : 1
-----------
Configuration   : Enabled
Operation State : Active
switchB(config)#
```

Step 3 Configure Ethernet interface 1/4 as trusted.

```
switchB(config)#  interface  ethernet  1/4
switchB(config-if)# ip arp inspection trust
switchB(config-if)# exit
switchB(config)# exit
switchB# show ip arp inspection interface ethernet 1/4
 Interface        Trust State    Rate (pps)    Burst Interval
 -------------    -----------    ----------    --------------
 Ethernet1/4      Trusted          15             5
```

```
switchB#
```

Step 4 Verify the list of DHCP snooping bindings.

```
switchB# show ip dhcp snooping binding
MacAddress         IpAddress        LeaseSec  Type           VLAN  Interface
----------------- --------------- --------    -------------  ----  -------------
00:01:00:01:00:01  10.0.0.2         4995      dhcp-snooping  1     Ethernet1/4
switchB#
```

Step 5 Check the statistics before and after DAI processes any packets.

```
switchB# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded = 0
ARP Res Forwarded = 0
ARP  Req  Dropped  =  0
ARP  Res  Dropped  =  0
DHCP Drops         = 0
DHCP  Permits     =  0
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0


IP Fails-ARP Req = 0
IP Fails-ARP Res = 0
switchB#
```

If Host 2 sends out an ARP request with the IP address 10.0.0.2 and the MAC address 0001.0001.0001, the packet is forwarded and the statistics are updated.

```
switchB# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded = 1
ARP Res Forwarded = 0
ARP Req Dropped = 0
ARP Res Dropped = 0
DHCP Drops         = 0
DHCP Permits   = 1
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req = 0
IP Fails-ARP Res = 0
switchB#
```

If Host 2 attempts to send an ARP request with the IP address 10.0.0.1, DAI drops the request and logs the following system message:

```
00:18:08: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Ethernet1/4, vlan
1.([0001.0001.0001/10.0.0.1/0000.0000.0000/0.0.0.0/01:53:21 UTC Fri Jun 13 2008])
```

The statistics display as follows:

```
switchB# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded = 1
ARP Res Forwarded = 0
```

```
ARP Req Dropped = 1
ARP Res Dropped = 0
DHCP Drops        = 1
DHCP Permits   = 1
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req = 0
IP Fails-ARP Res = 0
switchB#
```

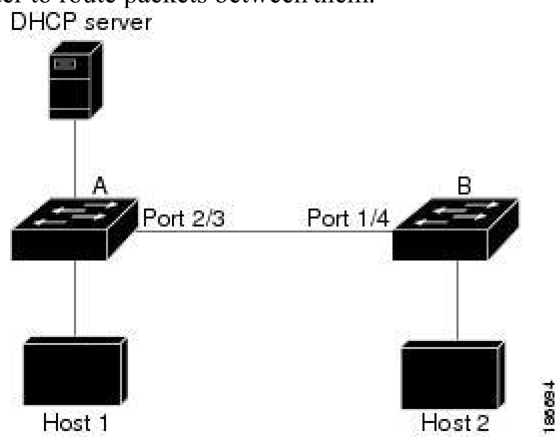
20.11.2 Example 2 One Device Supports DAI

This procedure shows how to configure DAI when the second device involved in the network configuration does not support DAI or DHCP snooping.

Figure 36 : One Device Supporting DAI

Device B, shown in this figure does not support DAI or DHCP snooping; therefore, configuring Ethernet interface 2/3 on device A as trusted creates a security hole because both device A and Host 1 could be attacked by either device B or Host 2.

To prevent this possibility, you must configure Ethernet interface 2/3 on device A as untrusted. To permit ARP packets from Host 2, you must set up an ARP ACL and apply it to VLAN 1. If the IP address of Host 2 is not static, which would make it impossible to accurately configure the ARP ACL on device A, you must separate device A from device B at Layer 3 and use a router to route packets between them.



Step 1 Configure the access list to permit the IP address 10.0.0.1 and the MAC address 0001.0001.0001, and verify the configuration.

```
switchA# config t
switchA(config)# arp access-list H2
switchA(config-arp-acl)# permit ip host 10.0.0.1 mac host 0001.0001.0001
switchA(config-arp-acl)# exit
switchA(config)# show arp access-lists H2
ARP access list H2
10 permit ip host 1.1.1.1 mac host 0001.0001.0001
switchA(config)#
```

Step 2 Apply the ACL to VLAN 1, and verify the configuration.

```
switchA(config)# ip arp inspection filter H2 vlan 1
```

```
switchA(config)# show ip arp inspection vlan 1
Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation   : Disabled
Vlan : 200
-----------
Configuration    : Enabled
Operation State : Active
ACL Match/Static : H2 / No
```

Step 3 Configure Ethernet interface 2/3 as untrusted, and verify the configuration.

Note By default, the interface is untrusted.

```
switchA(config)# interface ethernet 2/3
switchA(config-if)# no ip arp inspection trust
switchA(config-if)# exit
switchA# show ip arp inspection interface ethernet 2/3
switchA#
```

The **show ip arp inspection interface** command has no output because the interface has the default configuration, which includes an untrusted state.

When Host 2 sends 5 ARP requests through Ethernet interface 2/3 on device A and a "get" is permitted by device A, the statistics are updated.

```
switchA# show ip arp inspection statistics vlan 1
Vlan : 1
-----------
ARP Req Forwarded = 5
ARP Res Forwarded = 0
ARP Req Dropped = 0
ARP Res Dropped = 0
DHCP Drops         = 0
DHCP Permits   = 0
SMAC Fails-ARP Req = 0
SMAC Fails-ARP Res = 0
DMAC Fails-ARP Res = 0
IP Fails-ARP Req = 0
IP Fails-ARP Res = 0
switchA#
```

20.12 Configuring ARP ACLs

20.12.1 Session Manager Support for ARP ACLs

Session Manager supports the configuration of ARP ACLs. This feature allows you to create a configuration session and verify your ARP ACL configuration changes prior to committing them to the running configuration. For more information about Session Manager, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.

20.12.2 Creating an ARP ACL

You can create an ARP ACL on the device and add rules to it.

SUMMARY STEPS

1. **configure terminal**

2. **arp access-list** *name*
3. [*sequence-number*] {**permit** | **deny**} **ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**log**]
4. [*sequence-number*] {**permit** | **deny**} **request ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**log**]
5. [*sequence-number*] {**permit** | **deny**} **response ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} [**any** | **host** *target-IP* | *target-IP target-IP-mask*]] **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**any** | **host** *target-MAC* | *target-MAC target-MAC-mask*] [**log**]
6. (Optional) **show arp access-lists** *acl-name*
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**arp access-list** *name*
	
	Example:
	`switch(config)# arp access-list arp-acl-01`
	`switch(config-arp-acl)#`
		Creates the ARP ACL and enters ARP ACL configuration mode.

Step 3	[*sequence-number*] {**permit** | **deny**} **ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**log**] **Example:** `switch(config-arp-acl)# permit ip 192.168.2.0 255.2555.255.0 mac 00C0.4F00.0000 ffff.ff00.0000`	Creates a rule that permits or denies any ARP message based upon the IP address and MAC address of the sender of the message. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules.
Step 4	[*sequence-number*] {**permit** | **deny**} **request ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**log**] **Example:** `switch(config-arp-acl)# permit request ip 192.168.102.0 0.0.0.255 mac any`	Creates a rule that permits or denies ARP request messages based upon the IP address and MAC address of the sender of the message. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules.
Step 5	[*sequence-number*] {**permit** | **deny**} **response ip** {**any** | **host** *sender-IP* | *sender-IP sender-IP-mask*} [**any** | **host** *target-IP* | *target-IP target-IP-mask*]] **mac** {**any** | **host** *sender-MAC* | *sender-MAC sender-MAC-mask*} [**any** | **host** *target-MAC* | *target-MAC target-MAC-mask*] [**log**] **Example:** `switch(config-arp-acl)# permit response ip host 192.168.202.32 any mac host 00C0.4FA9.BCF3 any`	Creates a rule that permits or denies ARP response messages based upon the IPv4 address and MAC address of the sender and the target of the message. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules.
Step 6	(Optional) **show arp access-lists** *acl-name*	Shows the ARP ACL configuration.

	Command or Action	Purpose
	Example: `switch(config-arp-acl)# show arp access-lists arp-acl-01`	
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-arp-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

20.12.3 Changing an ARP ACL

You can change and remove rules in an existing ARP ACL. You cannot change existing rules. Instead, to change a rule, you can remove it and recreate it with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers.

SUMMARY STEPS

1. **configure terminal**
2. **arp access-list** *name*
3. (Optional) [*sequence-number*] {**permit** | **deny**} [**request** | **response**] **ip** *IP-data* **mac** *MAC-data*
4. (Optional) **no** {*sequence-number* | {**permit** | **deny**} [**request** | **response**] **ip** *IP-data* **mac** *MAC-data*
5. **show arp access-lists**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**arp access-list** *name*
	Example:
	`switch(config)# arp access-list arp-acl-01`
	`switch(config-acl)#`
		Enters ARP ACL configuration mode for the ACL that you specify by name.

Step 3	(Optional) [*sequence-number*] {**permit** | **deny**} [**request** | **response**] **ip** *IP-data* **mac** *MAC-data* **Example:** `switch(config-arp-acl)# 100 permit request ip 192.168.132.0 255.2555.255.0 mac any`	Creates a rule. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules.
Step 4	(Optional) **no** {*sequence-number* | {**permit** | **deny**} [**request** | **response**] **ip** *IP-data* **mac** *MAC-data* **Example:** `switch(config-arp-acl)# no 80`	Removes the rule that you specified from the ARP ACL.

	Command or Action	Purpose
Step 5	**show arp access-lists** **Example:** `switch(config-arp-acl)# show arp access-lists`	Displays the ARP ACL configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-arp-acl)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Creating an ARP ACL.
Changing Sequence Numbers in an ARP ACL.

20.12.4 Removing an ARP ACL

You can remove an ARP ACL from the device.

Before you begin

Ensure that you know whether the ACL is applied to a VLAN. The device allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of VLANs where you have applied the ACL. Instead, the device considers the removed ACL to be empty.

SUMMARY STEPS

1. **configure terminal**
2. **no arp access-list** *name*
3. **show arp access-lists**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no arp access-list** *name* **Example:** `switch(config)# no arp access-list arp-acl-01`	Removes the ARP ACL you specified by name from running configuration.
Step 3	**show arp access-lists** **Example:** `switch(config)# show arp access-lists`	Displays the ARP ACL configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

20.12.5 Changing Sequence Numbers in an ARP ACL

You can change all the sequence numbers assigned to rules in an ARP ACL.

SUMMARY STEPS

1. **configure terminal**
2. **resequence arp access-list** *name starting-sequence-number increment*
3. **show arp access-lists name**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**resequence arp access-list** *name starting-sequence-number increment*
	Example:
	`switch(config)# resequence arp access-list arp-acl-01 100 10`
	`switch(config)#`
		Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify.

Step 3	**show arp access-lists** *name* **Example:** `switch(config)# show arp access-lists arp-acl-01`	Displays the ARP ACL configuration for the ACL specified by the *name* argument.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

20.13 Verifying the ARP ACL Configuration

To display ARP ACL configuration information, use the commands in this table. For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show arp access-lists	Displays the ARP ACL configuration.
show running-config aclmgr	Displays ACLs in the running configuration.

20.14 Additional References for DAI

Related Documents

Related Topic	Document Title

DAI commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Inspur CN12700 Series INOS Security Command Reference*
DHCP snooping commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Inspur CN12700 Series INOS Security Command Reference*

Standards

Standards	Title
RFC-826	An Ethernet Address Resolution Protocol

20.15 Feature History for DAI

This table lists the release history for this feature.

Table 44： Feature History for DAI

Feature Name	Releases	Feature Information	
Dynamic ARP Inspection	8.2(3)	No change from Release 8.2(3)	
Dynamic ARP Inspection	8.2(3)	No change from Release 8.2(3).	

CHAPTER 21 Configuring IP Source Guard

This chapter describes how to configure IP Source Guard on Inspur INOS devices. This chapter includes the following sections:

- Finding Feature Information
- Information About IP Source Guard
- Licensing Requirements for IP Source Guard
- Prerequisites for IP Source Guard
- Guidelines and Limitations for IP Source Guard
- Default Settings for IP Source Guard
- Configuring IP Source Guard
- Displaying IP Source Guard Bindings
- Configuration Example for IP Source Guard
- Additional References for IP Source Guard
- Feature History for IP Source Guard

21.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

21.2 Information About IP Source Guard

IP Source Guard is a per-interface traffic filter that permits IP traffic only when the IP address and MAC address of each packet matches one of two sources of IP and MAC address bindings:

- Entries in the Dynamic Host Configuration Protocol (DHCP) snooping binding table.
- Static IP source entries that you configure.

Filtering on trusted IP and MAC address bindings helps prevent spoofing attacks, in which an attacker uses the IP address of a valid host to gain unauthorized network access. To circumvent IP Source Guard, an attacker would have to spoof both the IP address and the MAC address of a valid host.

You can enable IP Source Guard on Layer 2 interfaces that are not trusted by DHCP snooping. IP Source Guard supports interfaces that are configured to operate in access mode and trunk mode. When you initially enable IP Source Guard, all inbound IP traffic on the interface is blocked except for the following:

- DHCP packets, which DHCP snooping inspects and then forwards or drops, depending upon the results of inspecting the packet.
- IP traffic from static IP source entries that you have configured in the Inspur INOS device.

The device permits the IP traffic when DHCP snooping adds a binding table entry for the IP address and MAC address of an IP packet or when you have configured a static IP source entry.

The device drops IP packets when the IP address and MAC address of the packet do not have a binding table entry or a static IP source entry. For example, assume that the **show ip dhcp snooping binding** command displays the following binding table entry:

```
MacAddress         IpAddress    LeaseSec   Type        VLAN      Interface
----------         ----------   ---------  ------      -------    ---------
```

```
00:02:B3:3F:3B:99  10.5.5.2       6943    dhcp-snooping  10      Ethernet2/3



MacAddress         IpAddress    LeaseSec   Type         VLAN      Interface
----------         ----------   ---------  ------       -------    ---------
00:02:B3:3F:3B:99  10.5.5.2       6943    dhcp-snooping  10      Ethernet2/3
```

If the device receives an IP packet with an IP address of 10.5.5.2, IP Source Guard forwards the packet only if the MAC address of the packet is 00:02:B3:3F:3B:99.

21.2.1 Virtualization Support for IP Source Guard

The following information applies to IP Source Guard used in virtual device contexts (VDCs):

- IP-MAC address bindings are unique per VDC. Bindings in one VDC do not affect IP Source Guard in other VDCs.
- Inspur INOS does not limit the binding database size on a per-VDC basis.

21.3 Licensing Requirements for IP Source Guard

This table shows the licensing requirements for IP Source Guard.

Product	License Requirement
Inspur INOS	IP Source Guard requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

21.4 Prerequisites for IP Source Guard

IP Source Guard has the following prerequisite:

- You must enable the DHCP feature.

21.5 Guidelines and Limitations for IP Source Guard

IP Source Guard has the following configuration guidelines and limitations:

- IP Source Guard limits IP traffic on an interface to only those sources that have an IP-MAC address binding table entry or static IP source entry. When you first enable IP Source Guard on an interface, you may experience disruption in IP traffic until the hosts on the interface receive a new IP address from a DHCP server.
- IP Source Guard is dependent upon DHCP snooping to build and maintain the IP-MAC address binding table or upon manual maintenance of static IP source entries.

21.6 Default Settings for IP Source Guard

This table lists the default settings for IP Source Guard parameters.

Table 45： Default IP Source Guard Parameters

Parameters	Default
IP Source Guard	Disabled on each interface.

IP source entries	None. No static or default IP source entries exist by default.

21.7 Configuring IP Source Guard

21.7.1 Enabling or Disabling IP Source Guard on a Layer 2 Interface

You can enable or disable IP Source Guard on a Layer 2 interface. By default, IP Source Guard is disabled on all interfaces.

Before you begin

Ensure that the DHCP feature is enabled.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*
3. **[no] ip verify source dhcp-snooping-vlan**
4. (Optional) **show running-config dhcp**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		Enters interface configuration mode for the specified interface.

Step 3	[**no**] **ip verify source dhcp-snooping-vlan** **Example:** `switch(config-if)# ip verify source dhcp-snooping vlan`	Enables IP Source Guard on the interface. The **no** option disables IP Source Guard on the interface.
Step 4	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Displays the running configuration for DHCP snooping, including the IP Source Guard configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Adding or Removing a Static IP Source Entry.

21.7.2 Adding or Removing a Static IP Source Entry

You can add or remove a static IP source entry on a device. By default, there are no static IP source entries on a device.

SUMMARY STEPS

1. **configure terminal**
2. [**no**] **ip source binding** *IP-address MAC-address* **vlan** *vlan-ID* **interface ethernet** *slot*/*port*
3. (Optional) **show ip dhcp snooping binding** [**interface ethernet** *slot*/*port*]
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	[**no**] **ip source binding** *IP-address MAC-address* **vlan** *vlan-ID* **interface ethernet** *slot*/*port* **Example:** `switch(config)# ip source binding 10.5.22.17 001f.28bd.0013 vlan 100 interface ethernet 2/3`	Creates a static IP source entry for the current interface, or if you use the **no** option, removes a static IP source entry.
Step 3	(Optional) **show ip dhcp snooping binding** [**interface ethernet** *slot*/*port*] **Example:** `switch(config)# show ip dhcp snooping binding interface ethernet 2/3`	Displays IP-MAC address bindings for the interface specified, including static IP source entries. Static entries appear with the term in the Type column.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Enabling or Disabling IP Source Guard on a Layer 2 Interface
Displaying IP Source Guard Bindings.

21.8 Displaying IP Source Guard Bindings

Use the **show ip verify source** command to display IP-MAC address bindings.

21.9 Configuration Example for IP Source Guard

This example shows how to create a static IP source entry and then how to enable IP Source Guard on an interface.

```
ip source binding 10.5.22.17 001f.28bd.0013 vlan 100 interface ethernet 2/3
interface ethernet 2/3
  no shutdown
  ip verify source dhcp-snooping-vlan
```

21.10 Additional References for IP Source Guard

Related Documents

Related Topic	Document Title
IP Source Guard commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Inspur CN12700 Series INOS Security Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

Feature History for IP Source Guard

This table lists the release history for this feature.

Table 46： Feature History for IP Source Guard

Feature Name	Releases	Feature Information	
IP Source Guard	8.2(3)	No change from Release 8.2(3).	
IP Source Guard	8.2(3)	No change from Release 8.2(3)	

CHAPTER 22 Configuring Keychain Management

This chapter describes how to configure keychain management on a Inspur INOS device. This chapter includes the following sections:

- Finding Feature Information
- Information About Keychain Management
- Virtualization Support for Keychain Management
- Licensing Requirements for Keychain Management
- Prerequisites for Keychain Management
- Guidelines and Limitations for Keychain Management
- Default Settings for Keychain Management
- Configuring Keychain Management
- Determining Active Key Lifetimes
- Verifying the Keychain Management Configuration
- Configuration Example for Keychain Management
- Where to Go Next
- Additional References for Keychain Management

22.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

22.2 Information About Keychain Management

22.2.1 Keychains and Keychain Management

Keychain management allows you to create and maintain keychains, which are sequences of keys (sometimes called shared secrets). You can use keychains with features that secure communications with other devices by using key-based authentication. The device allows you to configure multiple keychains.

Some routing protocols that support key-based authentication can use a keychain to implement a hitless key rollover for authentication. For more information, see the *Inspur CN12700 Series INOS Unicast Routing Configuration Guide*.

22.2.2 Lifetime of a Key

To maintain stable communications, each device that uses a protocol that is secured by key-based authentication must be able to store and use more than one key for a feature at the same time. Based on the send and accept lifetimes of a key, keychain management provides a secure mechanism to handle key rollover. The device uses the lifetimes of keys to determine which keys in a keychain are active.

Each key in a keychain has two lifetimes, as follows:

Accept lifetime

The time interval within which the device accepts the key during a key exchange with another device.

Send lifetime

The time interval within which the device sends the key during a key exchange with another device.

You define the send and accept lifetimes of a key using the following parameters:

Start-time

The absolute time that the lifetime begins.

End-time

The end time can be defined in one of the following ways:

- The absolute time that the lifetime ends
- The number of seconds after the start time that the lifetime ends
- Infinite lifetime (no end-time)

During a key send lifetime, the device sends routing update packets with the key. The device does not accept communication from other devices when the key sent is not within the accept lifetime of the key on the device.

We recommend that you configure key lifetimes that overlap within every keychain. This practice avoids failure of neighbor authentication due to the absence of active keys.

22.3 Virtualization Support for Keychain Management

The following information applies to keychains used in virtual device contexts (VDCs):

- Keychains are unique per VDC. You cannot use a keychain that you created in one VDC in a different VDC.
- Because keychains are not shared by VDCs, you can reuse keychain names in different VDCs.
- The device does not limit keychains on a per-VDC basis.

22.4 Licensing Requirements for Keychain Management

This table shows the licensing requirements for keychain management.

Product	**License Requirement**
Inspur INOS	Keychain management requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

22.5 Prerequisites for Keychain Management

Keychain management has no prerequisites.

22.6 Guidelines and Limitations for Keychain Management

Keychain management has the following configuration guideline and limitation:

- Changing the system clock impacts when the keys are active.

22.7 Default Settings for Keychain Management

This table lists the default settings for Inspur INOS keychain management parameters.

Table 47 : Default Keychain Management Parameters

Parameters	**Default**
Key chains	No keychain exists by default.
Keys	No keys are created by default when you create a new keychain.
Accept lifetime	Always valid.
Send lifetime	Always valid.
Key-string entry encryption	Unencrypted.

22.8 Configuring Keychain Management

22.8.1 Creating a Keychain

You can create a keychain on the device. A new keychain contains no keys.

SUMMARY STEPS

1. **configure terminal**
2. **key chain** *name*
3. (Optional) **show key chain** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**key chain** *name*
	Example:
	`switch(config)# key chain glbp-keys`
	`switch(config-keychain)#`
		Creates the keychain and enters keychain configuration mode.

Step 3	(Optional) **show key chain** *name* **Example:** `switch(config-keychain)# show key chain glbp-keys`	Displays the keychain configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-keychain)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring a Master Key and Enabling the AES Password Encryption Feature.

22.8.2 Removing a Keychain

You can remove a keychain on the device.

Before you begin

If you are removing a keychain, ensure that no feature uses it. If a feature is configured to use a keychain that you remove, that feature is likely to fail to communicate with other devices.

SUMMARY STEPS

1. **configure terminal**
2. **no key chain** *name*
3. (Optional) **show key chain** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**no key chain** *name* **Example:** `switch(config)# no key chain glbp-keys`	Removes the keychain and any keys that the keychain contains.
Step 3	(Optional) **show key chain** *name* **Example:** `switch(config-keychain)# show key chain glbp-keys`	Confirms that the keychain no longer exists in running configuration.
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config-keychain)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Creating a Keychain.

22.8.3 Configuring a Master Key and Enabling the AES Password Encryption Feature

You can configure a master key for type-6 encryption and enable the Advanced Encryption Standard (AES) password encryption feature.

SUMMARY STEPS

1. [**no**] **key config-key ascii**
2. **configure terminal**
3. [**no**] **feature password encryption aes**
4. (Optional) **show encryption service stat**
5. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	[**no**] **key config-key ascii** **Example:** `switch# key config-key ascii New Master Key: Retype Master Key:`	Configures a master key to be used with the AES password encryption feature. The master key can contain between 16 and 32 alphanumeric characters. You can use the **no** form of this command to delete the master key at any time. If you enable the AES password encryption feature before configuring a master key, a message appears stating that password encryption will not take place unless a master key is configured. If a master key is already configured, you are prompted to enter the current master key before entering a new master key.

Step 2	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 3	[**no**] **feature password encryption aes** **Example:** `switch(config)# feature password encryption aes`	Enables or disables the AES password encryption feature.
Step 4	(Optional) **show encryption service stat** **Example:** `switch(config)# show encryption service stat`	Displays the configuration status of the AES password encryption feature and the master key.
Step 5	Required: **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration. **Note** This command is necessary to synchronize the master key in the running configuration and the startup configuration.

Related Topics

Configuring Text for a Key.
Configuring Accept and Send Lifetimes for a Key.

22.8.4 Configuring Text for a Key

You can configure the text for a key. The text is the shared secret. The device stores the text in a secure format.

By default, accept and send lifetimes for a key are infinite, which means that the key is always valid. After you configure the text for a key, configure the accept and send lifetimes for the key.

Before you begin

Determine the text for the key. You can enter the text as unencrypted text or in the encrypted form that Inspur

INOS uses to display key text when you use the **show key chain** command. Using the encrypted form is particularly helpful if you are creating key text to match a key as shown in the **show key chain** command output from another device.

SUMMARY STEPS

1. **configure terminal**
2. **key chain** *name*
3. **key** *key-ID*
4. **key-string** [*encryption-type*] *text-string*
5. (Optional) **show key chain** *name* [**mode decrypt**]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**key chain** *name*
	Example:
	`switch(config)# key chain glbp-`
	`keys switch(config-keychain)#`
		Enters keychain configuration mode for the keychain that you specified.

Step 3	**key** *key-ID*
	Example:
	`switch(config-keychain)# key 13`
	`switch(config-keychain-key)#`
		Enters key configuration mode for the key that you specified. The *key-ID* argument must be a whole number between 0 and 65535.

Step 4	**key-string** [*encryption-type*] *text-string*
	Example:
	`switch(config-keychain-key)# key-string 0`
	`AS3cureStr1ng`
		Configures the text string for the key. The *text-string* argument is alphanumeric, case-sensitive, and supports special characters.
		The *encryption-type* argument can be one of the following values:
		• 0—The *text-string* argument that you enter is unencrypted text. This is the default.
		• 7—The *text-string* argument that you enter is encrypted. The encryption method is a Inspur proprietary method. This option is useful when you are entering a text string based on the encrypted output of a **show key chain** command that you ran on another Inspur INOS device. The value of the first 2 digits of a type 7 key string configured by using the **key-string 7** *text-string* command has to be between 0 and 15. For example, you can configure 07372b557e2c1a as the key string value in

	Command or Action	Purpose
		which case the sum value of the first 2 digits will be 7. But, you cannot configure 85782916342021 as the key string value because the value of the first 2 digits will be 85. We recommend unconfiguring any type 7 key strings that do not adhere to this value or to configure a type 0 string.
Step 5	(Optional) **show key chain** *name* [**mode decrypt**] **Example:** `switch(config-keychain-key)# show key chain glbp-keys`	Shows the keychain configuration, including the key text configuration. The mode decrypt option, which can be used by a device administrator only, displays the keys in cleartext.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config-keychain-key)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring Accept and Send Lifetimes for a Key.

22.8.5 Configuring Accept and Send Lifetimes for a Key

You can configure the accept lifetime and send lifetime for a key. By default, accept and send lifetimes for a key are infinite, which means that the key is always valid.

SUMMARY STEPS

1. **configure terminal**
2. **key chain** *name*
3. **key** *key-ID*
4. **accept-lifetime** [**local**] *start-time* **duration** *duration-value* | **infinite** | *end-time*]
5. **send-lifetime** [**local**] *start-time* **duration** *duration-value* | **infinite** | *end-time*]
6. (Optional) **show key chain** *name* [**mode decrypt**]
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure`	Enters global configuration mode.

	Command or Action	Purpose
	`terminal` `switch(config)#`	

Step 2	**key chain** *name*
	Example:
	`switch(config)# key chain glbp-keys`
	`switch(config-keychain)#`
		Enters keychain configuration mode for the keychain that you specified.

Step 3	**key** *key-ID*
	Example:
	`switch(config-keychain)# key 13`
	`switch(config-keychain-key)#`
		Enters key configuration mode for the key that you specified.

Step 4	**accept-lifetime** [**local**] *start-time* **duration** *duration-value* | **infinite** | *end-time*] **Example:** `switch(config-keychain-key)# accept-lifetime 00:00:00 Jun 13 2008 23:59:59 Sep 12 2008`	Configures an accept lifetime for the key. By default, the device treats the *start-time* and *end-time* arguments as UTC. If you specify the **local** keyword, the device treats these times as local times. The *start-time* argument is the time of day and date that the key becomes active. Specify the end of the lifetime with one of the following options: • **duration** *duration-value* —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years). • **infinite**—The accept lifetime of the key never expires. • *end-time* —The *end-time* argument is the time of day and date that the key becomes inactive.
Step 5	**send-lifetime** [**local**] *start-time* **duration** *duration-value* | **infinite** | *end-time*] **Example:** `switch(config-keychain-key)# send-lifetime 00:00:00 Jun 13 2008 23:59:59 Aug 12 2008`	Configures a send lifetime for the key. By default, the device treats the *start-time* and *end-time* arguments as UTC. If you specify the **local** keyword, the device treats these times as local times. The *start-time* argument is the time of day and date that the key becomes active. You can specify the end of the send lifetime with one of the following options: • **duration** *duration-value* —The length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years). • **infinite—**The send lifetime of the key never expires. • *end-time* —The *end-time* argument is the time of day and date that the key becomes inactive.

	Command or Action	Purpose
Step 6	(Optional) **show key chain** *name* [**mode decrypt**] **Example:** `switch(config-keychain-key)# show key chain glbp-keys`	Shows the keychain configuration, including the key text configuration. The mode decrypt option, which can be used by a device administrator only, displays the keys in cleartext.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config-keychain-key)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Lifetime of a Key.

22.9 Determining Active Key Lifetimes

To determine which keys within a keychain have active accept or send lifetimes, use the command in this table. For detailed information about the fields in the output from this command, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show key chain	Displays the keychains configured on the device.

22.10 Verifying the Keychain Management Configuration

To display keychain management configuration information, perform the following task. For detailed information about the fields in the output from this command, see the *Inspur CN12700 Series INOS Security Command Reference*.

Command	Purpose
show key chain	Displays the keychains configured on the device.

22.11 Configuration Example for Keychain Management

This example shows how to configure a keychain named glbp keys. Each key text string is encrypted. Each key has longer accept lifetimes than send lifetimes, to help prevent lost communications by accidentally configuring a time in which there are no active keys.

```
key chain glbp-keys
  key 0
    key-string 7 zqdest
    accept-lifetime 00:00:00 Jun 01 2008 23:59:59 Sep 12 2008
    send-lifetime 00:00:00 Jun 01 2008 23:59:59 Aug 12 2008
  key 1
    key-string 7 uaeqdyito
    accept-lifetime 00:00:00 Aug 12 2008 23:59:59 Dec 12 2008
    send-lifetime 00:00:00 Sep 12 2008 23:59:59 Nov 12 2008

  key 2
    key-string 7 eekgsdyd
    accept-lifetime 00:00:00 Nov 12 2008 23:59:59 Mar 12 2009
    send-lifetime 00:00:00 Dec 12 2008 23:59:59 Feb 12 2009
```

22.12 Where to Go Next

For information about routing features that use keychains, see the *Inspur CN12700 Series INOS Unicast Routing Configuration Guide*.

22.13 Additional References for Keychain Management

Related Documents

Related Topic	**Document Title**
Gateway Load Balancing Protocol	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*
Border Gateway Protocol	*Inspur CN12700 Series INOS Unicast Routing Configuration Guide*
Keychain management commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Inspur CN12700 Series INOS Security Command Reference*

Standards

Standards	**Title**
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

CHAPTER 23 Configuring Traffic Storm Control

This chapter describes how to configure traffic storm control on the Inspur INOS device. This chapter includes the following sections:

- Finding Feature Information
- Information About Traffic Storm Control
- Virtualization Support for Traffic Storm Control
- Licensing Requirements for Traffic Storm Control
- Guidelines and Limitations for Traffic Storm Control
- Default Settings for Traffic Storm Control
- Configuring Traffic Storm Control
- Verifying Traffic Storm Control Configuration
- Monitoring Traffic Storm Control Counters
- Configuration Example for Traffic Storm Control
- Additional References for Traffic Storm Control
- Feature History for Traffic Storm Control

23.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

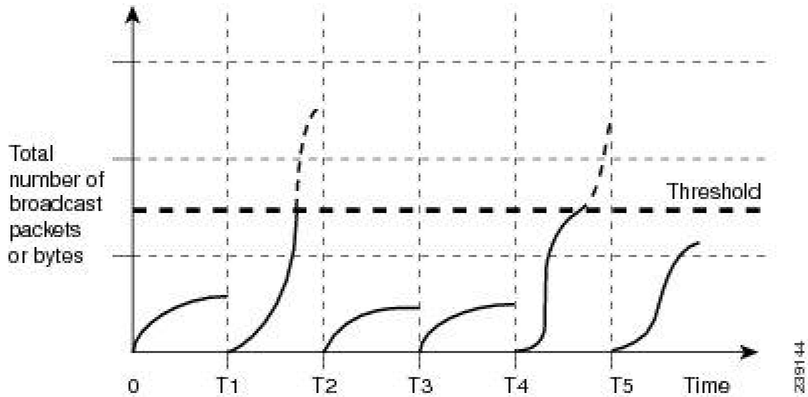
23.2 Information About Traffic Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. You can use the traffic storm control feature to prevent disruptions on Layer 2 ports by a broadcast, multicast, or unicast traffic storm on physical interfaces.

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming broadcast, multicast, and unicast traffic over a 10-millisecond interval. During this interval, the traffic level, which is a percentage of the total available bandwidth of the port, is compared with the traffic storm control level that you configured. When the ingress traffic reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the interval ends.

This table shows the broadcast traffic patterns on a Layer 2 interface over a given interval. In this example, traffic storm control occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

Figure 37 : Broadcast Suppression



The traffic storm control threshold numbers and the time interval allow the traffic storm control algorithm to work with different levels of granularity. A higher threshold allows more packets to pass through.

Traffic storm control on the Inspur INOS device is implemented in the hardware. The traffic storm control circuitry monitors packets that pass from a Layer 2 interface to the switching bus. Using the Individual/Group bit in the packet destination address, the circuitry determines if the packet is unicast or broadcast, tracks the current count of packets within the 10-millisecond interval, and filters out subsequent packets when a threshold is reached.

Traffic storm control uses a bandwidth-based method to measure traffic. You set the percentage of total available bandwidth that the controlled traffic can use. Because packets do not arrive at uniform intervals, the 10-millisecond interval can affect the behavior of traffic storm control.

The following are examples of traffic storm control behavior:

- If you enable broadcast traffic storm control, and broadcast traffic exceeds the level within the 10-millisecond interval, traffic storm control drops all broadcast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and the combined broadcast and multicast traffic exceeds the level within the 10-millisecond interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and broadcast traffic exceeds the level within the 10-millisecond interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and multicast traffic exceeds the level within the 10-millisecond interval, traffic storm control drops all broadcast and multicast traffic until the end of the interval.

By default, the Inspur INOS software takes no corrective action when the traffic exceeds the configured level. However, you can configure an Embedded Event Management (EEM) action to error-disable an interface if the traffic does not subside (drop below the threshold) within a certain time period. For information on configuring EEM, see the *Inspur CN12700 Series INOS System Management Command Reference*.

23.3 Virtualization Support for Traffic Storm Control

Traffic storm control configuration and operation are local to the virtual device context (VDC).

For more information on VDCs, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

23.4 Licensing Requirements for Traffic Storm Control

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS	Traffic storm control requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

23.5 Guidelines and Limitations for Traffic Storm Control

When configuring the traffic storm control level, note the following guidelines and limitations:

- You can configure traffic storm control on a port-channel interface.
- Do not configure traffic storm control on interfaces that are members of a port-channel interface. Configuring traffic storm control on interfaces that are configured as members of a port channel puts the ports into a suspended state.
- When you use the **storm-control unicast level** *percentage* command in a module, both the unknown and known unicast traffic gets discarded after reaching the threshold value.
- Traffic storm control on all Inspur FEX devices connected to Inspur CN12700 series switches has following guidelines and limitations:
- Traffic storm control is not supported on HIF ports.
- Traffic storm control is supported only on NIF ports.
- Specify the level as a percentage of the total interface bandwidth:
- The level can be from 0 to 100.
- The optional fraction of a level can be from 0 to 99.
- 100 percent means no traffic storm control.
- 0.0 percent suppresses all traffic.

Because of hardware limitations and the method by which packets of different sizes are counted, the level percentage is an approximation. Depending on the sizes of the frames that make up the incoming traffic, the actual enforced level might differ from the configured level by several percentage points.

23.6 Default Settings for Traffic Storm Control

This table lists the default settings for traffic storm control parameters.

Table 48： Default Traffic Storm Control Parameters

Parameters	Default
Traffic storm control	Disabled
Threshold percentage	100

23.7 Configuring Traffic Storm Control

You can set the percentage of total available bandwidth that the controlled traffic can use.

SUMMARY STEPS

1. **configure terminal**
2. **interface** {**ethernet** *slot*/*port* | **port-channel** *number*}
3. **storm-control** {**broadcast** | **multicast** | **unicast**} **level***percentage*[.*fraction*]
4. **exit**
5. (Optional) **show running-config interface** {**ethernet** *slot*/*port* | **port-channel** *number*}

6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** {**ethernet** *slot*/*port* | **port-channel** *number*}
	Example:
	`switch# interface ethernet 1/1`
	`switch(config-if)#`	

Step 3	Enters interface configuration mode.	Configures traffic storm control for traffic on the interface. The default state is disabled. **Note** The **storm-control unicast** command configures traffic storm control for all the unicast packets.

Step 4	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

Step 5	(Optional) **show running-config interface** {**ethernet** *slot*/*port* | **port-channel** *number*} **Example:** `switch(config)# show running-config interface ethernet 1/1`	Displays the traffic storm control configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

23.8 Verifying Traffic Storm Control Configuration

To display traffic storm control configuration information, perform one of the following tasks:

Command	Purpose
show interface [**ethernet** *slot*/*port* | **port-channel** *number*] **counters storm-control**	Displays the traffic storm control configuration for the interfaces.
show running-config interface	Displays the traffic storm control configuration.

For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

23.9 Monitoring Traffic Storm Control Counters

You can monitor the counters the Inspur INOS device maintains for traffic storm control activity.

SUMMARY STEPS

1. **show interface** [**ethernet** *slot/port* | **port-channel** *number*] **counters storm-control**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show interface** [**ethernet** *slot/port* | **port-channel** *number*] **counters storm-control** **Example:** `switch# show interface counters storm-control`	Displays the traffic storm control counters.

23.10 Configuration Example for Traffic Storm Control

The following example shows how to configure traffic storm control:

```
interface Ethernet1/1
  storm-control broadcast level 40
  storm-control multicast level 40
  storm-control unicast level 40
```

23.11 Additional References for Traffic Storm Control

This section includes additional information related to implementing traffic storm control.

Related Documents

Related Topic	Document Title
Inspur INOS Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*

23.12 Feature History for Traffic Storm Control

This table lists the release history for this feature.

Table 49： Feature History for Traffic Storm Control

Feature Name	Releases	Feature Information	
Traffic storm control	8.2(3)	No change from Release 8.2(3)	
Traffic storm control	8.2(3)	No change from Release 8.2(3)	

CHAPTER 24 Configuring Unicast RPF

This chapter describes how to configure rate limits for egress traffic on Inspur INOS devices. This chapter includes the following sections:

- Finding Feature Information
- Information About Unicast RPF
- Virtualization Support for Unicast RPF
- Licensing Requirements for Unicast RPF
- Guidelines and Limitations for Unicast RPF
- Default Settings for Unicast RPF
- Configuring Unicast RPF
- Configuration Examples for Unicast RPF
- Verifying the Unicast RPF Configuration
- Additional References for Unicast RPF
- Feature History for Unicast RPF

24.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

24.2 Information About Unicast RPF

The Unicast RPF feature reduces problems that are caused by the introduction of malformed or forged (spoofed) IPv4 or IPv6 source addresses into a network by discarding IPv4 or IPv6 packets that lack a verifiable IP source address. For example, a number of common types of Denial-of-Service (DoS) attacks, including Smurf and Tribal Flood Network (TFN) attacks, can take advantage of forged or rapidly changing source IPv4 or IPv6 addresses to allow attackers to thwart efforts to locate or filter the attacks. Unicast RPF deflects attacks by forwarding only the packets that have source addresses that are valid and consistent with the IP routing table.

When you enable Unicast RPF on an interface, the device examines all ingress packets received on that interface to ensure that the source address and source interface appear in the routing table and match the interface on which the packet was received. This examination of source addresses relies on the Forwarding Information Base (FIB).

Unicast RPF verifies that any packet received at a device interface arrives on the best return path (return route) to the source of the packet by doing a reverse lookup in the FIB. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. If there is no reverse path route on the same interface from which the packet was received, the source address might have been modified by the attacker. If Unicast RPF does not find a reverse path for the packet, the packet is dropped.

24.2.1 Unicast RPF Process

Unicast RPF has several key implementation principles:

- The packet must be received at an interface that has the best return path (route) to the packet source (a process called *symmetric routing*). There must be a route in the FIB that matches the route to the receiving interface. Static routes, network statements, and dynamic routing add routes to the FIB.
- IP source addresses at the receiving interface must match the routing entry for the interface.
- Unicast RPF is an input function and is applied only on the input interface of a device at the upstream end of a connection.

You can use Unicast RPF for downstream networks, even if the downstream network has other connections to the Internet.

Caution	Be careful when using optional BGP attributes, such as weight and local preference, because an attacker can modify the best path back to the source address. Modification would affect the operation of Unicast RPF.

When a packet is received at the interface where you have configured Unicast RPF and ACLs, the Inspur INOS software performs the following actions:

SUMMARY STEPS

1. Checks the input ACLs on the inbound interface.
2. Uses Unicast RPF to verify that the packet has arrived on the best return path to the source, which it does by doing a reverse lookup in the FIB table.
3. Conducts a FIB lookup for packet forwarding.
4. Checks the output ACLs on the outbound interface.
5. Forwards the packet.

DETAILED STEPS

Step 1 Checks the input ACLs on the inbound interface.

Step 2 Uses Unicast RPF to verify that the packet has arrived on the best return path to the source, which it does by doing a reverse lookup in the FIB table.

Step 3 Conducts a FIB lookup for packet forwarding.

Step 4 Checks the output ACLs on the outbound interface.

Step 5 Forwards the packet.

24.2.2 Global Statistics

Each time the Inspur INOS device drops a packet at an interface due to a failed unicast RPF check, that information is counted globally on the device on a per-forwarding engine (FE) basis. Global statistics on dropped packets provide information about potential attacks on the network, but they do not specify which interface is the source of the attack. Per-interface statistics on packets dropped due to a failed unicast RPF check are not available.

24.3 Virtualization Support for Unicast RPF

Unicast RPF configuration and operation is local to the virtual device context (VDC). For more information on VDCs, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

24.4 Licensing Requirements for Unicast RPF

Product	License Requirement
Inspur INOS	Unicast RPF requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

24.5 Guidelines and Limitations for Unicast RPF

Unicast RPF has the following configuration guidelines and limitations:

- You must apply Unicast RPF at the interface downstream from the larger portion of the network, preferably at the edges of your network.
- The further downstream that you apply Unicast RPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying Unicast RPF on an aggregation device helps to mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying Unicast RPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying Unicast RPF across many sites does add to the administration cost of operating the network.
- The more entities that deploy Unicast RPF across Internet, intranet, and extranet resources, means that the better the chances are of mitigating large-scale network disruptions throughout the Internet community, and the better the chances are of tracing the source of an attack.
- Unicast RPF will not inspect IP packets that are encapsulated in tunnels, such as generic routing encapsulation (GRE) tunnels. You must configure Unicast RPF at a home gateway so that Unicast RPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.
- You can use Unicast RPF in any “single-homed” environment where there is only one access point out of the network or one upstream connection. Networks that have one access point provide symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet.
- Do not use Unicast RPF on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry, which means that multiple routes to the source of a packet exist. You should configure Unicast RPF only where there is natural or configured symmetry. Do not configure strict Unicast RPF.
- Unicast RPF allows packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that the Bootstrap Protocol (BOOTP) and the Dynamic Host Configuration Protocol (DHCP) can operate correctly.

24.6 Default Settings for Unicast RPF

This table lists the default settings for Unicast RPF parameters.

Table 50：Default Unicast RPF Parameter Settings

Parameters	**Default**
Unicast RPF	Disabled

24.7 Configuring Unicast RPF

You can configure one the following Unicast RPF modes on an ingress interface:

Strict Unicast RPF mode

A strict mode check is successful when Unicast RFP finds a match in the FIB for the packet source address and the ingress interface through which the packet is received matches one of the Unicast RPF interfaces in the FIB match. If this check fails, the packet is discarded. You can use this type of Unicast RPF check where packet flows are expected to be symmetrical.

Loose Unicast RPF mode

A loose mode check is successful when a lookup of a packet source address in the FIB returns a match and the FIB result indicates that the source is reachable through at least one real interface. The ingress interface through which the packet is received is not required to match any of the interfaces in the FIB result.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot*/*port*
3. **ip verify unicast source reachable-via {any [allow-default] | rx}**
4. **ipv6 verify unicast source reachable-via** {**any** [**allow-default**] | **rx**}
5. **exit**
6. (Optional) **show ip interface ethernet** *slot*/*port*
7. (Optional) **show running-config interface ethernet** *slot*/*port*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface ethernet** *slot*/*port*
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		Specifies an Ethernet interface and enters interface configuration mode.

Step 3	**ip verify unicast source reachable-via** {**any** [**allow-default**] | **rx**} **Example:** `switch(config-if)# ip verify unicast source reachable-via any`	Configures Unicast RPF on the interface for IPv4. The **any** keyword specifies loose Unicast RPF. If you specify the **allow-default** keyword, the source address lookup can match the default route and use that for verification. The **rx** keyword specifies strict Unicast RPF.
Step 4	**ipv6 verify unicast source reachable-via** {**any** [**allow-default**] | **rx**} **Example:** `switch(config-if)# ipv6 verify unicast source reachable-via any`	Configures Unicast RPF on the interface for IPv6. The **any** keyword specifies loose Unicast RPF. If you specify the **allow-default** keyword, the source address lookup can match the default route and use that for verification. The **rx** keyword specifies strict Unicast RPF.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface configuration mode.

	Command or Action	Purpose
Step 6	(Optional) **show ip interface ethernet** *slot/port* **Example:** `switch(config)# show ip interface ethernet 2/3`	Displays the IP information for an interface.
Step 7	(Optional) **show running-config interface ethernet** *slot/port* **Example:** `switch(config)# show running-config interface ethernet 2/3`	Displays the configuration for an interface in the running configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

24.8 Configuration Examples for Unicast RPF

The following example shows how to configure loose Unicast RFP for IPv4 packets:

```
interface Ethernet2/3
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via any
```

The following example shows how to configure strict Unicast RFP for IPv4 packets:

```
interface Ethernet2/2
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via rx
```

The following example shows how to configure loose Unicast RFP for IPv6 packets:

```
interface Ethernet2/1
  ipv6 address 2001:0DB8:c18:1::3/64
  ipv6 verify unicast source reachable-via any
```

The following example shows how to configure strict Unicast RFP for IPv6 packets:

```
interface Ethernet2/4
  ipv6 address 2001:0DB8:c18:1::3/64
  ipv6 verify unicast source reachable-via rx
```

24.9 Verifying the Unicast RPF Configuration

To display Unicast RPF configuration information, perform one of the following tasks:

Command	Purpose
show running-config interface ethernet *slot/port*	Displays the interface configuration in the running configuration.
show running-config ip [**all**]	Displays the IPv4 configuration in the running configuration.

show running-config ipv6 [**all**]	Displays the IPv6 configuration in the running configuration.
show startup-config interface ethernet *slot/port*	Displays the interface configuration in the startup configuration.
show startup-config ip	Displays the IP configuration in the startup configuration.

For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

24.10 Additional References for Unicast RPF

This section includes additional information related to implementing Unicast RPF.

Related Documents

Related Topic	Document Title
Inspur INOS Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*

24.11 Feature History for Unicast RPF

This table lists the release history for this feature.

Table 51： Feature History for Unicast RPF

Feature Name	Releases	Feature Information	
Unicast RPF	8.2(3)	No change from Release 8.2(3).	
Unicast RPF	8.2(3)	No change from Release 8.2(3).	

CHAPTER 25 Configuring Control Plane Policing

This chapter contains the following sections:

- Finding Feature Information
- Information About CoPP
- Licensing Requirements for CoPP
- Guidelines and Limitations for CoPP
- Default Settings for CoPP
- Configuring CoPP
- Verifying the CoPP Configuration
- Displaying the CoPP Configuration Status
- Monitoring CoPP
- Monitoring CoPP with SNMP
- Clearing the CoPP Statistics
- Configuration Examples for CoPP
- Changing or Reapplying the Default CoPP Policy Using the Setup Utility
- Additional References for CoPP
- Feature History for CoPP

25.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

25.2 Information About CoPP

Control Plane Policing (CoPP) protects the control plane and separates it from the data plane, which ensures network stability, reachability, and packet delivery.

This feature allows a policy map to be applied to the control plane. This policy map looks like a normal QoS policy and is applied to all traffic destined to any of the IP addresses of the router or Layer 3 switch. A common attack vector for network devices is the denial-of-service (DoS) attack, where excessive traffic is directed at the device interfaces.

The Inspur INOS device provides CoPP to prevent DoS attacks from impacting performance. Such attacks, which can be perpetrated either inadvertently or maliciously, typically involve high rates of traffic destined to the supervisor module or CPU itself.

The supervisor module divides the traffic that it manages into three functional components or planes:

Data plane

Handles all the data traffic. The basic functionality of a Inspur INOS device is to forward packets from one interface to another. The packets that are not meant for the switch itself are called the transit packets. These packets are handled by the data plane.

Control plane

Handles all routing protocol control traffic. These protocols, such as the Border Gateway Protocol (BGP) and the Open Shortest Path First (OSPF) Protocol, send control packets between devices. These packets are destined to router addresses and are called control plane packets.

Management plane

Runs the components meant for Inspur INOS device management purposes such as the command-line interface (CLI) and Simple Network Management Protocol (SNMP).

The supervisor module has both the management plane and control plane and is critical to the operation of the network. Any disruption or attacks to the supervisor module will result in serious network outages. For example, excessive traffic to the supervisor module could overload and slow down the performance of the entire Inspur INOS device. Another example is a DoS attack on the supervisor module that could generate IP traffic streams to the control plane at a very high rate, forcing the control plane to spend a large amount of time in handling these packets and preventing the control plane from processing genuine traffic.

Examples of DoS attacks are as follows:

- Internet Control Message Protocol (ICMP) echo requests
- IP fragments
- TCP SYN flooding

These attacks can impact the device performance and have the following negative effects:

- Reduced service quality (such as poor voice, video, or critical applications traffic)
- High route processor or switch processor CPU utilization
- Route flaps due to loss of routing protocol updates or keepalives
- Unstable Layer 2 topology
- Slow or unresponsive interactive sessions with the CLI
- Processor resource exhaustion, such as the memory and buffers
- Indiscriminate drops of incoming packets

Caution	It is important to ensure that you protect the supervisor module from accidental or malicious attacks by configuring control plane protection.

25.2.1 Control Plane Protection

To protect the control plane, the Inspur INOS device segregates different packets destined for the control plane into different classes. Once these classes are identified, the Inspur INOS device polices the packets, which ensures that the supervisor module is not overwhelmed.

Control Plane Packet Types

Different types of packets can reach the control plane:

Receive packets

Packets that have the destination address of a router. The destination address can be a Layer 2 address (such as a router MAC address) or a Layer 3 address (such as the IP address of a router interface). These packets include router updates and keepalive messages. Multicast packets can also be in this category where packets are sent to multicast addresses that are used by a router.

Exception packets

Packets that need special handling by the supervisor module. For example, if a destination address is not present in the Forwarding Information Base (FIB) and results in a miss, the supervisor module sends an ICMP unreachable packet back to the sender. Another example is a packet with IP options set.

Redirected packets

Packets that are redirected to the supervisor module. Features such as Dynamic Host Configuration Protocol

(DHCP) snooping or dynamic Address Resolution Protocol (ARP) inspection redirect some packets to the supervisor module.

Glean packets

If a Layer 2 MAC address for a destination IP address is not present in the FIB, the supervisor module receives the packet and sends an ARP request to the host.

All of these different packets could be maliciously used to attack the control plane and overwhelm the Inspur INOS device. CoPP classifies these packets to different classes and provides a mechanism to individually control the rate at which the supervisor module receives these packets.

Classification for CoPP

For effective protection, the Inspur INOS device classifies the packets that reach the supervisor modules to allow you to apply different rate controlling policies based on the type of the packet. For example, you might want to be less strict with a protocol packet such as Hello messages but more strict with a packet that is sent to the supervisor module because the IP option is set.

Rate Controlling Mechanisms

Once the packets are classified, the Inspur INOS device has different mechanisms to control the rate at which packets arrive at the supervisor module. Two mechanisms control the rate of traffic to the supervisor module. One is called policing and the other is called rate limiting.

Using hardware policers, you can define separate actions for traffic that conforms to, exceeds, or violates certain conditions. The actions can transmit the packet, mark down the packet, or drop the packet.

You can configure the following parameters for policing:

Committed information rate (CIR)

Desired bandwidth, specified as a bit rate or a percentage of the link rate.

Peak information rate (PIR)

Desired bandwidth, specified as a bit rate or a percentage of the link rate.

Committed burst (BC)

Size of a traffic burst that can exceed the CIR within a given unit of time and not impact scheduling.

Extended burst (BE)

Size that a traffic burst can reach before all traffic exceeds the PIR.

In addition, you can set separate actions such as transmit or drop for conform, exceed, and violate traffic.

For more information on policing parameters, see the *Inspur CN12700 Series INOS Quality of Service Configuration Guide*.

Default Policing Policies

When you bring up your Inspur INOS device for the first time, the Inspur INOS software installs the default copp-system-p-policy-strict policy to protect the supervisor module from DoS attacks. You can set the level of protection by choosing one of the following CoPP policy options from the initial setup utility:

- Strict—This policy is 1 rate and 2 color and has a BC value of 250 ms (except for the important class, which has a value of 1000 ms).
- Moderate—This policy is 1 rate and 2 color and has a BC value of 310 ms (except for the important class, which has a value of 1250 ms). These values are 25 percent greater than the strict policy.
- Lenient—This policy is 1 rate and 2 color and has a BC value of 375 ms (except for the important class, which has a value of 1500 ms). These values are 50 percent greater than the strict policy.

- Dense—This policy is 1 rate and 2 color. The classes critical, normal, redirect, exception, undesirable, l2-default, and default have a BC value of 250 ms. The classes important, management, normal-dhcp, normal-dhcp-relay-response, and monitoring have a BC value of 1000 ms. The class l2-unpoliced has a BC value of 5 MB.
- Skip—No control plane policy is applied. In Inspur INOS releases prior to 8.2(3), this option is named none.

If you do not select an option or choose not to execute the setup utility, the Inspur INOS software applies strict policing. We recommend that you start with the strict policy and later modify the CoPP policies as required.

The copp-system-p-policy policy has optimized values suitable for basic device operations. You must add specific class and access-control list (ACL) rules that meet your DoS protection requirements. The default CoPP policy does not change when you upgrade the Inspur INOS software.

Caution	Selecting the **skip** option and not subsequently configuring CoPP protection can leave your Inspur INOS device vulnerable to DoS attacks.

You can reassign the CoPP default policy by entering the setup utility again using the **setup** command from the CLI prompt or by using the **copp profile** command in Inspur INOS Release 8.2(3) or later releases.

Related Topics

Changing or Reapplying the Default CoPP Policy.

Default Class Maps

The copp-system-class-exception class has the following configuration:

```
class-map type control-plane match-any copp-system-class-exception
    match exception ip option
    match exception ip icmp unreachable
    match exception ipv6 option
    match exception ipv6 icmp unreachable
```

The copp-system-class-critical class has the following configuration:

```
ip access-list copp-system-acl-igmp
    permit igmp any 224.0.0.0/3

ip access-list copp-system-p-acl-lisp
    permit udp any any eq 4342

ip access-list copp-system-acl-msdp
    permit tcp any gt 1024 any eq 639
    permit tcp any eq 639 any gt 1024

ip access-list copp-system-acl-bgp
    permit tcp any gt 1024 any eq bgp
    permit tcp any eq bgp any gt 1024

ip access-list copp-system-acl-eigrp
    permit eigrp any any

ip access-list copp-system-p-acl-lisp6
    permit udp any any eq 4342

ip access-list copp-system-acl-rip permit
    udp any 224.0.0.0/24 eq rip

ip access-list copp-system-acl-ospf
```

```
    permit ospf any any

ip access-list copp-system-acl-pim
    permit pim any 224.0.0.0/24




ipv6 access-list copp-system-acl-bgp6
    permit tcp any gt 1024 any eq bgp
    permit tcp any eq bgp any gt 1024

ipv6 access-list copp-system-acl-ospf6
    permit 89 any any
ipv6 access-list copp-system-acl-pim6
    permit 103 any FF02::D/128
    permit udp any any eq pim-auto-rp



ip access-list copp-system-acl-vpc
    permit udp any any eq 3200


mac access-list copp-system-acl-mac-fabricpath-isis
  permit any 0180.c200.0041 0000.0000.0000

mac access-list copp-system-p-acl-mac-l3-isis
    permit any 0180.c200.0015 0000.0000.0000
    permit any 0180.c200.0014.0000.0000.0000

class-map type control-plane match-any copp-system-class-critical
    match access-group name copp-system-acl-bgp
    match access-group name copp-system-acl-rip
    match access-group name copp-system-acl-vpc
    match access-group name copp-system-acl-bgp6
    match access-group name copp-system-p-acl-lisp
    match access-group name copp-system-acl-ospf

    match access-group name copp-system-acl-eigrp
    match access-group name copp-system-p-acl-lisp6
    match access-group name copp-system-acl-ospf6
    match access-group name copp-system-acl-eigrp6





    match access-group name copp-system-p-acl-mac-l3-isis
```

The copp-system-class-important class has the following configuration:

```
ip access-list copp-system-p-acl-hsrp
    permit udp any 224.0.0.2/32 eq 1985
    permit udp any 224.0.0.102/32 eq 1985


   permit udp any 224.0.0.0/24 eq 1985


ip access-list copp-system-acl-vrrp



ip access-list copp-system-acl-glbp
```

```
    permit udp any eq 3222 224.0.0.0/24 eq 3222

ip access-list copp-system-acl-pim-reg
    permit pim any any

ipv6 access-list copp-system-acl-icmp6-msgs
    permit icmp any any router-advertisement
    permit icmp any any router-solicitation
    permit icmp any any nd-na
    permit icmp any any nd-ns
    permit icmp any any mld-query
    permit icmp any any mld-report
    permit icmp any any mld-reduction
    permit icmp any any 143

ip access-list copp-system-acl-cts
    permit tcp any any eq 64999
    permit tcp any eq 64999 any



ipv6 access-list copp-system-p-acl-vrrp6
    permit ipv6 any ff02::12/128

ip access-list copp-system-acl-wccp





class-map type control-plane match-any copp-system-class-important
    match access-group name copp-system-acl-cts
    match access-group name copp-system-acl-glbp
    match access-group name copp-system-acl-hsrp
    match access-group name copp-system-acl-vrrp
    match access-group name copp-system-acl-wccp

    match access-group name copp-system-p-acl-vrrp6


ip access-list copp-system-p-acl-igmp
  permit igmp any 224.0.0.0/3
ipv6 access-list copp-system-p-acl-mld
  permit icmp any any mld-query
  permit icmp any any mld-report
  permit icmp any any mld-reduction
  permit icmp any any 143
ip access-list copp-system-p-acl-msdp
  permit tcp any gt 1024 any eq 639
  permit tcp any eq 639 any gt 1024
ipv6 access-list copp-system-p-acl-ndp
  permit icmp any any router-solicitation
  permit icmp any any router-advertisement
  permit icmp any any 137
  permit icmp any any nd-ns
  permit icmp any any nd-na
ip access-list copp-system-p-acl-pim
  permit pim any 224.0.0.0/24
  permit udp any any eq 496
  permit ip any 224.0.0.13/32
ip access-list copp-system-p-acl-pim-mdt-join
  permit udp any 224.0.0.13/32
ip access-list copp-system-p-acl-pim-reg
  permit pim any any
```

```
ipv6 access-list copp-system-p-acl-pim6
  permit pim any ff02::d/128
  permit udp any any eq 496
ipv6 access-list copp-system-p-acl-pim6-reg
  permit pim any any
mac access-list copp-system-p-acl-mac-dot1x
  permit any 0180.c200.0003 0000.0000.0000 0x888e
class-map type control-plane match-any copp-system-p-class-multicast-host
  match access-group name copp-system-p-acl-mld
  match access-group name copp-system-p-acl-igmp
class-map type control-plane match-any copp-system-p-class-multicast-router
  match access-group name copp-system-p-acl-pim
  match access-group name copp-system-p-acl-msdp
  match access-group name copp-system-p-acl-pim6
  match access-group name copp-system-p-acl-pim-reg
  match access-group name copp-system-p-acl-pim6-reg
  match access-group name copp-system-p-acl-pim-mdt-join
class-map type control-plane match-any copp-system-p-class-ndp
  match access-group name copp-system-p-acl-ndp
```

The copp-system-class-management class has the following configuration:

```
ip access-list copp-system-acl-tacacs
    permit tcp any any eq tacacs
    permit tcp any eq tacacs any


ip access-list copp-system-acl-radius
    permit udp any any eq 1812
    permit udp any any eq 1813
    permit udp any any eq 1645
    permit udp any any eq 1646
    permit udp any eq 1812 any
    permit udp any eq 1813 any
    permit udp any eq 1645 any
    permit udp any eq 1646 any

ip access-list copp-system-acl-ntp
    permit udp any any eq ntp

ip access-list copp-system-acl-ftp
    permit tcp any any eq ftp-data
    permit tcp any any eq ftp
    permit tcp any eq ftp-data any
    permit tcp any eq ftp any

ip access-list copp-system-acl-tftp
    permit udp any any eq tftp
    permit udp any any eq 1758
    permit udp any eq tftp any
    permit udp any eq 1758 any

ip access-list copp-system-acl-sftp
    permit tcp any any eq 115
    permit tcp any eq 115 any

ip access-list copp-system-acl-ssh
    permit tcp any any eq 22
    permit tcp any eq 22 any

ip access-list copp-system-acl-snmp
    permit udp any any eq snmp
    permit udp any any eq snmptrap
```

```
ip access-list copp-system-acl-telnet
    permit tcp any any eq telnet
    permit tcp any any eq 107
    permit tcp any eq telnet any
    permit tcp any eq 107 any

ipv6 access-list copp-system-acl-tacacs6
    permit tcp any any eq tacacs
    permit tcp any eq tacacs any

ipv6 access-list copp-system-acl-radius6
    permit udp any any eq 1812
    permit udp any any eq 1813
    permit udp any any eq 1645
    permit udp any any eq 1646
    permit udp any eq 1812 any
    permit udp any eq 1813 any
    permit udp any eq 1645 any
    permit udp any eq 1646 any

ipv6 access-list copp-system-acl-ntp6
    permit udp any any eq ntp
    permit udp any eq ntp any

ipv6 access-list copp-system-acl-tftp6
    permit udp any any eq tftp


    permit udp any any eq 1758
    permit udp any eq tftp any
    permit udp any eq 1758 any

ipv6 access-list copp-system-acl-ssh6
    permit tcp any any eq 22
    permit tcp any eq 22 any

ipv6 access-list copp-system-acl-telnet6
    permit tcp any any eq telnet
    permit tcp any any eq 107
    permit tcp any eq telnet any
    permit tcp any eq 107 any

class-map type control-plane match-any copp-system-class-management
    match access-group name copp-system-acl-tacacs
    match access-group name copp-system-acl-radius
    match access-group name copp-system-acl-ntp
    match access-group name copp-system-acl-ftp
    match access-group name copp-system-acl-tftp
    match access-group name copp-system-acl-sftp
    match access-group name copp-system-acl-ssh
    match access-group name copp-system-acl-snmp
    match access-group name copp-system-acl-telnet
    match access-group name copp-system-acl-tacacs6
    match access-group name copp-system-acl-radius6
    match access-group name copp-system-acl-ntp6
    match access-group name copp-system-acl-tftp6
    match access-group name copp-system-acl-ssh6
    match access-group name copp-system-acl-telnet6
```

The copp-system-class-normal class has the following configuration:

```
class-map type control-plane match-any copp-system-class-normal

   match exception multicast directly-connected-sources
   match protocol arp
```

The copp-system-class-redirect class has the following configuration:

```
class-map type control-plane match-any copp-system-class-redirect
    match redirect arp-inspect
```

The copp-system-class-monitoring class has the following configuration:

```
ip access-list copp-system-acl-icmp
    permit icmp any any echo
    permit icmp any any echo-reply

ip access-list copp-system-acl-traceroute
    permit icmp any any ttl-exceeded
    permit icmp any any port-unreachable


ipv6 access-list copp-system-acl-icmp6
    permit icmp any any echo-request
    permit icmp any any echo-reply

class-map type control-plane match-any copp-system-class-monitoring
    match access-group name copp-system-acl-icmp
    match access-group name copp-system-acl-traceroute
    match access-group name copp-system-acl-icmp6
```

```
mac access-list copp-system-p-acl-mac-l2-tunnel
    permit any any 0x8840


    match access-group name copp-system-p-acl-mac-l2-tunnel
```

The copp-system-class-fcoe class has the following configuration:

```
mac access-list copp-system-p-acl-mac-fcoe
    permit any any 0x8906
    permit any any 0x8914

class-map type control-plane match-any copp-system-p-class-fcoe
    match access-group name copp-system-p-acl-mac-fcoe
```

The copp-system-class-undesirable class has the following configuration:

```
ip access-list copp-system-acl-undesirable
    permit udp any any eq 1434
```

```
class-map type control-plane match-any copp-system-class-undesirable
    match access-group name copp-system-acl-undesirable
    match exception fcoe-fib-miss


mac access-list copp-system-acl-mac-cdp-udld-vtp
  permit any 0100.0ccc.cccc 0000.0000.0000
mac access-list copp-system-acl-mac-cfsoe
  permit any 0180.c200.000e 0000.0000.0000 0x8843
mac access-list copp-system-acl-mac-dot1x

  permit any 0180.c200.0003 0000.0000.0000 0x888e
mac access-list copp-system-acl-mac-flow-control
  permit any 0180.c200.0001 0000.0000.0000 0x8808
mac access-list copp-system-acl-mac-l2mp-isis
  permit any 0180.c200.0015 0000.0000.0000
  permit any 0180.c200.0014 0000.0000.0000
mac access-list copp-system-acl-mac-l2pt
  permit any 0100.0ccd.cdd0 0000.0000.0000
mac access-list copp-system-acl-mac-lacp
  permit any 0180.c200.0002 0000.0000.0000 0x8809
mac access-list copp-system-acl-mac-lldp
  permit any 0180.c200.000e 0000.0000.0000 0x88c
mac access-list copp-system-acl-mac-stp
  permit any 0100.0ccc.cccd 0000.0000.0000
  permit any 0180.c200.0000 0000.0000.0000
mac access-list copp-system-acl-mac-undesirable
  permit any any
```

Strict Default CoPP Policy

The strict CoPP policy has the following configuration in Inspur INOS Release 8.2(3):

```
policy-map type control-plane copp-system-policy

    class copp-system-class-critical

        police cir 36000 kbps bc 250 ms conform transmit violate drop

    class copp-system-class-important

        police cir 1400 kbps bc 1500 ms conform transmit violate drop

    class copp-system-p-class-multicast-router
        set cos 6
        police cir 2600 kbps bc 1000 ms conform transmit violate drop

    class copp-system-class-management

        police cir 10000 kbps bc 250 ms conform transmit violate drop

    class copp-system-p-class-multicast-host
        set cos 1
        police cir 1000 kbps bc 1000 ms conform transmit violate drop

    class copp-system-class-normal

        police cir 680 kbps bc 250 ms conform transmit violate drop

    class copp-system-p-class-ndp
        set cos 6
        police cir 680 kbps bc 250 ms conform transmit violate drop
```

```
        class copp-system-class-redirect

            police cir 280 kbps bc 250 ms conform transmit violate drop

        class copp-system-class-exception

            police cir 360 kbps bc 250 ms conform transmit violate drop

        class copp-system-class-monitoring

            police cir 130 kbps bc 1000 ms conform transmit violate drop



        class copp-system-class-undesirable

            police cir 32 kbps bc 250 ms conform drop violate drop

        class copp-system-p-class-fcoe
            set cos 6
            police cir 1060 kbps bc 1000 ms conform transmit violate drop



        class class-default

            police cir 10 kbps bc 250 ms conform transmit violate drop
```

Moderate Default CoPP Policy

The moderate CoPP policy has the following configuration in Inspur INOS Release 8.2(3):

```
    policy-map type control-plane copp-system-policy-moderate

        class copp-system-class-critical

            police cir 36000 kbps bc 310 ms conform transmit violate drop

        class copp-system-class-important

            police cir 1400 kbps bc 1250 ms conform transmit violate drop

        class copp-system-p-class-multicast-router
            set cos 6
            police cir 2600 kbps bc 1000 ms conform transmit violate drop

        class copp-system-class-management

            police cir 10000 kbps bc 310 ms conform transmit violate drop

        class copp-system-p-class-multicast-host
            set cos 1
            police cir 1000 kbps bc 1000 ms conform transmit violate drop

        class copp-system-class-normal

            police cir 680 kbps bc 310 ms conform transmit violate drop

        class copp-system-p-class-ndp
            set cos 6
            police cir 680 kbps bc 310 ms conform transmit violate drop
```

```
    class copp-system-class-redirect

        police cir 280 kbps bc 310 ms conform transmit violate drop

    class copp-system-class-exception

        police cir 360 kbps bc 310 ms conform transmit violate drop

    class copp-system-class-monitoring

        police cir 130 kbps bc 1250 ms conform transmit violate drop


    class class-default

        police cir 10 kbps bc 250 ms conform transmit violate drop
```

Lenient Default CoPP Policy

The lenient CoPP policy has the following configuration in Inspur INOS Release 8.2(3):

```
policy-map type control-plane copp-system-policy-lenient

    class copp-system-class-critical

        police cir 36000 kbps bc 375 ms conform transmit violate drop

    class copp-system-class-important

        police cir 1400 kbps bc 1500 ms conform transmit violate drop

    class copp-system-p-class-multicast-router
        set cos 6
        police cir 2600 kbps bc 1000 ms conform transmit violate drop

    class copp-system-class-management

        police cir 10000 kbps bc 375 ms conform transmit violate drop

    class copp-system-p-class-multicast-host
        set cos 1
        police cir 1000 kbps bc 1000 ms conform transmit violate drop

    class copp-system-class-normal

        police cir 680 kbps bc 375 ms conform transmit violate drop

    class copp-system-p-class-ndp
        set cos 6
        police cir 680 kbps bc 375 ms conform transmit violate drop


    class copp-system-class-redirect

        police cir 280 kbps bc 375 ms conform transmit violate drop

    class copp-system-class-exception

        police cir 360 kbps bc 375 ms conform transmit violate drop

    class copp-system-class-monitoring
```

```
        police cir 130 kbps bc 1500 ms conform transmit violate drop


    class copp-system-p-class-fcoe
        set cos 6
        police cir 1060 kbps bc 1500 ms conform transmit violate drop

    class copp-system-class-l2-default
        police cir 10 kbps bc 375 ms conform transmit violate drop

    class class-default

        police cir 10 kbps bc 250 ms conform transmit violate drop
```

Dense Default CoPP Policy

The dense CoPP policy has the following configuration in Inspur INOS Release 8.2(3):

```
policy-map type control-plane copp-system-p-policy-dense
  class copp-system-p-class-critical
    set cos 7
    police cir 4500 kbps bc 250 ms conform transmit violate drop
  class copp-system-p-class-important
    set cos 6
    police cir 1400 kbps bc 1500 ms conform transmit violate drop
  class copp-system-p-class-multicast-router
    set cos 6
    police cir 370 kbps bc 1000 ms conform transmit violate drop
  class copp-system-p-class-management
    set cos 2
    police cir 2500 kbps bc 1000 ms conform transmit violate drop
  class copp-system-p-class-multicast-host
    set cos 1
    police cir 190 kbps bc 1000 ms conform transmit violate drop
  class copp-system-p-class-normal
    set cos 1
    police cir 300 kbps bc 250 ms conform transmit violate drop
  class copp-system-p-class-ndp
    set cos 6
    police cir 300 kbps bc 250 ms conform transmit violate drop
  class copp-system-p-class-normal-dhcp


    set cos 1
    police cir 660 kbps bc 1000 ms conform transmit violate drop
  class copp-system-p-class-normal-dhcp-relay-response
    set cos 1
    police cir 800 kbps bc 1000 ms conform transmit violate drop
  class copp-system-p-class-redirect
    set cos 1
    police cir 200 kbps bc 250 ms conform transmit violate drop
  class copp-system-p-class-exception
    set cos 1
    police cir 200 kbps bc 250 ms conform transmit violate drop
  class copp-system-p-class-monitoring
    set cos 1
    police cir 130 kbps bc 1000 ms conform transmit violate drop
  class copp-system-p-class-l2-unpoliced
    police cir 8 gbps bc 5 mbytes conform transmit violate transmit
  class copp-system-p-class-undesirable
    set cos 0
    police cir 32 kbps bc 250 ms conform drop violate drop
  class copp-system-p-class-fcoe
```

```
  set cos 6
  police cir 600 kbps bc 1000 ms conform transmit violate drop
class copp-system-p-class-l2-default
  police cir 10 kbps bc 250 ms conform transmit violate drop
class class-default
  set cos 0
  police cir 10 kbps bc 250 ms conform transmit violate drop
```

Packets Per Second Credit Limit

The aggregate packets per second (PPS) for a given policy (sum of PPS of each class part of the policy) is capped by an upper PPS Credit Limit (PCL). If an increase in PPS of a given class causes a PCL exceed, the configuration is rejected. To increase the desired PPS, the additional PPS beyond PCL should be decreased from other class(es).

25.2.2 Modular QoS Command-Line Interface

CoPP uses the Modular Quality of Service Command-Line Interface (MQC). MQC is a CLI structure that allows you to define a traffic class, create a traffic policy (policy map), and attach the traffic policy to an interface. The traffic policy contains the CoPP feature that will be applied to the traffic class.

SUMMARY STEPS

1. Define a traffic class using the **class-map** command. A traffic class is used to classify traffic.
2. Create a traffic policy using the **policy-map** command. A traffic policy (policy map) contains a traffic class and one or more CoPP features that will be applied to the traffic class. The CoPP features in the traffic policy determine how to treat the classified traffic.
3. Attach the traffic policy (policy map) to the control plane using the **control-plane** and **service-policy** commands.

DETAILED STEPS

Step 1 Define a traffic class using the **class-map** command. A traffic class is used to classify traffic.

This example shows how to create a new class-map called copp-sample-class:

```
class-map type control-plane copp-sample-class
```

Step 2 Create a traffic policy using the **policy-map** command. A traffic policy (policy map) contains a traffic class and one or more CoPP features that will be applied to the traffic class. The CoPP features in the traffic policy determine how to treat the classified traffic.

Step 3 Attach the traffic policy (policy map) to the control plane using the **control-plane** and **service-policy** commands.

This example shows how to attach the policy map to the control plane:

```
control-plane
service-policy input copp-system-policy
```

Note The copp-system-policy is always configured and applied. There is no need to use this command explicitly.

25.2.3 CoPP and the Management Interface

The Inspur INOS device supports only hardware-based CoPP which does not support the management interface (mgmt0). The out-of-band mgmt0 interface connects directly to the CPU and does not pass through the in-band traffic hardware where CoPP is implemented.

On the mgmt0 interface, ACLs can be configured to give or deny access to a particular type of traffic.

Related Topics

Configuring IP ACLs
Configuring MAC ACLs.

25.2.4 Virtualization Support for CoPP

You can configure CoPP in the default virtual device context (VDC) or the admin VDC, but the CoPP configuration applies to all VDCs on the Inspur INOS device. For more information on VDCs, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

25.3 Licensing Requirements for CoPP

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

25.4 Guidelines and Limitations for CoPP

CoPP has the following configuration guidelines and limitations:CoPP classification does not work for the Layer 2 control traffic in native VLAN in the following scenarios:

- When the **native vlan** (ID other than 1) command is configured on the interface and the native VLAN ID is missing in the configuration.
- If the **vlan dot1q tag native exclude control command** is configured.
- We recommend that you use the strict default CoPP policy initially and then later modify the CoPP policies based on the data center and application requirements.
- We recommend applying the default dense policy when the chassis is fully loaded with F3 Series modules.
- Customizing CoPP is an ongoing process. CoPP must be configured according to the protocols and features used in your specific environment as well as the supervisor features that are required by the server environment. As these protocols and features change, CoPP must be modified.
- We recommend that you continuously monitor CoPP. If drops occur, determine if CoPP dropped traffic unintentionally or in response to a malfunction or attack. In either event, analyze the situation and evaluate the need to modify the CoPP policies.
- All the traffic that you do not specify in the other class maps is put into the last class, the default class. Monitor the drops in this class and investigate if these drops are based on traffic that you do not want or the result of a feature that was not configured and you need to add.
- All broadcast traffic is sent through CoPP logic in order to determine which packets (for example, ARP and DHCP) need to be redirected through an access control list (ACL) to the router processor. Broadcast traffic that does not need to be redirected is matched against the CoPP logic, and both conforming and violated packets are counted in the hardware but not sent to the CPU. Broadcast traffic that needs to be sent to the CPU and broadcast traffic that does not need to be sent to the CPU must be separated into different classes.
- After you have configured CoPP, delete anything that is not being used, such as old class maps and unused routing protocols.
- You must ensure that the CoPP policy does not filter critical traffic such as routing protocols or interactive

access to the device. Filtering this traffic could prevent remote access to the Inspur INOS device and require a console connection.

- The Inspur INOS software does not support egress CoPP or silent mode. CoPP is supported only on ingress (you cannot use the **service-policy output copp** command to the control plane interface).
- You can use the access control entry (ACE) hit counters in the hardware only for ACL logic. Use the software ACE hit counters and the **show access-lists** and **show policy-map type control-plane** commands to evaluate CPU traffic.
- The Inspur INOS device hardware performs CoPP on a per-forwarding-engine basis. CoPP does not support distributed policing. Therefore, you should choose rates so that the aggregate traffic does not overwhelm the supervisor module.
- To get a more granular view of traffic that reaches the supervisor and might be dropped by CoPP, you can use the NetFlow feature on SVIs. To do so, compare the ACL hit counts by the values listed in the NetFlow table.
- When you use ISSU to upgrade to a new Inspur INOS release, the default CoPP policy for the new release is not applied. Because you might have your own configured CoPP policy and want to continue using it, the policy for the prior release continues to be applied. However, if you have not modified the default CoPP policy in prior versions, we recommend that when you install Inspur INOS Release 8.2(3)or later releases, you apply the latest default CoPP policy for that version by using the **copp profile** [**strict** | **moderate** | **lenient**] command. This action removes the previous policy and applies the new one.
- Beginning with Inspur INOS Release8.2(3), the default CoPP policies are read only. To make modifications, copy the default profile by using the **copp copy profile** {**strict** | **moderate** | **lenient**} {**prefix** | **suffix**} *string*, make modifications, and then apply that policy to the control plane using the **service-policy input** *policy-map-name* command.
- If multiple flows map to the same class, individual flow statistics will not be available.
- Support for monitoring CoPP with SNMP is limited to the listed cbQoSMIB tables and the elements attached to the control plane.

25.5 Default Settings for CoPP

This table lists the default settings for CoPP parameters.

Table 52： Default CoPP Parameters Settings

Parameters	**Default**
Default policy	Strict
Default policy	9 policy entries **Note** The maximum number of supported policies with associated class maps is 128.
Scale factor value	1.00

25.6 Configuring CoPP

This section describes how to configure CoPP.

25.6.1 Configuring a Control Plane Class Map

You must configure control plane class maps for control plane policies.

You can classify traffic by matching packets based on existing ACLs. The permit and deny ACL keywords are ignored in the matching.

You can configure policies for IP version 4 (IPv4) and IP version 6 (IPv6) packets.

Before you begin

Ensure that you are in the default VDC.

Ensure that you have configured the IP ACLs if you want to use ACE hit counters in the class maps.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **class-map type control-plane** [**match-all** | **match-any**] *class-map-name*
3. (Optional) switch(config-cmap)# **match access-group name** *access-list-name*
4. (Optional) switch(config-cmap)# **match exception** {**ip** | **ipv6**} **icmp redirect**
5. (Optional) switch(config-cmap)# **match exception** {**ip** | **ipv6**} **icmp unreachable**
6. (Optional) switch(config-cmap)# **match exception** {**ip** | **ipv6**} **option**
7. (Optional) switch(config-cmap)# **match exception** {**ip** | **ipv6**} **unicast rpf-failure**
8. switch(config-cmap)# **match protocol arp**
9. (Optional) switch(config-cmap)# **match redirect arp-inspect**
10. (Optional) switch(config-cmap)# **match redirect dhcp-snoop**
11. switch(config-cmap)# **exit**
12. (Optional) switch(config)# **show class-map type control-plane** [*class-map-name*]
13. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **class-map type control-plane** [**match-all** | **match-any**] *class-map-name*	Specifies a control plane class map and enters class map configuration mode. The default class matching is match-any. The name can be a maximum of 64 characters long and is case sensitive. **Note** You cannot use class-default, match-all, or match-any as class map names.
Step 3	(Optional) switch(config-cmap)# **match access-group name** *access-list-name*	Specifies matching for an IP ACL. **Note** The permit and deny ACL keywords are ignored in the CoPP matching.
Step 4	(Optional) switch(config-cmap)# **match exception** {**ip** | **ipv6**} **icmp redirect**	Specifies matching for IPv4 or IPv6 ICMP redirect exception packets.
Step 5	(Optional) switch(config-cmap)# **match exception** {**ip** |	Specifies matching for IPv4 or IPv6 ICMP unreachable exception packets.

	Command or Action	Purpose
	ipv6} **icmp unreachable**	
Step 6	(Optional) switch(config-cmap)# **match exception** {**ip** | **ipv6**} **option**	Specifies matching for IPv4 or IPv6 option exception packets.
Step 7	(Optional) switch(config-cmap)# **match exception** {**ip** | **ipv6**} **unicast rpf-failure**	Specifies matching for IPv4 or IPv6 Unicast Reverse Path Forwarding (Unicast RPF) exception packets. For any CoPP class map, you can rate limit the IPv4 or IPv6 URPF exception packets as per the class map's rate limit configuration.
Step 8	switch(config-cmap)# **match protocol arp**	Specifies matching for IP Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (RARP) packets.
Step 9	(Optional) switch(config-cmap)# **match redirect arp-inspect**	Specifies matching for ARP inspection redirected packets.
Step 10	(Optional) switch(config-cmap)# **match redirect dhcp-snoop**	Specifies matching for Dynamic Host Configuration Protocol (DHCP) snooping redirected packets.
Step 11	switch(config-cmap)# **exit**	Exits class map configuration mode.
Step 12	(Optional) switch(config)# **show class-map type control-plane** [*class-map-name*]	Displays the control plane class map configuration.
Step 13	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

25.6.2 Configuring a Control Plane PolicyMap

You must configure a policy map for CoPP, which includes policing parameters. If you do not configure a policer for a class, the default policer conform action is drop. The Inspur INOS software supports 1-rate 2-color and 2-rate 3-color policing.

Before you begin

Ensure that you are in the default VDC.
Ensure that you have configured a control plane class map.

SUMMARY STEPS

1. **configure terminal**
2. **policy-map type control-plane** *policy-map-name*
3. **class** {*class-map-name* [**insert-before** *class-map-name2*] | **class-default**}
4. **police** [**cir**] {*cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**]}
5. **police** [**cir**] {cir-rate [**bps** | **gbps** | **kbps** | **mbps** | **pps**]} [**bc**] *burst-size* [**bytes** | **kbytes** | **mbytes** | **ms** | **packets** | **us**]
6. **police** [**cir**] {*cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**]} **conform** {**drop** | **set-cos-transmit** *cos-value* | **set-dscp-transmit** *dscp-value* | **set-prec-transmit** *prec-value* | **transmit**} [**exceed** {**drop** | **set dscp dscp table cir-markdown-map** | **transmit**}] [**violate** {**drop** | **set dscp dscp table pir-markdown-map** | **transmit**}]
7. **police** [**cir**] {*cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**]} **pir** *pir-rate* [**bps** | **gbps** | **kbps** | **mbps**] [[**be**] *burst-size* [**bytes** | **kbytes** | **mbytes** | **ms** | **packets** | **us**]]

8. (Optional) **set cos** [**inner**] *cos-value*
9. (Optional) **set dscp** [**tunnel**] {*dscp-value* | **af11** | **af12** | **af13** | **af21** | **af22** | **af23** | **af31** | **af32** | **af33** | **af41** | **af42** | **af43** | **cs1** | **cs2** | **cs3** | **cs4** | **cs5** | **cs6** | **cs7** | **ef** | **default**}
10. (Optional) **set precedence** [**tunnel**] {*prec-value* | **critical** | **flash** | **flash-override** | **immediate** | **internet** | **network** | **priority** | **routine**}
11. **exit**
12. **exit**
13. (Optional) **show policy-map type control-plane** [**expand**] [**name** *class-map-name*]
14. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**policy-map type control-plane** *policy-map-name*
	Example:
	`switch(config)# policy-map type control-plane ClassMapA`
	`switch(config-pmap)#`
		Specifies a control plane policy map and enters policy map configuration mode. The policy map name can have a maximum of 64 characters and is case sensitive.

Step 3	**class** {*class-map-name* [**insert-before***class-map-name2*]
	| **class-default**}
	Example:
	`switch(config-pmap)# class ClassMapA`
	`switch(config-pmap-c)#`
		Specifies a control plane class map name or the class default and enters control plane class configuration mode.
		The class-default class map is always at the end of the class map list for a policy map.

Step 4	**police** [**cir**] {*cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**]} **Example:** `switch(config-pmap-c)# police cir 52000`	Specifies the committed information rate (CIR). The rate range is from 0 to 80000000000. The default CIR unit is bps.
Step 5	**police** [**cir**] {cir-rate [**bps** | **gbps** | **kbps** | **mbps** | **pps**]} [**bc**] *burst-size* [**bytes** | **kbytes** | **mbytes** | **ms** | **packets** | **us**] **Example:** `switch(config-pmap-c)# police cir 52000 bc 1000`	Specifies the CIR with the committed burst (BC). The CIR range is from 0 to 80000000000 and the BC range is from 0 to 512000000. The default CIR unit is bps and the default BC size unit is **bytes**.
Step 6	**police** [**cir**] {*cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**]} **conform** {**drop** | **set-cos-transmit** *cos-value* | **set-dscp-transmit** *dscp-value* | **set-prec-transmit** *prec-value* | **transmit**} [**exceed** {**drop** | **set dscp dscp table cir-markdown-map** | **transmit**}] [**violate** {**drop** | **set dscp dscp table pir-markdown-map** | **transmit**}]	Specifies the CIR with the conform action. The CIR range is from 0 to 80000000000. The default rate unit is bps. The range for the *cos-value* and *prec-value* arguments is from 0 to 7. The range for the *dscp-value* argument is from 0 to 63. The options are as follows: • **drop**—Drops the packet.

	Command or Action	Purpose
	Example: `switch(config-pmap-c)# police cir 52000 conform transmit exceed drop`	• **set-cos-transmit**—Sets the class of service (CoS) value. • **set-dscp-transmit**—Sets the differentiated services code point value. • **set-prec-transmit**—Sets the precedence value. • **transmit**—Transmits the packet. • **set dscp dscp table cir-markdown-map**—Sets the exceed action to the CIR markdown map. • **set dscp dscp table pir-markdown-map**—Sets the violate action to the PIR markdown map. **Note** You can specify the BC and conform action for the same CIR.
Step 7	**police** [**cir**] {*cir-rate* [**bps** | **gbps** | **kbps** | **mbps** | **pps**]} **pir** *pir-rate* [**bps** | **gbps** | **kbps** | **mbps**] [[**be**] *burst-size* [**bytes** | **kbytes** | **mbytes** | **ms** | **packets** | **us**]] **Example:** `switch(config-pmap-c)# police cir 52000 pir 78000 be 2000`	Specifies the CIR with the peak information rate (PIR). The CIR range is from 0 to 80000000000 and the PIR range is from 1 to 80000000000. You can optionally set an extended burst (BE) size. The BE range is from 1 to 512000000. The default CIR unit is bps, the default PIR unit is **bps**, and the default BE size unit is **bytes**. **Note** You can specify the BC, conform action, and PIR for the same CIR.
Step 8	(Optional) **set cos** [**inner**] *cos-value* **Example:** `switch(config-pmap-c)# set cos 1`	Specifies the 802.1Q class of service (CoS) value. Use the **inner** keyword in a Q-in-Q environment. The range is from 0 to 7. The default value is 0.
Step 9	(Optional) **set dscp** [**tunnel**] {*dscp-value* | **af11** | **af12** | **af13** | **af21** | **af22** | **af23** | **af31** | **af32** | **af33** | **af41** | **af42** | **af43** | **cs1** | **cs2** | **cs3** | **cs4** | **cs5** | **cs6** | **cs7** | **ef** | **default**} **Example:** `switch(config-pmap-c)# set dscp 10`	Specifies the differentiated services code point value in IPv4 and IPv6 packets. Use the **tunnel** keyword to set tunnel encapsulation. The range is from 0 to 63. The default value is 0.
Step 10	(Optional) **set precedence** [**tunnel**] {*prec-value* | **critical** | **flash** | **flash-override** | **immediate** | **internet** | **network** | **priority** | **routine**} **Example:** `switch(config-pmap-c)# set precedence 2`	Specifies the precedence value in IPv4 and IPv6 packets. Use the **tunnel** keyword to set tunnel encapsulation. The range is from 0 to 7. The default value is 0.
Step 11	**exit** **Example:** `switch(config-pmap-c)# exit`	Exits policy map class configuration mode.

	Command or Action	Purpose
	`switch(config-pmap)#`	
Step 12	**exit** **Example:** `switch(config-pmap)# exit switch(config)#`	Exits policy map configuration mode.
Step 13	(Optional) **show policy-map type control-plane** [**expand**] [**name** *class-map-name*] **Example:** `switch(config)# show policy-map type control-plane`	Displays the control plane policy map configuration.
Step 14	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring a Control Plane Class Map.

25.6.3 Configuring the Control Plane Service Policy

You can configure one or more policy maps for the CoPP service policy.

Before you begin

Ensure that you are in the default VDC.
Ensure that you have configured a control plane policy map.

SUMMARY STEPS

1. **configure terminal**
2. **control-plane**
3. **service-policy input** *policy-map-name*
4. **exit**
5. (Optional) **show running-config copp** [**all**]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**control-plane**
	Example:
	`switch(config)# control-plane`
	`switch(config-cp)#`
		Enters control plane configuration mode.

Step 3	**service-policy input** *policy-map-name*	Specifies a policy map for the input traffic. Repeat

	Command or Action	Purpose
	Example: `switch(config-cp)# service-policy input PolicyMapA`	this step if you have more than one policy map. Use the **no service-policy input** *policy-map-name* command to remove the policy from the control plane.

Step 4	**exit**
	Example:
	`switch(config-cp)# exit`
	`switch(config)#`
		Exits control plane configuration mode.

Step 5	(Optional) **show running-config copp** [**all**] **Example:** `switch(config)# show running-config copp`	Displays the CoPP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Related Topics

Configuring a Control Plane Policy Map.

25.6.4 Configuring the CoPP Scale Factor Per Line Card

You can configure the CoPP scale factor per line card.

The scale factor configuration is used to scale the policer rate of the applied CoPP policy for a particular line card. The accepted value is from 0.10 to 2.00. You can increase or reduce the policer rate for a particular line card without changing the current CoPP policy. The changes are effective immediately, so you do not need to reapply the CoPP policy.

Before you begin

Ensure that you are in the default VDC.

SUMMARY STEPS

1. **configure terminal**
2. **control-plane**
3. **scale-factor** *value* **module** *multiple-module-range*
4. (Optional) **show running-config copp** [**all**]
5. (Optional) **show policy-map interface control-plane** [**class** *class-map* | **module** *slot*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**control-plane**
	Example:
	`switch(config)# control-plane`
	`switch(config-cp)#`
		Enters control plane configuration mode.

Step 3	**scale-factor** *value* **module** *multiple-module-range* **Example:** `switch(config-cp)# scale-factor 1.10 module 1-2`	Configures the policer rate per line card. The allowed scale factor value is from 0.10 to 2.00. When the scale factor value is configured, the policing values are multiplied by the corresponding scale factor value of the module, and it is programmed in the particular module. To revert to the default scale factor value of 1.00, use the **no scale-factor** *value* **module** *multiple-module-range* command, or explicitly set the default scale factor value to 1.00 using the **scale-factor 1 module** *multiple-module-range* command.
Step 4	(Optional) **show running-config copp** [**all**] **Example:** `switch(config-cp)# show running-config copp`	Displays the CoPP configuration in the running configuration.
Step 5	(Optional) **show policy-map interface control-plane** [**class** *class-map* | **module** *slot*] **Example:** `switch(config-cp)# show policy-map interface control-plane`	Displays the applied scale factor values when a CoPP policy is applied.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

25.6.5 Changing or Reapplying the Default CoPP Policy

You can change to a different default CoPP policy, or you can reapply the same default CoPP policy.

SUMMARY STEPS

1. [**no**] **copp profile** [**strict** | **moderate** | **lenient** | **dense**]
2. (Optional) **show copp status**
3. (Optional) **show running-config copp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	[**no**] **copp profile** [**strict** | **moderate** | **lenient** | **dense**] **Example:** `switch(config)# copp profile moderate`	Applies the CoPP best practice policy.

	Command or Action	Purpose
Step 2	(Optional) **show copp status** **Example:** `switch(config)# show copp status`	Displays the CoPP status, including the last configuration operation and its status. This command also enables you to verify that the CoPP best practice policy is attached to the control plane.
Step 3	(Optional) **show running-config copp** **Example:** `switch(config)# show running-config copp`	Displays the CoPP configuration in the running configuration.

Related Topics

Changing or Reapplying the Default CoPP Policy Using the Setup Utility.

25.7 Verifying the CoPP Configuration

To display CoPP configuration information, perform one of the following tasks:

Command	Purpose
show policy-map type control-plane [**expand**] [**name** *policy-map-name*]	Displays the control plane policy map with associated class maps and CIR and BC values.
show policy-map interface control-plane [**class** *class-map* | **module** *slot*]	Displays the policy values with associated class maps and drops per policy or class map. It also displays the scale factor values when a CoPP policy is applied. When the scale factor value is the default (1.00), it is not displayed. **Note** The scale factor changes the CIR, BC, PIR, and BE values internally on each module, but the display shows the configured CIR, BC, PIR, and BE values only. The actual applied value on a module is the scale factor multiplied by the configured value.
show class-map type control-plane [*class-map-name*]	Displays the control plane class map configuration, including the ACLs that are bound to this class map.
show ip access-lists [*acl-name*]	Displays the access lists, including the ACLs. If the **statistics per-entry** command is used, it also displays hit counts for specific entries.
show running-config copp [**all**]	Displays the CoPP configuration in the

	running configuration.

For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

25.8 Displaying the CoPP Configuration Status

Before you begin

Ensure that you are in the default VDC.

SUMMARY STEPS

1. switch# **show copp status**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show copp status**	Displays the configuration status for the CoPP feature.

Example

This example shows how to display the CoPP configuration status:

```
switch# show copp status
```

25.9 Monitoring CoPP

Before you begin

Ensure that you are in the default VDC.

SUMMARY STEPS

1. switch# **show policy-map interface control-plane** {[**module** *module-number* [**inst-all**]] [**class** {*class-map* | **violated**}] | [**class** {*class-map* | **violated**}] [**module** *module-number* [**inst-all**]]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show policy-map interface control-plane** {[**module** *module-number* [**inst-all**]] [**class** {*class-map* | **violated**}] | [**class** {*class-map* | **violated**}] [**module** *module-number* [**inst-all**]]}	Displays packet-level statistics for all classes that are part of the applied CoPP policy. Statistics are specified in terms of OutPackets (packets admitted to the control plane) and DropPackets (packets dropped because of rate limiting). **Note** With Supervisor 3 or F3Series modules, the output of this command uses Layer 3 packet lengths when displaying the byte count. With F3 modules, the command output uses Layer 2 packet lengths for the byte count.

	Command or Action	Purpose
		Note From Inspur INOS Release 8.2(3), you can display the per-instance statistics for all classes that are part of the applied control plane policing (CoPP) policy for a module by using the **inst-all** keyword.

Example

This example shows how to monitor CoPP:

```
switch# show policy-map interface control-plane
Control Plane

service-policy input: copp-system-policy-default

class-map copp-system-class-igmp (match-any)
match protocol igmp
police cir 1024 kbps , bc 65535 bytes
conformed 0 bytes; action: transmit
violated 0 bytes;
class-map copp-system-class-pim-hello (match-any)
match protocol pim
police cir 1024 kbps , bc 4800000 bytes
conformed 0 bytes; action: transmit
violated 0 bytes;
....
```

This example shows the 5-minute moving averages and peaks of the conformed and violated byte counts in the output of the **show policy-map interface control-plane** command. In this example, the 5-minute offered rate is the 5-minute moving average of the conformed bytes, the 5-minute violate rate is the 5-minute moving average of the violated bytes, and the peak rate is the highest value since boot-up or counter reset, with the peak occurring at the timestamp shown.

```
module 9:
    conformed 0 bytes,
        5-min offered rate 10 bytes/sec
        peak rate 12 bytes/sec at 12:29:38.654 UTC Sun Jun 30 2013

    violated 0 bytes,
        5-min violate rate 20 bytes/sec
        peak rate 22 bytes/sec at 12:26:22.652 UTC Sun Jun 30 2013
```

This example displays the per-instance statistics for all classes that are part of the applied control plane policing (CoPP) policy for a module.

```
switch(config)# show policy-map interface control-plane module 9 inst-all
Control Plane
  service-policy input copp-system-p-policy-strict

    class-map copp-system-p-class-critical (match-any)
      match access-group name copp-system-p-acl-bgp
      match access-group name copp-system-p-acl-rip
      match access-group name copp-system-p-acl-vpc
      match access-group name copp-system-p-acl-bgp6
      match access-group name copp-system-p-acl-lisp
      match access-group name copp-system-p-acl-ospf
      match access-group name copp-system-p-acl-rip6
      match access-group name copp-system-p-acl-rise
      match access-group name copp-system-p-acl-eigrp
```

```
match access-group name copp-system-p-acl-lisp6
match access-group name copp-system-p-acl-ospf6
match access-group name copp-system-p-acl-rise6
match access-group name copp-system-p-acl-eigrp6
match access-group name copp-system-p-acl-otv-as
match access-group name copp-system-p-acl-mac-l2pt
match access-group name copp-system-p-acl-mpls-ldp
match access-group name copp-system-p-acl-mpls-rsvp
match access-group name copp-system-p-acl-mac-l3-isis
match access-group name copp-system-p-acl-mac-otv-isis
match access-group name copp-system-p-acl-mac-fabricpath-isis
match protocol mpls router-alert
set cos 7
police cir 36000 kbps bc 250 ms
  conform action: transmit
  violate action: drop
module 9:
inst 0:
  conformed 3215360 bytes,
    5-min offered rate 7 bytes/sec
    peak rate 9 bytes/sec at Fri Apr 28 11:58:48 2017
inst 1:
  conformed 3210508 bytes,
    5-min offered rate 7 bytes/sec
    peak rate 8 bytes/sec at Wed May 03 05:19:24 2017
inst 2:
  conformed 0 bytes,
    5-min offered rate 0 bytes/sec
    peak rate 0 bytes/sec
inst 3:
  conformed 0 bytes,
    5-min offered rate 0 bytes/sec
    peak rate 0 bytes/sec
inst 4:
  conformed 0 bytes,
    5-min offered rate 0 bytes/sec
    peak rate 0 bytes/sec
inst 5:
  conformed 0 bytes,
    5-min offered rate 0 bytes/sec
    peak rate 0 bytes/sec
inst 0:
  violated 0 bytes,

    5-min violate rate 0 bytes/sec
    peak rate 0 bytes/sec
inst 1:
  violated 0 bytes,
    5-min violate rate 0 bytes/sec
    peak rate 0 bytes/sec
inst 2:
  violated 0 bytes,
    5-min violate rate 0 bytes/sec
    peak rate 0 bytes/sec
inst 3:
  violated 0 bytes,
    5-min violate rate 0 bytes/sec
    peak rate 0 bytes/sec
inst 4:
  violated 0 bytes,
    5-min violate rate 0 bytes/sec
    peak rate 0 bytes/sec
inst 5:
  violated 0 bytes,
```

```
      5-min violate rate 0 bytes/sec
      peak rate 0 bytes/sec

class-map copp-system-p-class-important (match-any)
  match access-group name copp-system-p-acl-cts
  match access-group name copp-system-p-acl-glbp
  match access-group name copp-system-p-acl-hsrp
  match access-group name copp-system-p-acl-vrrp
  match access-group name copp-system-p-acl-wccp
  match access-group name copp-system-p-acl-hsrp6
  match access-group name copp-system-p-acl-vrrp6
  match access-group name copp-system-p-acl-opflex
  match access-group name copp-system-p-acl-mac-lldp
  match access-group name copp-system-p-acl-mac-mvrp
  match access-group name copp-system-p-acl-mac-flow-control
  set cos 6
  police cir 1400 kbps bc 1500 ms
    conform action: transmit
    violate action: drop
  module 9:
  inst 0:
    conformed 0 bytes,
      5-min offered rate 0 bytes/sec
      peak rate 0 bytes/sec
  inst 1:
    conformed 0 bytes,
      5-min offered rate 0 bytes/sec
      peak rate 0 bytes/sec
  inst 2:
    conformed 0 bytes,
      5-min offered rate 0 bytes/sec
      peak rate 0 bytes/sec
```

25.10 Monitoring CoPP with SNMP

Beginning with Inspur INOS Release 8.2(3), CoPP supports the Inspur class-based QoS MIB (cbQoSMIB). All of the CoPP elements can now be monitored (but not modified) using SNMP. This feature applies only to policies and their subelements (such as classes, match rules, and set actions) that are attached to the control plane. Elements of policies that are not in service on the control plane are not visible through SNMP.

The following cbQoSMIB tables are supported:

- ccbQosServicePolicy
- cbQosInterfacePolicy
- cbQosObjects
- cbQosPolicyMapCfg
- cbQosClassMapCfg
- cbQosMatchStmtCfg
- cbQosPoliceCfg
- cbQosSetCfg

25.11 Clearing the CoPP Statistics

Before you begin

Ensure that you are in the default VDC.

SUMMARY STEPS

1. (Optional) switch# **show policy-map interface control-plane** [**class** *class-map* | **module** *slot*]
2. switch# **clear copp statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) switch# **show policy-map interface control-plane** [**class** *class-map* | **module** *slot*]	Displays the currently applied CoPP policy and per-class statistics.
Step 2	switch# **clear copp statistics**	Clears the CoPP statistics.

Example

This example shows how to clear the CoPP statistics for your installation:

```
switch# show policy-map interface control-plane
switch# clear copp statistics
```

25.12 Configuration Examples for CoPP

This section includes example CoPP configurations.

25.12.1 CoPP Configuration Example

The following example shows how to configure CoPP using IP ACLs and MAC ACLs:

```
configure terminal
ip access-list copp-system-acl-igmp
permit igmp any 10.0.0.0/24

ip access-list copp-system-acl-msdp
permit tcp any any eq 639

mac access-list copp-system-acl-arp
permit any any 0x0806

ip access-list copp-system-acl-tacas
permit udp any any eq 49

ip access-list copp-system-acl-gre
permit 47 any any

ip access-list copp-system-acl-ntp
permit udp any 10.0.1.1/23 eq 123

ip access-list copp-system-acl-icmp
permit icmp any any

class-map type control-plane match-any copp-system-class-critical
match access-group name copp-system-acl-igmp
match access-group name copp-system-acl-msdp

class-map type control-plane match-any copp-system-class-important
match access-group name copp-system-acl-gre

class-map type control-plane match-any copp-system-class-normal
match access-group name copp-system-acl-icmp
match exception ip icmp redirect
match exception ip icmp unreachable
```

```
match exception ip option
match redirect arp-inspect
match redirect dhcp-snoop

policy-map type control-plane copp-system-policy

class copp-system-class-critical
police cir 2000 kbps bc 1500 bytes pir 3000 kbps be 1500 bytes conform
    transmit exceed transmit violate drop

class copp-system-class-important
police cir 1000 kbps bc 1500 bytes pir 1500 kbps be 1500 bytes conform
    transmit exceed transmit violate drop

class copp-system-class-normal
police cir 400 kbps bc 1500 bytes pir 600 kbps be 1500 bytes conform
    transmit exceed transmit violate drop


class class-default
police cir 200 kbps bc 1500 bytes pir 300 kbps be 1500 bytes conform
    transmit exceed transmit violate drop

control-plane
service-policy input copp-system-policy
```

The following example shows how to create the CoPP class and associate an ACL:

```
class-map type control-plane copp-arp-class
match access-group name copp-arp-acl
```

The following example shows how to add the class to the CoPP policy:

```
policy-map type control-plane copp-system-policy
class copp-arp-class
police pps 500
```

25.12.2 Preventing CoPP Overflow by Splitting ICMP Pings and ARP Requests

Some servers use ICMP pings and ARP requests to the default gateway to verify that the active NIC still has access to the aggregation switch. As a result, if the CoPP values are exceeded, CoPP starts dropping traffic for all networks. One malfunctioning server can send out thousands of ICMP pings and ARP requests, causing all servers in one aggregation block to lose their active NIC and start swapping NICs.

If your server is configured as such, you can minimize the CoPP overflow by splitting the ICMP pings and ARP requests based on subnets or groups of subnets. Then if a server malfunctions and overflows CoPP, the supervisor answers the ICMP pings and ARP requests only on some subnetworks.

The last entry in the class map or policy map should identify all of the ICMP pings and ARP requests in the networks that are not specified. If these counters increase, it means that a new network was added that was not specified in the existing ACLs for ICMP and ARP. In this case, you would need to update the ACLs related to ICMP and ARP.

The following example shows how to prevent a CoPP overflow by splitting ICMP and ARP requests.

First, add the new ACLs that identify the networks you want to group together based on the findings of the investigations of the applications:

```
arp access-list copp-arp-1
statistics per-entry
10 permit ip 10.1.1.0 255.255.255.0 mac any
20 permit ip 10.1.2.0 255.255.255.0 mac any
30 permit ip 10.1.3.0 255.255.255.0 mac any
arp access-list copp-arp-2
statistics per-entry
```

```
10 permit ip 10.2.1.0 255.255.255.0 mac any
20 permit ip 10.2.2.0 255.255.255.0 mac any
30 permit ip 10.2.3.0 255.255.255.0 mac any
arp access-list copp-arp-3
statistics per-entry
10 permit ip 10.3.1.0 255.255.255.0 mac any
20 permit ip 10.3.2.0 255.255.255.0 mac any
30 permit ip 10.3.3.0 255.255.255.0 mac any


...
arp access-list copp-arp-10
10 permit ip any any mac any

ip access-list copp-icmp-1
statistics per-entry
10 permit icmp 10.2.1.0 255.255.255.0 any
20 permit icmp 10.2.2.0 255.255.255.0 any
30 permit icmp 10.2.3.0 255.255.255.0 any
ip access-list copp-icmp-2
statistics per-entry
10 permit icmp 10.3.1.0 255.255.255.0 any
10 permit icmp 10.3.2.0 255.255.255.0 any
10 permit icmp 10.3.3.0 255.255.255.0 any
ip access-list copp-icmp-3
statistics per-entry
10 permit icmp 10.4.1.0 255.255.255.0 any
10 permit icmp 10.4.2.0 255.255.255.0 any
10 permit icmp 10.4.3.0 255.255.255.0 any
...
ip access-list copp-icmp-10
10 permit icmp any any
```

Add these ACLs to the new class maps for CoPP:

```
class-map type control-plane match-any copp-cm-arp-1
 match access-group name copp-arp-1
class-map type control-plane match-any copp-cm-arp-2
 match access-group name copp-arp-2
class-map type control-plane match-any copp-cm-arp-3
 match access-group name copp-arp-3
...
class-map type control-plane match-any copp-cm-arp-10
 match access-group name copp-arp-10# class-map type control-plane match-any copp-cm-icmp-1

 match access-group name copp-icmp-1
class-map type control-plane match-any copp-cm-icmp-2
 match access-group name copp-icmp-2
class-map type control-plane match-any copp-cm-icmp-3
 match access-group name copp-icmp-3
...
class-map type control-plane match-any copp-cm-icmp-10
 match access-group name copp-icmp-10
```

Modify the CoPP policy map by adding new policies with the above created class maps:

```
policy-map type control-plane copp-system-policy
class copp-cm-icmp-1
      police cir X kbps bc X ms conform transmit violate drop
class copp-cm-icmp-2
      police cir X kbps bc X ms conform transmit violate drop
class copp-cm-icmp-3
      police cir X kbps bc X ms conform transmit violate drop
class copp-cm-icmp-4
```

```
      police cir X kbps bc X ms conform transmit violate drop
class copp-cm-icmp-10
      police cir X kbps bc X ms conform transmit violate drop
class copp-cm-arp-1
      police cir X kbps bc X ms conform transmit violate drop
class copp-cm-arp-2
      police cir X kbps bc X ms conform transmit violate drop
class copp-cm-arp-3
      police cir X kbps bc X ms conform transmit violate drop


class copp-cm-arp-4
      police cir X kbps bc X ms conform transmit violate drop
class copp-cm-arp-10
      police cir X kbps bc X ms conform transmit violate drop
```

Delete ICMP and ARP from the existing class maps:

```
class-map type control-plane match-any copp-system-class-normal
no match protocol arp

class-map type control-plane match-any copp-system-class-monitoring
no match access-grp name copp-system-acl-icmp
```

25.13 Changingor Reapplyingthe Default CoPPPolicy Usingthe Setup Utility

The following example shows how to change or reapply the default CoPP policy using the setup utility.

```
switch# setup

         ---- Basic System Configuration Dialog VDC: 1 ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.


*Note: setup is mainly used for configuring the system initially,
when no configuration is present. So setup always assumes system
defaults and not the current system configuration values.


Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.

Would you like to enter the basic configuration dialog (yes/no): yes

Do you want to enforce secure password standard (yes/no)[y]: <CR>

  Create another login account (yes/no) [n]: n

  Configure read-only SNMP community string (yes/no) [n]: n

  Configure read-write SNMP community string (yes/no) [n]: n

  Enter the switch name : <CR>

  Enable license grace period? (yes/no) [n]: n

  Continue with Out-of-band (mgmt0) management configuration? (yes/no) [y]: n
```

```
  Configure the default gateway? (yes/no) [y]: n

  Configure advanced IP options? (yes/no) [n]: <CR>

  Enable the telnet service? (yes/no) [n]: y

  Enable the ssh service? (yes/no) [y]: <CR>


    Type of ssh key you would like to generate (dsa/rsa) : <CR>

  Configure the ntp server? (yes/no) [n]: n

  Configure default interface layer (L3/L2) [L3]: <CR>

  Configure default switchport interface state (shut/noshut) [shut]: <CR>

  Configure best practices CoPP profile (strict/moderate/lenient/dense/) [strict]: strict

  Configure CMP processor on current sup (slot 6)? (yes/no) [y]: n

  Configure CMP processor on redundant sup (slot 5)? (yes/no) [y]: n

The following configuration will be applied:
  password strength-check
  no license grace-period
  no telnet server enable
  no system default switchport
  system default switchport shutdown
  policy-map type control-plane copp-system-policy

Would you like to edit the configuration? (yes/no) [n]: <CR>

Use this configuration and save it? (yes/no) [y]: y

switch#
```

25.14 Additional References forCoPP

This section provides additional information related to implementing CoPP.

Related Documents

Related Topic	Document Title
Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*

Standards

Standards	Title
RFC 2698	A Two Rate Three Color Marker

25.15 Feature History for CoPP

This table lists the release history for this feature.

Table 53： Feature History for CoPP

Feature Name	Releases	Feature Information
CoPP	8.2(3)	Updated the output of the **show policy-map interface control-plane** command to show the 5-minute moving averages and peaks of the conformed and violated byte counts for each policy in each module.
CoPP	8.2(3)	Added VRRP6 ACL support to police VRRP IPv6 traffic. The HSRP ACL is modified to reflect the correct destination addresses of control packets.
CoPP	8.2(3)	Changed the behavior of multicast traffic from being policed at different rates in different classes to being grouped into three classes (multicast-host, multicast-router, and normal) and policed at consistent rates.
CoPP	8.2(3)	Added the ability to monitor CoPP with SNMP.
CoPP	8.2(3)	Added a new class for FCoE; added the LISP, LISP6, and MAC Layer 3 IS-IS ACLs to the critical class; added the fcoe-fib-miss match exception to the undesirable class; added the MAC Layer 2 tunnel ACL to the Layer 2 unpoliced class, and added the "permit icmp any any 143" rule to the acl-icmp6-msgs ACL.
CoPP	8.2(3)	Added the dense default CoPP policy.
CoPP	8.2(3)	Added the ability to configure the CoPP scale factor per line card.
CoPP	8.2(3)	Updated the default policies with support for ACL DHCP.
CoPP	8.2(3)	Updated the default policies with support for WCCP and Inspur TrustSec.

CHAPTER 26 Configuring Rate Limits

This chapter describes how to configure rate limits for supervisor-bound traffic on Inspur INOS devices. This chapter includes the following sections:

- Finding Feature Information
- Information About Rate Limits
- Virtualization Support for Rate Limits
- Licensing Requirements for Rate Limits
- Guidelines and Limitations for Rate Limits
- Default Settings for Rate Limits
- Configuring Rate Limits
- Monitoring Rate Limits
- Clearing the Rate Limit Statistics
- Verifying the Rate Limit Configuration
- Configuration Examples for Rate Limits
- Additional References for Rate Limits
- Feature History for Rate Limits

26.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

26.2 Information About Rate Limits

Rate limits can prevent redirected packets for exceptions from overwhelming the supervisor module on a Inspur INOS device. You can configure rate limits in packets per second for the following types of redirected packets:

- Access-list log packets
- Data and control packets copied to the supervisor module
- Layer 2 multicast-snooping packets
- Layer 2 port-security packets
- Layer 2 storm-control packets
- Layer 2 virtual port channel (vPC) low packets
- Layer 3 control packets
- Layer 3 glean packets
- Layer 3 glean fast-path packets
- Layer 3 maximum transmission unit (MTU) check failure packets
- Layer 3 multicast data packets
- Layer 3 Time-to-Live (TTL) check failure packets
- Receive packets

26.3 Virtualization Support for Rate Limits

You can configure rate limits only in the default virtual device context (VDC), but the rate limits configuration applies to all VDCs on the Inspur INOS device. For more information on VDCs, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

26.4 Licensing Requirements for Rate Limits

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS	Rate limits require no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For an explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

26.5 Guidelines and Limitations for Rate Limits

The rate limits feature has the following configuration guidelines and limitations:

- You can set rate limits for supervisor-bound exception and redirected traffic. Use control plane policing (CoPP) for other types of supervisor-bound traffic.

Note

Related Topics

Configuring Control Plane Policing.

26.6 Default Settings for Rate Limits

This table lists the default settings for rate limits parameters.

Table 54： Default Rate Limits Parameters Settings

Parameters	Default
Access-list log packets rate limit	100 packets per second
Copy packets rate limit	30,000 packets per second
Layer 2 multicast-snooping packets rate limit	10,000 packets per second
Layer 2 port-security packets rate limit	Disabled
Layer 2 storm-control packets rate limit	Disabled
Layer 2 VPC low packets rate limit	4,000 packets per second
Layer 3 control packets rate limit	10,000 packets per second
Layer 3 glean packets rate limit	100 packets per second
Layer 3 glean fast-path rate limit	100 packets per second
Layer 3 MTU packets rate limit	500 packets per second
Layer 3 Time-to-Live (TTL) packets rate limit	500 packets per second
Receive packets rate limit	30,000 packets per second

26.7 Configuring Rate Limits

You can set rate limits on supervisor-bound traffic.

SUMMARY STEPS

1. **configure terminal**
2. **hardware rate-limiter access-list-log** *packets*
3. **hardware rate-limiter copy** *packets*
4. **hardware rate-limiter layer-2 mcast-snooping** *packets*
5. **hardware rate-limiter layer-2 port-security** *packets*
6. **hardware rate-limiter layer-2 storm-control** *packets*
7. **hardware rate-limiter layer-2 vpc-low** *packets*
8. **hardware rate-limiter layer-3 control** *packets*
9. **hardware rate-limiter layer-3 glean** *packets*
10. **hardware rate-limiter layer-3 glean-fast** *packets*
11. **hardware rate-limiter layer-3 mtu** *packets*
12. **hardware rate-limiter layer-3 multicast** *packets*
13. **hardware rate-limiter layer-3 ttl** *packets*
14. **hardware rate-limiter receive** *packets*
15. **exit**
16. (Optional) **show hardware rate-limiter** [**access-list-log** | **copy** | **layer-2** {**mcast-snooping** | **port-security** | **storm-control** | **vpc-low**} | **layer-3** {**control** | **glean** | **glean-fast** | **mtu** | **multicast** | **ttl**} | **module** *module* | **receive**]
17. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**hardware rate-limiter access-list-log** *packets* **Example:** `switch(config)# hardware rate-limiter access-list-log 200`	Configures rate limits in packets per second for packets copied to the supervisor module for access list logging. The range is from 0 to 30000.
Step 3	**hardware rate-limiter copy** *packets* **Example:** `switch(config)# hardware rate-limiter copy 30000`	Configures rate limits in packets per second for data and control packets copied to the supervisor module. The range is from 0 to 30000. **Note** Layer 3 control, multicast direct-connect, and ARP request packets are controlled by the Layer 2 copy rate limiter. The first two types of packets are also controlled by Layer 3 rate limiters, and the last two types are also subject to control plane policing (CoPP).
Step 4	**hardware rate-limiter layer-2 mcast-snooping** *packets* **Example:** `switch(config)# hardware rate-limiter layer-2`	Configures rate limits in packets per second for Layer 2 multicast-snooping packets. The range is from 0 to 30000.

	Command or Action	Purpose
	`mcast-snooping 20000`	
Step 5	**hardware rate-limiter layer-2 port-security** *packets* **Example:** `switch(config)# hardware rate-limiter layer-2 port-security 100000`	Configures rate limits in packets per second for port-security packets. The range is from 0 to 30000.
Step 6	**hardware rate-limiter layer-2 storm-control** *packets* **Example:** `switch(config)# hardware rate-limiter layer-2 storm-control 10000`	Configures rate limits in packets per second for broadcast, multicast, and unknown unicast storm-control traffic. The range is from 0 to 30000.
Step 7	**hardware rate-limiter layer-2 vpc-low** *packets* **Example:** `switch(config)# hardware rate-limiter layer-2 vpc-low 10000`	Configures rate limits in packets per second for Layer 2 control packets over the VPC low queue. The range is from 0 to 30000.
Step 8	**hardware rate-limiter layer-3 control** *packets* **Example:** `switch(config)# hardware rate-limiter layer-3 control 20000`	Configures rate limits in packets per second for Layer 3 control packets. The range is from 0 to 30000.
Step 9	**hardware rate-limiter layer-3 glean** *packets* **Example:** `switch(config)# hardware rate-limiter layer-3 glean 200`	Configures rate limits in packets per second for Layer 3 glean packets. The range is from 0 to 30000.
Step 10	**hardware rate-limiter layer-3 glean-fast** *packets* **Example:** `switch(config)# hardware rate-limiter layer-3 glean-fast 500`	Configures rate limits in packets per second for Layer 3 glean fast-path packets. This command sends packets to Glean fast path optimizes the processing of glean packets by the supervisor. Specifically, the line card provides the information needed to trigger an ARP within the packet and relieves the supervisor from having to look up this information. The packets sent to the supervisor using the glean fast path are rate limited **Note** Glean fast path is enabled by default. If glean fast-path programming does not occur due to adjacency resource exhaustion, the system falls back to regular glean programming.
Step 11	**hardware rate-limiter layer-3 mtu** *packets* **Example:** `switch(config)# hardware rate-limiter layer-3 mtu 1000`	Configures rate limits in packets per second for Layer 3 MTU failure redirected packets. The range is from 0 to 30000.
Step 12	**hardware rate-limiter layer-3 multicast** *packets*	Configures rate limits in packets per second for Layer 3 multicast packets in packets per second.

	Command or Action	Purpose
	Example: `switch(config)# hardware rate-limiter layer-3 multicast 20000`	The range is from 0 to 30000.
Step 13	**hardware rate-limiter layer-3 ttl** *packets* **Example:** `switch(config)# hardware rate-limiter layer-3 ttl 1000`	Configures rate limits in packets per second for Layer 3 failed Time-to-Live redirected packets. The range is from 0 to 30000.
Step 14	**hardware rate-limiter receive** *packets* **Example:** `switch(config)# hardware rate-limiter receive 40000`	Configures rate limits in packets per second for packets redirected to the supervisor module. The range is from 0 to 30000.

Step 15	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 16	(Optional) **show hardware rate-limiter** [**access-list-log** | **copy** | **layer-2** {**mcast-snooping** | **port-security** | **storm-control** | **vpc-low**} | **layer-3** {**control** | **glean** | **glean-fast** | **mtu** | **multicast** | **ttl**} | **module** *module* | **receive**] **Example:** `switch# show hardware rate-limiter`	Displays the rate limit configuration.
Step 17	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

26.8 Monitoring Rate Limits

You can monitor rate limits.

SUMMARY STEPS

1. **show hardware rate-limiter** [**access-list-log** | **copy** | **layer-2** {**mcast-snooping** | **port-security** | **storm-control** | **vpc-low**} | **layer-3** {**control** | **glean** | **glean-fast** | **mtu** | **multicast** | **ttl**} | **module** *module* | **receive**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show hardware rate-limiter** [**access-list-log** | **copy** | **layer-2** {**mcast-snooping** | **port-security** | **storm-control** | **vpc-low**} | **layer-3** {**control** | **glean** | **glean-fast** | **mtu** | **multicast** | **ttl**} | **module** *module* | **receive**]	Displays the rate limit statistics.

	Example: `switch# show hardware rate-limiter layer-3 glean`	

26.9 Clearing the Rate Limit Statistics

You can clear the rate limit statistics.

SUMMARY STEPS

1. **clear hardware rate-limiter** {**all** | **access-list-log** | **copy** | **layer-2** {**mcast-snooping** | **port-security** | **storm-control** | **vpc-low**} | **layer-3** {**control** | **glean** | **glean-fast** | **mtu** | **multicast** | **ttl**} |**receive**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**clear hardware rate-limiter** {**all** | **access-list-log** | **copy** | **layer-2** {**mcast-snooping** | **port-security** | **storm-control** | **vpc-low**} | **layer-3** {**control** | **glean** | **glean-fast** | **mtu** | **multicast** | **ttl**} | **receive**} **Example:** `switch# clear hardware rate-limiter`	Clears the rate limit statistics.

26.10 Verifying the Rate Limit Configuration

To display the rate limit configuration information, perform the following tasks:

Command	Purpose
show hardware rate-limiter [**access-list-log** | **copy** | **layer-2** {**mcast-snooping** | **port-security** | **storm-control** | **vpc-low**} | **layer-3** {**control** | **glean** | **glean-fast** | **mtu** | **multicast** | **ttl**} | **module** *module* | **receive**]	Displays the rate limit configuration.

For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Security Command Reference*.

26.11 Configuration Examples for Rate Limits

The following example shows how to configure rate limits:

```
switch(config)# hardware rate-limiter layer-3 control 20000
switch(config)# hardware rate-limiter copy 30000
```

26.12 Additional References for Rate Limits

This section includes additional information related to implementing rate limits.

Related Documents

Related Topic	Document Title
Inspur INOS	*Inspur INOS Licensing Guide*

Licensing	
Command reference	*Inspur CN12700 Series INOS Security Command Reference*

26.13 Feature History for Rate Limits

This table lists the release history for this feature.

Table 55 : Feature History for Rate Limits

Feature Name	Releases	Feature Information	
Rate limits	8.2(3)	Added support for Layer 3 glean fast-path packets.	
Rate limits	8.2(3)	Added support for F3Series modules.	
Rate limits	8.2(3)	No change from Release 8.2(3)	

CHAPTER 27 Monitoring System Security

This chapter describes System Security Monitoring feature. This chapter includes the following sections:

- Finding Feature Information
- Overview of System Security Monitoring
- Additional References for Monitoring System Security
- Feature History for Monitoring System Security

27.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

27.2 Overview of System Security Monitoring

The security features in Inspur INOS provides resilience against attacks. From Inspur INOS Release 8.2(3), the system security monitoring functionality provides status for the following security features:

- XSPACE — An operating system capability to enforce mutual exclusivity between execution and write permissions. This capability prevents an attacker from executing malicious code by removing executable permissions in program data areas, such as the heap and the stack.
- Address Space Layout Randomization (ASLR) — Randomizes memory segment of a program when it is loaded to run. Randomization makes it statistically impossible for an attacker to predict a target address to jump by using Return Oriented Programming (ROP) technique.
- Object Size Checking (OSC) — A compiler technique to protect against buffer overflow. During run time, a buffer overflow may be detected and logged as DATACORRUPTION-DATAINCONSISTENCY errors.
- SafeC — Enhances the security of a new software. SafeC provides enhanced boundary checking as an alternative to certain C library functions. SafeC constraint violations are reported as DATACORRUPTION-DATAINCONSISTENCY errors.

Additionally, system configuration and capability for these security features are being monitored. If an unexpected negative change occurs, a critical syslog message is issued.

27.2.1 Displaying Information About System Security Monitoring

Use the following commands to display runtime integrity information:

- **show security system state -** Displays the status of system related security features.
- **show data-corruption -** Displays the DATACORRUPTION-DATAINCONSISTENCY errors collected from all running processes by OSC and SafeC techniques during runtime.

Displaying System Security Status

The following example displays the status of system related security features.

```
switch# show security system state
  XSPACE:
        Non-Executable stack: Yes
        Non-Executable heap: Yes
        Non-Writable text:      Yes
  ASLR:
```

```
          ASLR enabled:           Yes
          CVE-offset2lib Patch:   Present
          Randomization entropy:  Good
OSC:
          Version:                1.0.0
          Version:                3.0.1
```

This output displays information about the following fields:

- Non-Executable stack – Indicates whether system prevents execution from stack.
- Non-Executable heap – Indicates whether system prevents execution from heap.
- Non-Writable text – Indicates whether system prevents text section to be writable.
- ASLR enabled – Indicates whether ASLR is enabled in Linux kernel and system has capability to randomize all memory sections for binaries compiled with PIC/PIE flags.
- CVE-offset2lib Patch – Indicates whether Offset2lib patch is in kernel, so that randomized memory segment for text and data are not adjacent to libraries.
- Randomization entropy – Indicates whether entropy of randomization is sufficient.
- OSC version – Indicates the version of OSC library used by applications.
- SafeC version – Indicates the version of SafeC library used by applications.

Displaying OSC and SafeC Events

The following example displays the DATACORRUPTION-DATAINCONSISTENCY errors collected from all running processes by runtime OSC and SafeC techniques.

```
switch# show data-corruption
DATACORRUPTION-DATAINCONSISTENCY: -Traceback= vmtracker libhmm_dll.so+0x1b4d0 libhmm.so+0x2cf0
 libhmm_dll.so +0x1ba0a libhmm_dll.so+0x1c9e7 libhmm.so+0x2f49 +0x209d0
libvmtracker.so+0x4d586 libvmtracker.so+0x9b0c1 libvmtracker.so+0x43154 libvmtracker.so+0x42c
 happened 20 times since Mon Feb 15 09:05:20 2016
DATACORRUPTION-DATAINCONSISTENCY: -Traceback= hmm +0x40faf +0xbf870 +0xc0b4c +0x40292
+0xa37fa +0xa9f29 +0xc05aa +0xc060e +0xc0765 +0x42c35 +0x2c339 librsw.so+0xacc33
libpthread.so.0+0x6b75 libc.so.6+0xee02e happened 1 time since Fri Feb 12 00:01:16 2016
```

27.3 Additional References for Monitoring System Security

This section includes additional information related to monitoring system security.

Related Documents

Related Topic	**Document Title**
Inspur INOS Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*

27.4 Feature History for Monitoring System Security

This table lists the release history for this feature.

Table 56： Feature History for Monitoring System Security

Feature Name	**Release**	**Feature Information**
Monitoring System Security	8.2(3)	This feature was introduced. The following commands were introduced: • **show security system state**

		• **show data-corruption**

CHAPTER 28 Software Integrity Assurance

This chapter describes Runtime Integrity Assurance feature. This chapter includes the following sections:

- Finding Feature Information
- Overview of Runtime Integrity Assurance
- Additional References for Software Integrity Assurance
- Feature History for Software Integrity Assurance

28.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

28.2 Overview of Runtime Integrity Assurance

The Runtime Integrity Assurance feature provides assurance about the authenticity of the Inspur INOS system and its components. This feature ensures that the system is not exposed to any tampered code by measuring the Inspur INOS system and its components. Use CLI and NX-API to access the measurement of the Inspur INOS components on the Inspur CN switch. You can verify the authenticity of the Inspur INOS components by comparing the measurements against Known Good Values (KGVs) that are available on Inspur Connection Online (CCO) for the corresponding Inspur INOS release.

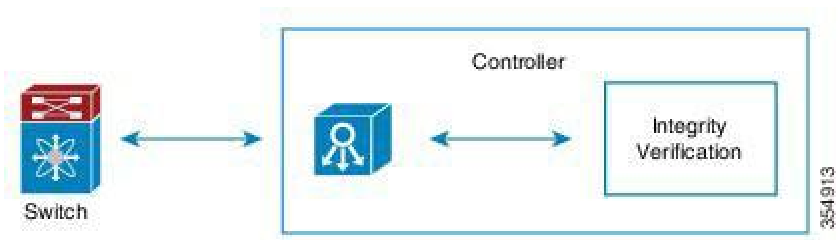
Runtime Integrity Assurance feature is enabled by default and cannot be disabled. However, verification at the controller is optional. In this case, you can access the measurements by using the CLI and compare the measurements against KGVs manually.

28.2.1 How Runtime Integrity Assurance Works

The security features in Inspur INOS provides resilience against attacks. From Inspur INOS Release8.2(3), the system security monitoring functionality provides status for the following security features:

Runtime integrity assurance involves two entities, namely, a switch and a controller. An integrity verification functionality is also embedded within the controller. This integrity verification entity within the controller analyzes the integrity data received from a switch.

Figure 38： Runtime Integrity Assurance on Inspur CN12700 Series Switch



On a switch, measurement of the running software is performed. This is carried out when a file is loaded for execution. The measurements are available through the CLI .

You can schedule verification at recurring intervals on the controller. Additionally, the controller collects the measurements from a switch and compare them against the KGVs. For more information, see the *Inspur DCNM Fundamentals Guide*.

28.2.2 Manual Verification of Files

Runtime integrity assurance through the controller is preferred for verification of files. However, you can also verify files manually by using the CLI.

To manually verify files, log in to CCO and download the KGVs. You can manually compare the hashes, which have been dumped through CLI, with the KGVs.

To display runtime integrity information, use one of the following commands:

- **show software integrity total -** Displays the number of measurements available in runtime integrity hash digests.
- **show software integrity** *index* **-** Displays hash digest entries by specifying the starting index value.

Displaying Information About Runtime Integrity Assurance

The following example shows how to display the number of measurements available in hash digests:

```
switch# show software integrity total
1092
```

The following example shows how to display the hash digest entries:

```
switch# show software integrity index 0
index pcr template-hash template-name al
gorithm:filedata-hash filena
me-hint
--------------------------------------------------------------------------------
-----------------------------------------------------------------------------
reference: 1481115089
1 10 1d8d532d463c9f8c205d0df7787669a85f93e260 ima-ng sh
a1:0000000000000000000000000000000000000000 boot_a
ggregate
2 10 1cb9d1e2795a75857f70d6a23cb77e4843467617 ima-ng sh
a256:850c63f1b32f19b2dcde9fa199a83da920c9e377e1e2dc52a6c7fdd045a21475 /etc/r
c.d/rcS.d/S98admin-login
3 10 95929573f5252fa80ad4bfb3b6dd644c5617d359 ima-ng sh
a256:1c684d45641dd23e1b2a763006030b9be46d8309581876c7a34feee1c87e037c /bin/b
ash
```

28.3 Additional References for Software Integrity Assurance

This section includes additional information related to Software Integrity Assurance feature.

Related Documents

Related Topic	Document Title
Inspur INOS Licensing	*Inspur INOS Licensing Guide*
Command reference	*Inspur CN12700 Series INOS Security Command Reference*

28.4 Feature History for Software Integrity Assurance

This table lists the release history for this feature.

Table 57：Feature History for Software Integrity Assurance

Feature Name	Release	Feature Information
Runtime Integrity Assurance	8.2(3)	This feature was introduced. The following command was introduced: • **show software integrity total**