Cisco Nexus 3400-S NX-OS Security Configuration Guide, Release 9.3(x)

First Published: 2019-12-23

Last Modified: 2021-03-31

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: http://www.cisco.com/go/softwareterms.Cisco product warranty information is available at http://www.cisco.com/go/warranty. US Federal Communications Commission Notices are found here http://www.cisco.com/c/en/us/products/us-fcc-notice.html.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com go trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2019–2021 Cisco Systems, Inc. All rights reserved.

CONTENTS

Trademarks ?

PREFACE

Preface xvii

Audience xvii

Document Conventions xvii

Related Documentation for Cisco Nexus 3000 Series Switches xviii

Documentation Feedback xviii

Communications, Services, and Additional Information xviii

CHAPTER 1

New and Changed Information 1

New and Changed Information 1

CHAPTER 2

Overview 3

Licensing Requirements 3

Authentication, Authorization, and Accounting 3

RADIUS and TACACS+ Security Protocols 4

LDAP 4

SSH and Telnet 5

User Accounts and Roles 5

IP ACLs 5

MAC ACLs 5

VACLs 5

DHCP Snooping 6

Password Encryption 6

Keychain Management 6

Control Plane Policing 6

Rate Limits 6
Virtual Device Contexts 7

CHAPTER 3 Configuring AAA 9
About AAA 9
Prerequisites for AAA 9
Guidelines and Limitations for AAA 10
Default Settings for AAA 10
Configuring AAA 10
Process for Configuring AAA 11
Configuring Console Login Authentication Methods 11
Configuring Default Login Authentication Methods 13
Disabling Fallback to Local Authentication 14
Enabling the Default User Role for AAA Authentication 15
Enabling Login Authentication Failure Messages 16
Logging Successful and Failed Login Attempts 17
Enabling CHAP Authentication 18
Enabling MSCHAP or MSCHAP V2 Authentication 19
Configuring AAA Accounting Default Methods 21
Using AAA Server VSAs with Cisco NX-OS Devices 22
Configuring Secure Login Features 23
Monitoring and Clearing the Local AAA Accounting Log 23
Verifying the AAA Configuration 23
Configuration Examples for AAA 24
Configuration Examples for Login Parameters 24
Configuration Examples for the Password Prompt Feature 25
Additional References for AAA 25

CHAPTER 4 Configuring RADIUS 27
About RADIUS 27
RADIUS Network Environments 27
RADIUS Operation 28
RADIUS Server Monitoring 28
Vendor-Specific Attributes 29

About RADIUS Change of Authorization 30
Session Reauthentication 31
Session Termination 31
Prerequisites for RADIUS 31
Guidelines and Limitations for RADIUS 31
Guidelines and Limitations for RADIUS Change of Authorization 32
Default Settings for RADIUS 32
Configuring RADIUS Servers 32
RADIUS Server Configuration Process 33
Configuring RADIUS Server Hosts 33
Configuring Global RADIUS Keys 34
Configuring a Key for a Specific RADIUS Server 35
Configuring RADIUS Server Groups 36
Configuring the Global Source Interface for RADIUS Server Groups 38
Allowing Users to Specify a RADIUS Server at Login 38
Configuring the Global RADIUS Transmission Retry Count and Timeout Interval 40
Configuring the RADIUS Transmission Retry Count and Timeout Interval for a Server 41
Configuring Accounting and Authentication Attributes for RADIUS Servers 42
Configuring Global Periodic RADIUS Server Monitoring 44
Configuring Periodic RADIUS Server Monitoring on Individual Servers 45
Configuring the RADIUS Dead-Time Interval 46
Configuring One-Time Passwords 47
Manually Monitoring RADIUS Servers or Groups 48
Enabling or Disabling Dynamic Author Server 49
Configuring RADIUS Change of Authorization 49
Verifying RADIUS Change of Authorization Configuration 50
Verifying the RADIUS Configuration 50
Monitoring RADIUS Servers 51
Clearing RADIUS Server Statistics 51
Configuration Examples of RADIUS Change of Authorization 52
Configuration Example for RADIUS 52
Additional References for RADIUS 52

CHAPTER 5 **Configuring TACACS+** 53

About TACACS+ 53
TACACS+ Advantages 53
TACACS+ Operation for User Login 54
Default TACACS+ Server Encryption Type and Secret Key 54
Command Authorization Support for TACACS+ Servers 55
TACACS+ Server Monitoring 55
Vendor-Specific Attributes for TACACS+ 56
Cisco VSA Format for TACACS+ 56
Prerequisites for TACACS+ 57
Guidelines and Limitations for TACACS+ 57
Default Settings for TACACS+ 57
One-Time Password Support 58
Configuring TACACS+ 58
TACACS+ Server Configuration Process 58
Enabling TACACS+ 58
Configuring TACACS+ Server Hosts 59
Configuring Global TACACS+ Keys 60
Configuring a Key for a Specific TACACS+ Server 61
Configuring TACACS+ Server Groups 62
Configuring the Global Source Interface for TACACS+ Server Groups 63
Allowing Users to Specify a TACACS+ Server at Login 64
Configuring the Timeout Interval for a TACACS+ Server 65
Configuring TCP Ports 66
Configuring Global Periodic TACACS+ Server Monitoring 67
Configuring the TACACS+ Dead-Time Interval 69
Configuring ASCII Authentication 70
Configuring AAA Authorization on TACACS+ Servers 71
Configuring Command Authorization on TACACS+ Servers 72
Testing Command Authorization on TACACS+ Servers 74
Enabling and Disabling Command Authorization Verification 75
Permitting or Denying Commands for Users of Privilege Roles 75
Manually Monitoring TACACS+ Servers or Groups 76
Disabling TACACS+ 77
Monitoring TACACS+ Servers 77

Verifying the TACACS+ Configuration 78
Configuration Examples for TACACS+ 78
Additional References for TACACS+ 80

CHAPTER 6 **Configuring LDAP 81**
About LDAP 81
LDAP Authentication and Authorization 81
LDAP Operation for User Login 82
LDAP Server Monitoring 83
Vendor-Specific Attributes for LDAP 83
Cisco VSA Format for LDAP 84
Virtualization Support for LDAP 84
Prerequisites for LDAP 84
Default Settings for LDAP 84
Configuring LDAP 85
LDAP Server Configuration Process 85
Enabling or Disabling LDAP 85
Configuring LDAP Server Hosts 86
Configuring the RootDN for an LDAP Server 86
Configuring LDAP Server Groups 87
Configuring the Global LDAP Timeout Interval 89
Configuring TCP Ports 89
Configuring LDAP Search Maps 90
Configuring Periodic LDAP Server Monitoring 91
Configuring the Global LDAP Timeout Interval 92
Configuring AAA Authorization on LDAP Servers 93
Monitoring LDAP Servers 94
Clearing LDAP Server Statistics 94
Verifying the LDAP Configuration 95
Configuration Examples for LDAP 95
Additional References for LDAP 96

CHAPTER 7 **Configuring SSH and Telnet 97**
About SSH and Telnet 97

SSH Server 97
SSH Client 97
SSH Server Keys 98
SSH Authentication Using Digital Certificates 98
Telnet Server 99
Prerequisites for SSH and Telnet 99
Guidelines and Limitations for SSH and Telnet 99
Default Settings for SSH and Telnet 100
Configuring SSH 100
Generating SSH Server Keys 100
Specifying the SSH Public Keys for User Accounts 101
Specifying the SSH Public Keys in IETF SECSH Format 102
Specifying the SSH Public Keys in OpenSSH Format 102
Configuring a Maximum Number of SSH Login Attempts 103
Starting SSH Sessions 104
Starting SSH Sessions from Boot Mode 105
Configuration Example for SSH Passwordless File Copy 105
Configuring SCP and SFTP Servers 107
Configuring X.509v3 Certificate-Based SSH Authentication 108
Configuring Legacy SSH Algorithm Support 110
Changing the Default SSH Server Port 112
Clearing SSH Hosts 113
Disabling the SSH Server 113
Deleting SSH Server Keys 114
Clearing SSH Sessions 115
Configuring Telnet 115
Enabling the Telnet Server 115
Starting Telnet Sessions to Remote Devices 116
Clearing Telnet Sessions 116
Verifying the SSH and Telnet Configuration 117
Configuration Example for SSH 117
Configuration Example for SSH Passwordless File Copy 118
Configuration Example for X.509v3 Certificate-Based SSH Authentication 120
Additional References for SSH and Telnet 121

CHAPTER 8 Configuring User Accounts and RBAC 123
About User Accounts and RBAC 123
User Accounts 123
Characteristics of Strong Passwords 124
User Roles 124
User Role Rules 125
Guidelines and Limitations for User Accounts and RBAC 126
Default Settings for User Accounts and RBAC 126
Enabling Password-Strength Checking 127
Configuring User Accounts 127
Configuring Roles 129
Creating User Roles and Rules 129
Creating Feature Groups 132
Changing User Role Interface Policies 133
Changing User Role VLAN Policies 134
Changing User Role VRF Policies 135
About No Service Password-Recovery 137
Enabling No Service Password-Recovery 137
Verifying User Accounts and RBAC Configuration 138
Configuration Examples for User Accounts and RBAC 139
Additional References for User Accounts and RBAC 141

CHAPTER 9 Configuring IP ACLs 143
About ACLs 143
ACL Types and Applications 143
Order of ACL Application 145
About Rules 146
Protocols for IP ACLs and MAC ACLs 146
Source and Destination 146
Implicit Rules for IP and MAC ACLs 147
Additional Filtering Options 147
Sequence Numbers 148
Logical Operators and Logical Operation Units 149

Time Ranges 149
Policy-Based ACLs 150
Statistics and ACLs 151
About Per-Port Stats 151
Atomic ACL Updates 152
Session Manager Support for IP ACLs 152
ACL TCAM Regions 152
Prerequisites for IP ACLs 154
Guidelines and Limitations for IP ACLs 154
Default Settings for IP ACLs 157
Configuring IP ACLs 157
Creating an IP ACL 157
Changing an IP ACL 159
Creating a VTY ACL 160
Changing Sequence Numbers in an IP ACL 161
Removing an IP ACL 162
Configuring ACL TCAM Region Sizes 163
Configuring TCAM Carving 165
Configuring UDF-Based Router ACLs 166
Applying an IP ACL as a Router ACL 168
Applying an IP ACL as a Port ACL 169
Applying an IP ACL as a VACL 170
Configuring Per-Port Stats 170
Verifying the IP ACL Configuration 171
Monitoring and Clearing IP ACL Statistics 172
Configuration Examples for IP ACLs 172
Configuring Object Groups 173
Session Manager Support for Object Groups 173
Creating and Changing an IPv4 Address Object Group 173
Creating and Changing an IPv6 Address Object Group 174
Creating and Changing a Protocol Port Object Group 175
Removing an Object Group 177
Verifying the Object-Group Configuration 177
Configuring Time-Ranges 178

Session Manager Support for Time-Ranges 178
Creating a Time-Range 178
Changing a Time-Range 179
Removing a Time-Range 180
Changing Sequence Numbers in a Time Range 181
Verifying the Time-Range Configuration 182

CHAPTER 10 Configuring MAC ACLs 183
About MAC ACLs 183
Guidelines and Limitations for MAC ACLs 183
Default Settings for MAC ACLs 184
Configuring MAC ACLs 184
Creating a MAC ACL 184
Changing a MAC ACL 185
Changing Sequence Numbers in a MAC ACL 186
Removing a MAC ACL 187
Applying a MAC ACL as a Port ACL 187
Applying a MAC ACL as a VACL 188
Verifying the MAC ACL Configuration 189
Monitoring and Clearing MAC ACL Statistics 189
Configuration Example for MAC ACLs 189

CHAPTER 11 Configuring VLAN ACLs 191
About VLAN ACLs 191
VLAN Access Maps and Entries 191
VACLs and Actions 191
VACL Statistics 192
Prerequisites for VACLs 192
Guidelines and Limitations for VACLs 192
Default Settings for VACLs 193
Configuring VACLs 193
Creating a VACL or Adding a VACL Entry 193
Removing a VACL or a VACL Entry 194
Applying a VACL to a VLAN 195

Verifying the VACL Configuration 196
Monitoring and Clearing VACL Statistics 196
Configuration Example for VACLs 196
Additional References for VACLs 197

CHAPTER 12 **Configuring DHCP** 199
About DHCP Snooping 199
Trusted and Untrusted Sources 200
DHCP Snooping Binding Database 200
DHCP Snooping in a vPC Environment 201
Synchronizing DHCP Snooping Binding Entries 201
Packet Validation 201
DHCP Snooping Option 82 Data Insertion 202
About the DHCP Relay Agent 203
DHCP Relay Agent 203
DHCP Relay Agent Option 82 204
VRF Support for the DHCP Relay Agent 205
DHCP Smart Relay Agent 206
About the DHCPv6 Relay Agent 206
DHCPv6 Relay Agent 206
VRF Support for the DHCPv6 Relay Agent 206
About DHCP Client 207
Prerequisites for DHCP 207
Guidelines and Limitations for DHCP 207
Default Settings for DHCP 208
Configuring DHCP 209
Minimum DHCP Configuration 209
Enabling or Disabling the DHCP Feature 209
Configuring DHCP Snooping 210
Enabling or Disabling DHCP Snooping Globally 210
Enabling or Disabling DHCP Snooping on a VLAN 210
Enabling or Disabling DHCP Snooping MAC Address Verification 211
Enabling or Disabling Option 82 Data Insertion and Removal 212
Enabling or Disabling Strict DHCP Packet Validation 214

Configuring an Interface as Trusted or Untrusted 215
Enabling or Disabling DHCP Relay Trusted Port Functionality 216
Configuring an Interface as a DHCP Relay Trusted or Untrusted Port 216
Configuring all Interfaces as Trusted or Untrusted 218
Enabling or Disabling the DHCP Relay Agent 219
Enabling or Disabling Option 82 for the DHCP Relay Agent 219
Enabling or Disabling VRF Support for the DHCPv6 Relay Agent 221
Configuring DHCP Server Addresses on an Interface 222
Configuring the DHCP Relay Source Interface 223
Enabling or Disabling DHCP Smart Relay Globally 224
Enabling or Disabling DHCP Smart Relay on a Layer 3 Interface 225
Configuring DHCPv6 226
Enabling or Disabling the DHCPv6 Relay Agent 226
Enabling or Disabling VRF Support for the DHCPv6 Relay Agent 227
Configuring DHCPv6 Server Addresses on an Interface 228
Configuring the DHCP Relay Source Interface 229
Configuring IPv6 RA Guard 230
Enabling DHCP Client 231
Verifying the DHCP Configuration 232
Displaying IPv6 RA Guard Statistics 233
Displaying DHCP Snooping Bindings 233
Clearing the DHCP Snooping Binding Database 233
Monitoring DHCP 234
Clearing DHCP Snooping Statistics 234
Clearing DHCP Relay Statistics 234
Clearing DHCPv6 Relay Statistics 234
Configuration Examples for DHCP 234
Configuration Examples for DHCP Client 235
Additional References for DHCP 235

CHAPTER 13 **Configuring IPv6 First Hop Security 237**
Introduction to First-Hop Security 237
IPv6 Global Policies 238
IPv6 First-Hop Security Binding Table 238

Guidelines and Limitations of First Hop Security 238
About vPC First Hop Security Configuration 238
DHCP Relay On-stack 239
DHCP Relay on VPC Leg 239
DHCP Client Relay on Orphan Ports 240
RA Guard 241
Overview of IPv6 RA Guard 241
Guidelines and Limitations of IPv6 RA Guard 242
DHCPv6 Guard 242
Overview of DHCP—DHCPv6 Guard 242
Limitation of DHCPv6 Guard 242
IPv6 Snooping 243
Overview of IPv6 Snooping 243
Guidelines and Limitations for IPv6 Snooping 243
How to Configure IPv6 FHS 244
Configuring the IPv6 RA Guard Policy on the Device 244
Configuring IPv6 RA Guard on an Interface 245
Configuring DHCP—DHCPv6 Guard 246
Configuring IPv6 Snooping 249
Verifying and Troubleshooting IPv6 Snooping 251
Configuration Examples 252
Example: IPv6 RA Guard Configuration 252
Example: Configuring DHCP—DHCPv6 Guard 252
Example: Configuring IPv6 First-Hop Security Binding Table 252
Example: Configuring IPv6 Snooping 253
Additional References for IPv6 First-Hop Security 253

CHAPTER 14 **Configuring Password Encryption** 255
About AES Password Encryption and Primary Encryption Keys 255
Guidelines and Limitations for Password Encryption 255
Default Settings for Password Encryption 256
Configuring Password Encryption 256
Configuring a Primary Key and Enabling the AES Password Encryption Feature 256
Converting Existing Passwords to Type-6 Encrypted Passwords 257

Converting Type-6 Encrypted Passwords Back to Their Original States 257
Deleting Type-6 Encrypted Passwords 258
Verifying the Password Encryption Configuration 258
Configuration Examples for Password Encryption 258

CHAPTER 15 **Configuring Keychain Management** 261
About Keychain Management 261
Lifetime of a Key 261
Prerequisites for Keychain Management 262
Guidelines and Limitations for Keychain Management 262
Default Settings for Keychain Management 262
Configuring Keychain Management 263
Creating a Keychain 263
Removing a Keychain 263
Configuring a Primary Key and Enabling the AES Password Encryption Feature 264
Configuring Text for a Key 265
Configuring Accept and Send Lifetimes for a Key 266
Configuring a Key for OSPFv2 Cryptographic Authentication 268
Determining Active Key Lifetimes 269
Verifying the Keychain Management Configuration 269
Determining Active Key Lifetimes 270
Verifying the Keychain Management Configuration 270
Additional References for Keychain Management 270

CHAPTER 16 **Configuring Unicast RPF** 271
About Unicast RPF 271
Unicast RPF Process 272
Guidelines and Limitations for Unicast RPF 272
Default Settings for Unicast RPF 273
Configuring Unicast RPF 273
Configuration Examples for Unicast RPF 275
Verifying the Unicast RPF Configuration 276
Additional References for Unicast RPF 276

CHAPTER 17 **Configuring Control Plane Policing** 279

About CoPP 279
Control Plane Protection 280
Guidelines and Limitations for CoPP 280
Default Settings for CoPP 282
Configuring CoPP 282
Configuring a Control Plane Policy Map 282
Configuring the Control Plane Service Policy 284
Configuring the CoPP Scale Factor Per Line Card 285
Changing or Reapplying the Default CoPP Policy 286
Copying the CoPP Best Practice Policy 286
Verifying the CoPP Configuration 287
Displaying the CoPP Configuration Status 289
Monitoring CoPP 289
Monitoring CoPP with SNMP 290
Clearing the CoPP Statistics 290
Configuration Examples for CoPP 290
Additional References for CoPP 291

CHAPTER 18 **Configuring Rate Limits** 293

About Rate Limits 293
Guidelines and Limitations for Rate Limits 293
Default Settings for Rate Limits 294
Configuring Rate Limits 294
Monitoring Rate Limits 296
Clearing the Rate Limit Statistics 296
Verifying the Rate Limit Configuration 297
Configuration Examples for Rate Limits 297
Additional References for Rate Limits 297

Preface

This preface includes the following sections:

- Audience, on page xvii
- Document Conventions, on page xvii
- Related Documentation for Cisco Nexus 3000 Series Switches, on page xviii
- Documentation Feedback, on page xviii
- Communications, Services, and Additional Information, on page xviii

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Related Documentation for Cisco Nexus 3000 Series Switches

The entire Cisco Nexus 3000 Series switch documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/tsd-products-support-series-home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3k-docfeedback@cisco.com. We appreciate your feedback.

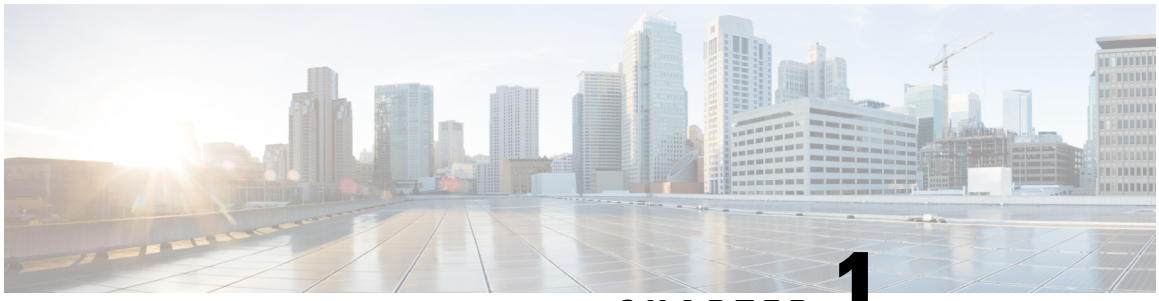
Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Preface



CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 3400-S NX-OS Security Configuration Guide, Release 9.3(x)*.

- New and Changed Information, on page 1

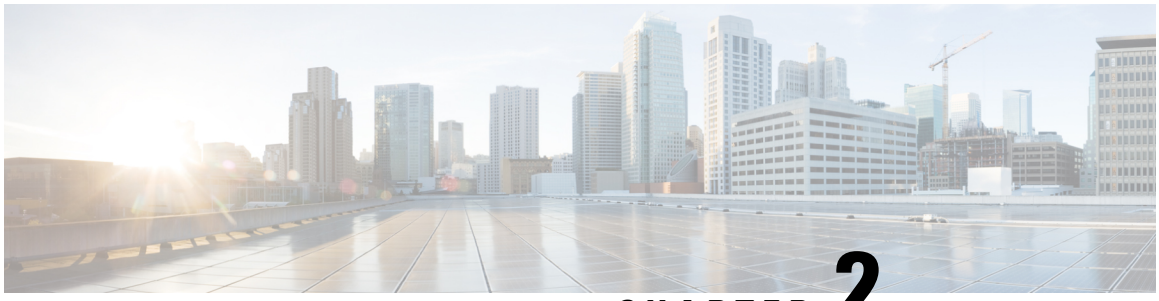
New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 3400-S NX-OS Security Configuration Guide, Release 9.3(x)* and tells you where they are documented.

Table 1: New and Changed Features for Cisco NX-OS Release 9.3(x)

Feature	Description	Changed in Release	Where Documented
Egress PACL	Added support for egress PACL on Cisco Nexus 3408-S and 3432D-S switches	9.3(5)	Guidelines and Limitations for IP ACLs, on page 154
No updates since Cisco NX-OS Release 9.2(2v)	Not applicable	Not applicable	Not applicable

New and Changed Information



CHAPTER 2

Overview

The Cisco NX-OS 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:

- Licensing Requirements, on page 3
- Authentication, Authorization, and Accounting, on page 3
- RADIUS and TACACS+ Security Protocols, on page 4
- LDAP, on page 4
- SSH and Telnet, on page 5
- User Accounts and Roles, on page 5
- IP ACLs, on page 5
- MAC ACLs, on page 5
- VACLs, on page 5
- DHCP Snooping, on page 6
- Password Encryption, on page 6
- Keychain Management, on page 6
- Control Plane Policing, on page 6
- Rate Limits, on page 6
- Virtual Device Contexts, on page 7

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide*.

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.

Note You can configure authentication outside of AAA. However, you must configure AAA if you want to use RADIUS or TACACS+, or if you want to configure a backup authentication method.

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 Cisco implementation, RADIUS clients run on Cisco 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.

LDAP

The Lightweight Directory Access Protocol (LDAP) provides centralized validation of users attempting to gain access to a Cisco NX-OS device. LDAP allows a single access control server (the LDAP daemon) to provide authentication and authorization independently.

SSH and Telnet

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

The SSH client in the Cisco NX-OS 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.

User Accounts and Roles

You can create and manage user accounts and assign roles that limit access to operations on the Cisco NX-OS 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.

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 Cisco NX-OS 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 Cisco NX-OS software applies the applicable default rule. The Cisco NX-OS software continues processing packets that are permitted and drops packets that are denied.

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 Cisco NX-OS 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 Cisco NX-OS software applies the applicable default rule. The Cisco NX-OS software continues processing packets that are permitted and drops packets that are denied.

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).

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 (IPSG) also use information stored in the DHCP snooping binding database.

Password Encryption

The Advanced Encryption Standard (AES) password encryption feature stores all existing and newly created clear-text passwords for supported applications (currently RADIUS and TACACS+) in the strong and reversible type-6 encrypted format. A primary encryption key is used to encrypt and decrypt the passwords. You can also use this feature to convert all existing weakly encrypted passwords to type-6 encrypted passwords.

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.

Control Plane Policing

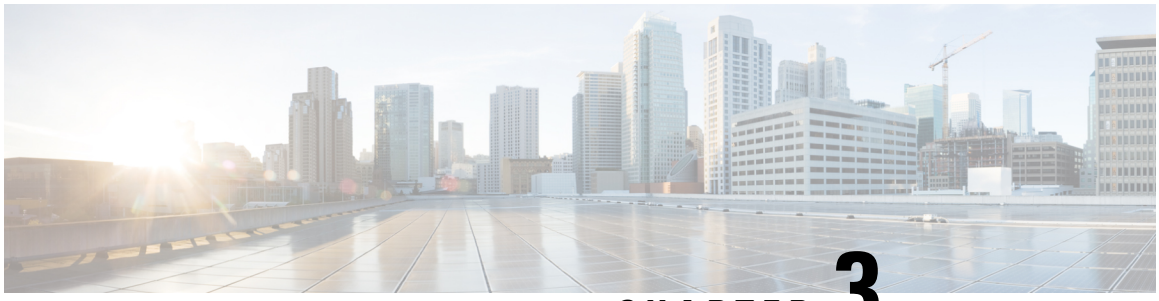
The Cisco NX-OS device provides control plane policing to prevent denial-of-service (DoS) attacks from impacting performance. The supervisor module of the Cisco NX-OS device has both the management plane and control plane and is critical to the operation of the network. Any disruption to the supervisor module would result in serious network outages. Excessive traffic to the supervisor module could overload it and slow down the performance of the entire Cisco NX-OS 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.

Rate Limits

Rate limits can prevent redirected packets for egress exceptions from overwhelming the supervisor module on a Cisco NX-OS device.

Virtual Device Contexts

Cisco NX-OS can segment operating system and hardware resources into virtual device contexts (VDCs) that emulate virtual devices. The Cisco Nexus 3400-S Series switches currently do not support multiple VDCs. All switch resources are managed in the default VDC.



CHAPTER 3

Configuring AAA

This chapter describes how to configure authentication, authorization, and accounting (AAA) on Cisco NX-OS devices.

This chapter includes the following sections:

- About AAA, on page 9
- Prerequisites for AAA, on page 9
- Guidelines and Limitations for AAA, on page 10
- Default Settings for AAA, on page 10
- Configuring AAA, on page 10
- Monitoring and Clearing the Local AAA Accounting Log , on page 23
- Verifying the AAA Configuration, on page 23
- Configuration Examples for AAA, on page 24
- Configuration Examples for Login Parameters, on page 24
- Configuration Examples for the Password Prompt Feature, on page 25
- Additional References for AAA, on page 25

About AAA

This section includes information about AAA on Cisco NX-OS devices.

Prerequisites for AAA

Remote AAA servers have the following prerequisites:

- Ensure that at least one RADIUS, TACACS+, or LDAP server is reachable through IP.
- Ensure that the Cisco NX-OS device is configured as a client of the AAA servers.
- Ensure that the secret key is configured on the Cisco NX-OS device and the remote AAA servers.
- Ensure that the remote server responds to AAA requests from the Cisco NX-OS device.

Guidelines and Limitations for AAA

AAA has the following guidelines and limitations:

- If you have a user account configured on the local Cisco NX-OS device that has the same name as a remote user account on an AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.
- Cisco Nexus 3400-S Series switches support the **aaa authentication login ascii-authentication** command only for TACACS+ (and not for RADIUS).
- If you modify the default login authentication method (without using the **local** keyword), the configuration overrides the console login authentication method. To explicitly configure the console authentication method, use the **aaa authentication login console** {**group** *group-list* [**none**] | **local** | **none**} command.

Default Settings for AAA

This table lists the default settings for AAA parameters.

Table 2: Default AAA Parameter Settings

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

Configuring AAA

This section describes the tasks for configuring AAA on Cisco NX-OS devices.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Note Cisco Nexus3400-S Series switches support the aaa authentication login ascii-authentication,command only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication command so that the default authentication, PAP is enabled. Otherwise, you will see syslog errors.

Process for Configuring AAA

Follow these steps to configure AAA authentication and accounting:

1. If you want to use remote RADIUS, TACACS+, or LDAP servers for authentication, configure the hosts on your Cisco NX-OS device.
2. Configure console login authentication methods.
3. Configure default login authentication methods for user logins.
4. Configure default AAA accounting default methods.

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
- Named subset of RADIUS, TACACS+, or LDAP servers
- Local database on the Cisco NX-OS device
- Username only (none)

The default method is local, but you have the option to disable it.

Note The **group radius** and **group** *server-name* forms of the **aaa authentication** command refer to a set of previously defined RADIUS servers. Use the **radius-server host** command to configure the host servers. Use the **aaa group server radius** command to create a named group of servers.

Note If you perform a password recovery when remote authentication is enabled, local authentication becomes enabled for console login as soon as the password recovery is done. As a result, you can log into the Cisco NX-OS device through the console port using the new password. After login, you can continue to use local authentication, or you can enable remote authentication after resetting the admin password configured at the AAA servers. For more information about the password recovery process, see the *Cisco Nexus 3400-S NX-OS Troubleshooting Guide.*

Before you begin

Configure RADIUS, TACACS+, or LDAP server groups, as needed.

Procedure

	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. ***named-group*** Uses a named subset of RADIUS, TACACS+, or LDAP 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, unless fallback to local is disabled for the console login.

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.

Configuring Default Login Authentication Methods

The authentication methods include the following:

- Global pool of RADIUS servers
- Named subset of RADIUS, TACACS+, or LDAP servers
- Local database on the Cisco NX-OS device
- Username only

The default method is local, but you have the option to disable it.

Before you begin

Configure RADIUS, TACACS+, or LDAP server groups, as needed.

Procedure

	Command or Action	Purpose

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

Step 2	**aaa authentication login default** {**group** *group-list* [**none**] | **local** | **none**} **Example:** `switch(config)# aaa authentication login default group radius`	Configures the default authentication methods. 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 a named subset of RADIUS, TACACS+, or LDAP 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 login method is **local**, which is used when no methods are configured or when all the configured methods fail to respond, unless fallback to local is disabled for the console login. You can configure one of the following: • AAA authentication groups • AAA authentication groups with no authentication

	Command or Action	Purpose
		• 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.

Disabling Fallback to Local Authentication

By default, if remote authentication is configured for console or default login and all AAA servers are unreachable (resulting in an authentication error), the Cisco NX-OS device falls back to local authentication to ensure that users are not locked out of the device. However, you can disable fallback to local authentication in order to increase security.

Caution Disabling fallback to local authentication can lock your Cisco NX-OS device, forcing you to perform a password recovery in order to gain access. To prevent being locked out of the device, we recommend that you disable fallback to local authentication for only the default login or the console login, not both.

Before you begin

Configure remote authentication for the console or default login.

Procedure

	Command or Action	Purpose

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

Step 2	**no aaa authentication login** {**console** | **default**} **fallback error local** **Example:** `switch(config)# no aaa authentication login console fallback error local`	Disables fallback to local authentication for the console or default login if remote authentication is configured and all AAA servers are unreachable. The following message appears when you disable fallback to local authentication: `"WARNING!!! Disabling fallback can lock your switch."`

Step 3	(Optional) **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 and 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.

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 Cisco NX-OS 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.

Procedure

	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.

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.
```

Procedure

	Command or Action	Purpose

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

Step 2	**aaa authentication login error-enable** **Example:**	Enables login authentication failure messages. The default is disabled.

	Command or Action	Purpose
	`switch(config)# aaa authentication login error-enable`	

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.

Logging Successful and Failed Login Attempts

You can configure the switch to log all successful and failed login attempts to the configured syslog server.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `switch# configure terminal`	Enters global configuration mode.
Step 2	Required: [**no**] **login on-failure log** **Example:** `switch(config)# login on-failure log`	Logs all failed authentication messages to the configured syslog server. With this configuration, the following syslog message appears after the failed login: AUTHPRIV-3-SYSTEM_MSG: pam_aaa:Authentication failed for user admin from 172.22.00.00 **Note** When logging level authpriv is 6, additional Linux kernel authentication messages appear along with the previous message. If these additional messages need to be ignored, the authpriv value should be set to 3.
Step 3	Required: [**no**] **login on-success log** **Example:** `switch(config)# login on-success log`	Logs all successful authentication messages to the configured syslog server. With this configuration, the following syslog message appears after the successful login:

	Command or Action	Purpose
		AUTHPRIV-6-SYSTEM_MSG: pam_aaa:Authentication success for user admin from 172.22.00.00 **Note** When logging level authpriv is 6, additional Linux kernel authentication messages appear along with the previous message.
Step 4	(Optional) **show login on-failure log** **Example:** `switch(config)# show login on-failure log`	Displays whether the switch is configured to log failed authentication messages to the syslog server.
Step 5	(Optional) **show login on-successful log** **Example:** `switch(config)# show login on-successful log`	Displays whether the switch is configured to log successful authentication messages to the syslog server.
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 CHAP Authentication

The Cisco NX-OS software supports the Challenge Handshake Authentication Protocol (CHAP), a challenge-response authentication protocol that uses the industry-standard Message Digest (MD5) hashing scheme to encrypt responses. You can use CHAP for user logins to a Cisco NX-OS device through a remote authentication server (RADIUS or TACACS+).

By default, the Cisco NX-OS device uses Password Authentication Protocol (PAP) authentication between the Cisco NX-OS device and the remote server. If you enable CHAP, you need to configure your RADIUS or TACACS+ server to recognize the CHAP vendor-specific attributes (VSAs).

This table shows the RADIUS and TACACS+ VSAs required for CHAP.

Table 3: CHAP RADIUS and TACACS+ VSAs

Vendor-ID Number	Vendor-Type Number	VSA	Description
311	11	CHAP-Challenge	Contains the challenge sent by an AAA server to a CHAP user. It can be used in both Access-Request and Access-Challenge packets.
211	11	CHAP-Response	Contains the response value provided by a CHAP user in response to the challenge. It is used only in Access-Request packets.

Before you begin

Disable AAA ASCII authentication for logins.

Procedure

	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 chap enable** **Example:** `switch(config)# aaa authentication login chap enable`	Enables CHAP authentication. The default is disabled. **Note** You cannot enable both CHAP and MSCHAP or MSCHAP V2 on your Cisco NX-OS device.

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

Step 5	(Optional) **show aaa authentication login chap** **Example:** `switch# show aaa authentication login chap`	Displays the CHAP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Enabling MSCHAP or MSCHAP V2 Authentication

Microsoft Challenge Handshake Authentication Protocol (MSCHAP) is the Microsoft version of CHAP. The Cisco NX-OS software also supports MSCHAP Version 2 (MSCHAP V2). You can use MSCHAP for user logins to a Cisco NX-OS device through a remote authentication server (RADIUS or TACACS+). MSCHAP V2 only supports user logins to a Cisco NX-OS 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.

Note The Cisco NX-OS software may display the following message:

" Warning: MSCHAP V2 is supported only with Radius."

This warning message is informational only and does not affect MSCHAP V2 operation with RADIUS.

By default, the Cisco NX-OS device uses Password Authentication Protocol (PAP) authentication between the Cisco NX-OS 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.

Procedure

	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 Cisco NX-OS device.

	Command or Action	Purpose

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.

Configuring AAA Accounting Default Methods

Cisco NX-OS software supports TACACS+ and RADIUS methods for accounting. Cisco NX-OS 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 Cisco NX-OS 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.

Note If you have configured server groups and the server groups do not respond, by default, the local database is used for authentication.

Before you begin

Configure RADIUS or TACACS+ server groups, as needed.

Procedure

	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. 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.

Using AAA Server VSAs with Cisco NX-OS Devices

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

Configuring Secure Login Features

Monitoring and Clearing the Local AAA Accounting Log

The Cisco NX-OS device maintains a local log for the AAA accounting activity. You can monitor this log and clear it.

Procedure

	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** [**logflash**] **Example:** `switch# clear aaa accounting log`	Clears the accounting log contents. The **logflash** keyword clears the accounting log stored in the logflash.

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** | **chap** | **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.

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
```

Configuration Examples for Login Parameters

The following example shows how to configure the switch to enter a 100-second quiet period if 3 failed login attempts is exceeded within 60 seconds. This example shows no login failures.

```
switch# configure terminal
switch(config)# login block-for 100 attempts 3 within 60
switch(config)# 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 3 login failures occur in 60 seconds or less,
 logins will be disabled for 100 seconds.

 Switch presently in Normal-Mode.
 Current Watch Window remaining time 45 seconds.
 Present login failure count 0.

switch(config)# show login failures
*** No logged failed login attempts with the device.***
```

The following example shows how to configure a quiet-mode ACL. All login requests are denied during the quiet period except hosts from the myacl ACL. This example also shows a login failure.

```
switch# configure terminal
switch(config)# login block-for 100 attempts 3 within 60
switch(config)# login quiet-mode access-class myacl

switch(config)# show login

 Switch is enabled to watch for login Attacks.
 If more than 3 login failures occur in 60 seconds or less,
 logins will be disabled for 100 seconds.

 Switch presently in Quiet-Mode.
 Will remain in Quiet-Mode for 98 seconds.
 Denying logins from all sources.

switch(config)# show login failures
Information about last 20 login failure's with the device.
--------------------------------------------------------------------------------
Username      Line          SourceIPAddr    Appname    TimeStamp
--------------------------------------------------------------------------------
asd           /dev/pts/0    171.70.55.158   login      Mon Aug  3 18:18:54 2015
qweq          /dev/pts/0    171.70.55.158   login      Mon Aug  3 18:19:02 2015
qwe           /dev/pts/0    171.70.55.158   login      Mon Aug  3 18:19:08 2015
```

```
--------------------------------------------------------------------------------
```

Configuration Examples for the Password Prompt Feature

The following example shows how to configure the switch to prompt the user to enter a password after she enters the **username** command and the error message that displays if she does not enter a password.

```
switch# configure terminal
switch(config)# password prompt username
Password prompt username is enabled.
After providing the required options in the username command, press enter.
User will be prompted for the username password and password will be hidden.
Note: Choosing password key in the same line while configuring user account, password will
 not be hidden.


switch(config)# username user1
Enter password:
Confirm password:
warning: password for user:user1 not set. S/he may not be able to login
```

The following example shows how to configure the switch to prompt the user to enter a password after she enters the **snmp-server user** command and the prompts that then display to the user.

```
switch# configure terminal
switch(config)# password prompt username
Password prompt username is enabled.
After providing the required options in the username command, press enter.
User will be prompted for the username password and password will be hidden.
Note: Choosing password key in the same line while configuring user account, password will
 not be hidden.


N9K-1(config)# snmp-server user user1
Enter auth md5 password (Press Enter to Skip):
Enter auth sha password (Press Enter to Skip):
```

Additional References for AAA

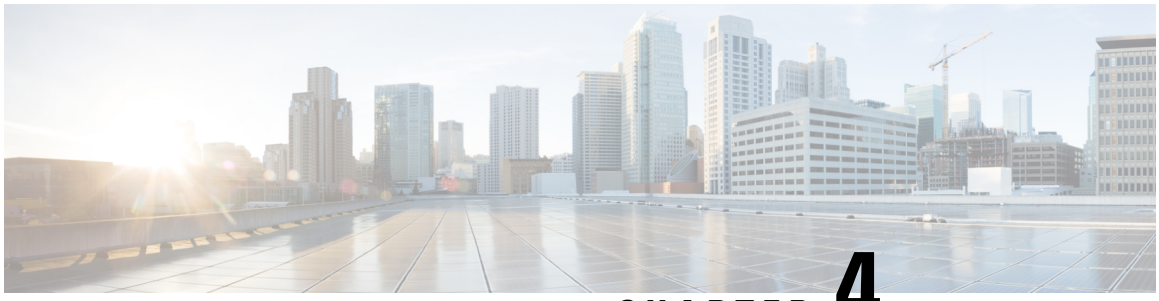
This section includes additional information related to implementing AAA.

Related Documents

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing 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.	—



CHAPTER 4

Configuring RADIUS

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

This chapter includes the following sections:

- About RADIUS, on page 27
- About RADIUS Change of Authorization, on page 30
- Prerequisites for RADIUS, on page 31
- Guidelines and Limitations for RADIUS, on page 31
- Guidelines and Limitations for RADIUS Change of Authorization, on page 32
- Default Settings for RADIUS, on page 32
- Configuring RADIUS Servers, on page 32
- Enabling or Disabling Dynamic Author Server, on page 49
- Configuring RADIUS Change of Authorization, on page 49
- Verifying RADIUS Change of Authorization Configuration, on page 50
- Verifying the RADIUS Configuration, on page 50
- Monitoring RADIUS Servers, on page 51
- Clearing RADIUS Server Statistics, on page 51
- Configuration Examples of RADIUS Change of Authorization, on page 52
- Configuration Example for RADIUS, on page 52
- Additional References for RADIUS, on page 52

About RADIUS

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

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 Cisco NX-OS 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 Cisco NX-OS device to better manage ports using their existing RADIUS solutions and to efficiently manage shared resources to offer different service-level agreements.

RADIUS Operation

When a user attempts to log in and authenticate to a Cisco NX-OS 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.

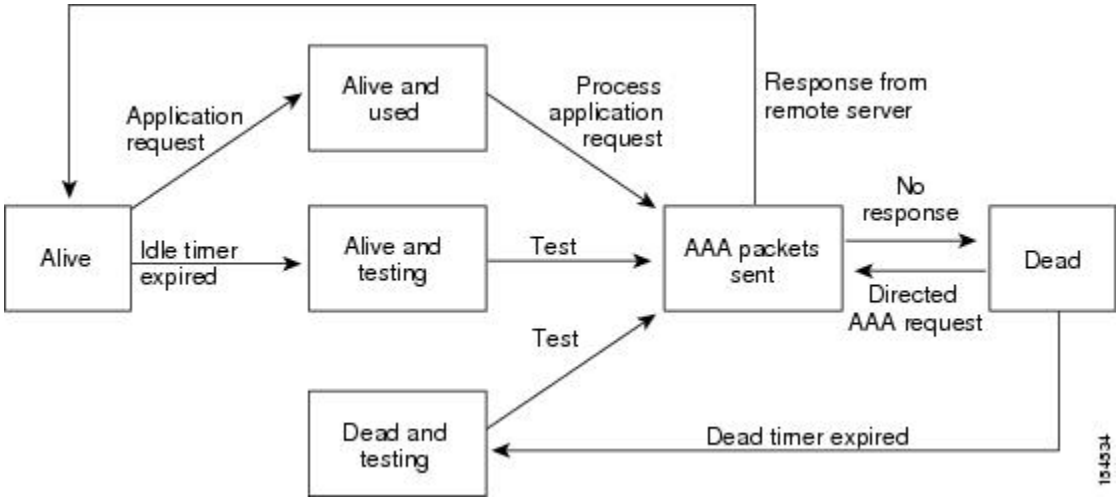
RADIUS Server Monitoring

An unresponsive RADIUS server can cause a delay in processing AAA requests. You can configure the Cisco NX-OS device to periodically monitor a RADIUS server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco NX-OS device marks unresponsive RADIUS servers as dead and does not send AAA requests to any dead RADIUS servers. The Cisco NX-OS 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 Cisco NX-OS 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.



Note The monitoring interval for alive servers and dead servers are different and can be configured by the user. The RADIUS server monitoring is performed by sending a test authentication request to the RADIUS server.

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

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

The protocol is a Cisco 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 Cisco NX-OS 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 Cisco NX-OS 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 Cisco NX-OS 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 network-admin, the value field would be network-operator network-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 Cisco Access Control Server (ACS):

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

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

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

```
Cisco-AVPair = shell:roles=\network-operator network-admin\

Cisco-AVPair = shell:roles*\network-operator network-admin\
```

Note When you specify a VSA as shell:roles*"network-operator network-admin" or "shell:roles*\"network-operator network-admin\"", this VSA is flagged as an optional attribute and other Cisco devices ignore this attribute.

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).

About RADIUS Change of Authorization

A standard RADIUS interface is typically used in a pulled model, in which the request originates from a device attached to a network and the response is sent from the queried servers. Cisco NX-OS sofware supports the RADIUS Change of Authorization (CoA) request defined in RFC 5176 that is used in a pushed model, in which the request originates from the external server to the device attached to the network, and enables the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

When Dot1x is enabled, the network device acts as the authenticator and is responsible for processing dynamic COA per session.

The following requests are supported:

- Session reauthentication
- Session termination

Session Reauthentication

To initiate session reauthentication, the authentication, authorization, and accounting (AAA) server sends a standard CoA-Request message that contains a Cisco VSA and one or more session identification attributes. The Cisco VSA is in the form of Cisco:Avpair="subscriber:command=reauthenticate".

The current session state determines the response of the device to the message in the following scenarios:

- If the session is currently authenticated by IEEE 802.1x, the device responds by sending an Extensible Authentication Protocol over LAN (EAPOL)-RequestId message to the server.
- If the session is currently authenticated by MAC authentication bypass (MAB), the device sends an access request to the server, passing the same identity attributes used for the initial successful authentication.
- If session authentication is in progress when the device receives the command, the device terminates the process and restarts the authentication sequence, starting with the method configured to be attempted first.

Session Termination

A CoA Disconnect-Request terminates the session without disabling the host port. CoA Disconnect-Request termination causes reinitialization of the authenticator state machine for the specified host, but does not restrict the host's access to the network.

If the session cannot be located, the device returns a Disconnect-NAK message with the "Session Context Not Found" error-code attribute.

If the session is located, but the NAS was unable to remove the session due to some internal error, the device returns a Disconnect-NAK message with the "Session Context Not Removable" error-code attribute.

If the session is located, the device terminates the session. After the session has been completely removed, the device returns a Disconnect-ACK message.

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 Cisco NX-OS device is configured as a RADIUS client of the AAA servers.

Guidelines and Limitations for RADIUS

RADIUS has the following guidelines and limitations:

- You can configure a maximum of 64 RADIUS servers on the Cisco NX-OS device.

- If you have a user account configured on the local Cisco NX-OS device that has the same name as a remote user account on an AAA server, the Cisco NX-OS software applies the user roles for the local user account to the remote user, not the user roles configured on the AAA server.
- Only the RADIUS protocol supports one-time passwords.
- Cisco Nexus 3400-S Series switches support the aaa authentication login ascii-authentication, command only for TACAAS+ and not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication command so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.

Guidelines and Limitations for RADIUS Change of Authorization

RADIUS Change of Authorization has the following guidelines and limitations:

- RADIUS Change of Authorization is supported on FEX.
- RADIUS change of Authorization is supported for VXLAN EVPN.

Default Settings for RADIUS

This table lists the default settings for RADIUS parameters.

Table 5: 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
Accounting port	1813
Idle timer interval	0 minutes
Periodic server monitoring username	test
Periodic server monitoring password	test

Configuring RADIUS Servers

This section describes how to configure RADIUS servers on a Cisco NX-OS device.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Note Cisco Nexus 3400-S Series switches support the aaa authentication login ascii-authentication command only for TACAAS+, but not for RADIUS. Ensure that you have disabled aaa authentication login ascii-authentication switch so that the default authentication, PAP, is enabled. Otherwise, you will see syslog errors.

RADIUS Server Configuration Process

1. Establish the RADIUS server connections to the Cisco NX-OS device.
2. Configure the RADIUS secret keys for the RADIUS servers.
3. If needed, configure RADIUS server groups with subsets of the RADIUS servers for AAA authentication methods.
4. If needed, configure any of the following optional parameters:
   - Dead-time interval
   - RADIUS server specification allowed at user login
   - Timeout interval
   - TCP port
5. (Optional) If RADIUS distribution is enabled, commit the RADIUS configuration to the fabric.

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.

Note By default, when you configure a RADIUS server IP address or hostname of the Cisco NX-OS device, the RADIUS server is added to the default RADIUS server group. You can also add the RADIUS server to another RADIUS server group.

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 Cisco NX-OS device is configured as a RADIUS client of the AAA servers.

Procedure

	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*} **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.

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.

Configuring Global RADIUS Keys

You can configure RADIUS keys for all servers used by the Cisco NX-OS device. A RADIUS key is a shared secret text string between the Cisco NX-OS 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.

Procedure

	Command or Action	Purpose

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

Step 2	**radius-server key** [**0** | **6** | **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**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS 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`
	`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.

Configuring a Key for a Specific RADIUS Server

You can configure a key on the Cisco NX-OS device for a specific RADIUS server. A RADIUS key is a secret text string shared between the Cisco NX-OS 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.

Procedure

	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** | **6** | **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**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS 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.

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.

Configuring RADIUS Server Groups

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 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.

Procedure

	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*
	Example:
	`switch(config)# aaa group server radius RadServer`
	`switch(config-radius)#`
		Creates a RADIUS server group and enters the RADIUS server group configuration submode for 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* | *hostname*} **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* | *hostname*} **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:**	Displays the RADIUS server group configuration.

	Command or Action	Purpose
	`switch(config)# show radius-server groups`	
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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 Cisco NX-OS software uses any available interface.

Procedure

	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.

Allowing Users to Specify a RADIUS Server at Login

By default, the Cisco NX-OS device forwards an authentication request based on the default AAA authentication method. You can configure the Cisco NX-OS 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.

Note If you enable the directed-request option, the Cisco NX-OS device uses only the RADIUS method for authentication and not the default local method.

Note User-specified logins are supported only for Telnet sessions.

Procedure

	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.

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:**	Copies the running configuration to the startup configuration.

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

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 Cisco NX-OS 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 Cisco NX-OS device waits for responses from RADIUS servers before declaring a timeout failure.

Procedure

	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.

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.

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

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

By default, a Cisco NX-OS 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 Cisco NX-OS device waits for responses from RADIUS servers before declaring a timeout failure.

Before you begin

Configure one or more RADIUS server hosts.

Procedure

	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*} **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* | *hostname*} **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.

	Command or Action	Purpose
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 Cisco NX-OS 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.

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.

Procedure

	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* | *hostname*} **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* | *hostname*} **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* | *hostname*} **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* | *hostname*} **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.

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.

Configuring Global Periodic RADIUS Server Monitoring

You can monitor the availability of all RADIUS servers without having to configure the test parameters for each server individually. Any servers for which test parameters are not configured are monitored using the global level parameters.

Note Test parameters that are configured for individual servers take precedence over global test parameters.

The global configuration parameters include the username and password to use for the servers and an idle timer. The idle timer specifies the interval in which a RADIUS server receives no requests before the Cisco NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Note To protect network security, we recommend that you use a username that is not the same as an existing username in the RADIUS database.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic RADIUS server monitoring is not performed.

Before you begin

Enable RADIUS.

Procedure

	Command or Action	Purpose

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

Step 2	**radius-server test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]} **Example:** `switch(config)# radius-server test username user1 password Ur2Gd2BH idle-time 3`	Specifies parameters for global 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, the idle timer value must be greater than 0.
Step 3	**radius-server deadtime** *minutes* **Example:**	Specifies the number of minutes before the Cisco NX-OS device checks a RADIUS server that was previously unresponsive. The default

	Command or Action	Purpose
	`switch(config)# radius-server deadtime 5`	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 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.

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 Cisco NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Note Test parameters that are configured for individual servers take precedence over global test parameters.

Note For security reasons, we recommend that you do not configure a test username that is the same as an existing user in the RADIUS database.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, the Cisco NX-OS device does not perform periodic RADIUS server monitoring.

Before you begin

Enable RADIUS.

Add one or more RADIUS server hosts.

Procedure

	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*} **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 Cisco NX-OS 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.

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 Cisco NX-OS 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.

Note When the dead-time interval is 0 minutes, RADIUS servers are not marked as dead even if they are not responding. You can configure the dead-time interval for a RADIUS server group.

Procedure

	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.

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.

Configuring One-Time Passwords

One-time password (OTP) support is available for Cisco NX-OS devices through the use of RSA SecurID token servers. With this feature, users authenticate to a Cisco NX-OS device by entering both a personal identification number (or one-time password) and the token code being displayed at that moment on their RSA SecurID token.

Note The token code used for logging into the Cisco NX-OS device changes every 60 seconds. To prevent problems with device discovery, we recommend using different usernames that are present on the Cisco Secure ACS internal database.

Before you begin

On the Cisco NX-OS device, configure a RADIUS server host and remote default login authentication.

Ensure that the following are installed:

- Cisco Secure Access Control Server (ACS) version 4.2
- RSA Authentication Manager version 7.1 (the RSA SecurID token server)
- RSA ACE Agent/Client

No configuration (other than a RADIUS server host and remote authentication) is required on the Cisco NX-OS device to support one-time passwords. However, you must configure the Cisco Secure ACS as follows:

1. Enable RSA SecurID token server authentication.
2. Add the RSA SecurID token server to the Unknown User Policy database.

Manually Monitoring RADIUS Servers or Groups

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

Procedure

	Command or Action	Purpose
Step 1	**test aaa server radius** {*ipv4-address* | *ipv6-address* | *hostname*} [**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.

Enabling or Disabling Dynamic Author Server

Procedure

	Command or Action	Purpose

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

Step 2	**aaa server radius dynamic-author** **Example:** `switch(config)# aaa server radius dynamic-author`	Enables the RADIUS dynamic author server. You can disable the RADIUS dynamic author server using the no form of this command.

Configuring RADIUS Change of Authorization

Before you begin

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **aaa server radius dynamic-author** **Example:** `switch(config)# aaa server radius dynamic-author`	Configures the switch as an AAA server to facilitate interaction with an external policy server. You can disable the RADIUS dynamic author and the associated clients using the no form of this command.
Step 3	[**no**] **client** {*ip-address* | **hostname** } [**server-key** [**0** | **7**] *string*] **Example:** `switch(config-locsvr-da-radius)# client 192.168.0.5 server-key cisco1`	Configures the IP address or the hostname of the AAA server client. Use the optional server-key keyword and string argument to configure the server key at the client level. You can remove the client server using the no form of this command. **Note** Configuring the server key at the client level overrides the server key that is configured at the global level.

	Command or Action	Purpose
Step 4	[**no**] **port** *port-number* **Example:** `switch(config-locsvr-da-radius)# port 3799`	Specifies the port on which a device listens to the RADIUS requests from the configured RADIUS clients. The port range is 1 - 65535. You can revert to the default port using the no form of this command. **Note** The default port for a packet of disconnect is 1700.
Step 5	[**no**] **server-key** [**0** | **7**] *string*	Configures the global RADIUS key to be shared between a device and the RADIUS clients. You can remove the server-key using the no form of this command.

Verifying RADIUS Change of Authorization Configuration

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

Command	Purpose
show running-config dot1x	Displays the dot1x configuration in the running configuration.
show running-config aaa	Displays the AAA configuration in the running configuration.
show running-config radius	Displays the RADIUS configuration in the running configuration.
show aaa server radius statistics	Displays the local RADIUS server statistics.
show aaa client radius statistics {*ip address* | *hostname* }	Displays the local RADIUS client statistics.
clear aaa server radius statistics	Clears the local RADIUS server statistics.
clear aaa client radius statistics {*ip address* | *hostname* }	Clears the local RADIUS client statistics.

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 Cisco Fabric Services distribution status and other details.

Command	Purpose
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 [*hostname* | *ipv4-address* | *ipv6-address*] [**directed-request** | **groups** | **sorted** | **statistics**]	Displays all configured RADIUS server parameters.

Monitoring RADIUS Servers

You can monitor the statistics that the Cisco NX-OS device maintains for RADIUS server activity.

Before you begin

Configure one or more RADIUS server hosts.

Procedure

	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.

Clearing RADIUS Server Statistics

You can display the statistics that the Cisco NX-OS device maintains for RADIUS server activity.

Before you begin

Configure RADIUS servers on the Cisco NX-OS device.

Procedure

	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 Cisco NX-OS device.

	Command or Action	Purpose
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.

Configuration Examples of RADIUS Change of Authorization

The following example shows how to configure RADIUS Change of Authorization:

```
radius-server host 10.77.143.170 key 7 "fewhg123" authentication accounting
aaa server radius dynamic-author
    client 10.77.143.170 vrf management server-key 7 "fewhg123"
```

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
```

Additional References for RADIUS

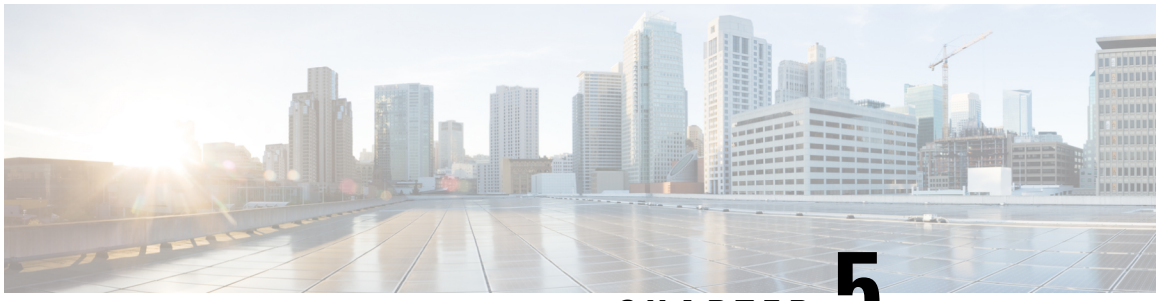
This section describes additional information related to implementing RADIUS.

Related Documents

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus 3400-S NX-OS 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.	—



CHAPTER 5

Configuring TACACS+

This chapter describes how to configure the Terminal Access Controller Access Control System Plus (TACACS+) protocol on Cisco NX-OS devices.

This chapter includes the following sections:

- About TACACS+, on page 53
- Prerequisites for TACACS+, on page 57
- Guidelines and Limitations for TACACS+, on page 57
- Default Settings for TACACS+, on page 57
- One-Time Password Support, on page 58
- Configuring TACACS+, on page 58

About TACACS+

The TACACS+ security protocol provides centralized validation of users attempting to gain access to a Cisco NX-OS 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 Cisco NX-OS 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. Cisco NX-OS devices provide centralized authentication using the TACACS+ protocol.

TACACS+ Advantages

TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the Cisco NX-OS 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.

TACACS+ Operation for User Login

When a user attempts a Password Authentication Protocol (PAP) login to a Cisco NX-OS device using TACACS+, the following actions occur:

Note TACACS+ allows an arbitrary conversation between the daemon and the user until the daemon receives enough information to authenticate the user. This action is usually done by prompting for a username and password combination, but may include prompts for other items, such as your mother's maiden name.

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

   ACCEPT
   User authentication succeeds and service begins. If the Cisco NX-OS 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 Cisco NX-OS device. If the Cisco NX-OS device receives an ERROR response, the Cisco NX-OS 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 Cisco NX-OS device. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.
3. If TACACS+ authorization is required, the Cisco NX-OS 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

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 Cisco NX-OS 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 Cisco NX-OS device to use.

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

Command Authorization Support for TACACS+ Servers

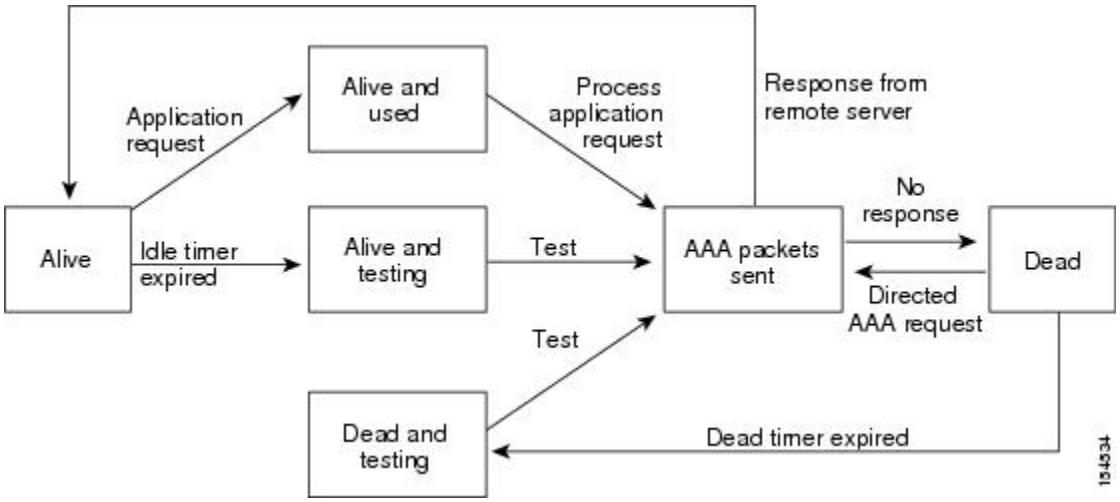
By default, command authorization is done against a local database in the Cisco NX-OS software when an authenticated user enters a command at the command-line interface (CLI). You can also verify authorized commands for authenticated users using TACACS+.

TACACS+ Server Monitoring

An unresponsive TACACS+ server can delay the processing of AAA requests. A Cisco NX-OS device can periodically monitor a TACACS+ server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco NX-OS device marks unresponsive TACACS+ servers as dead and does not send AAA requests to any dead TACACS+ servers. A Cisco NX-OS 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 Cisco NX-OS 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 monitoring.



Note The monitoring interval for alive servers and dead servers are different and can be configured by the user. The TACACS+ server monitoring is performed by sending a test authentication request to the TACACS+ server.

Vendor-Specific Attributes for TACACS+

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.

Cisco VSA Format for TACACS+

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

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

The protocol is a Cisco 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 Cisco NX-OS 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 Cisco NX-OS 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 Cisco NX-OS 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 network-admin, the value field would be network-operator network-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 Cisco ACS:

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

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

Note When you specify a VSA as shell:roles*"network-operator network-admin", this VSA is flagged as an optional attribute and other Cisco devices ignore this attribute.

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).

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 Cisco NX-OS device is configured as a TACACS+ client of the AAA servers.

Guidelines and Limitations for TACACS+

TACACS+ has the following guidelines and limitations:

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

Default Settings for TACACS+

This table lists the default settings for TACACS+ parameters.

Table 6: 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

Parameters	Default
Privilege level support for TACACS+ authorization	Disabled

One-Time Password Support

A one-time password (OTP) is a password that is valid for a single login session or a transaction. OTPs avoid multiple disadvantages that are associated with the static passwords. OTPs are not at risk to replay attacks. If an intruder manages to record an OTP that was already used to log into a service or to conduct an operation, it cannot be misused because it is no longer valid.

OTPs are applicable only to the RADIUS and TACACS+ protocol daemons. For a RADIUS protocol daemon, you must ensure that you disable the ASCII authentication mode. For a TACACS+ protocol daemon, you must enable the ASCII authentication mode. To enable the ASCII authentication mode, use the **aaa authentication login ascii-authentication** command.

Configuring TACACS+

This section describes how to configure TACACS+ on a Cisco NX-OS device.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

TACACS+ Server Configuration Process

Procedure

Step 1 Enable TACACS+.

Step 2 Establish the TACACS+ server connections to the Cisco NX-OS device.

Step 3 Configure the secret keys for the TACACS+ servers.

Step 4 If needed, configure TACACS+ server groups with subsets of the TACACS+ servers for AAA authentication methods.

Step 5 (Optional) Configure the TCP port.

Step 6 (Optional) If needed, configure periodic TACACS+ server monitoring.

Step 7 (Optional) If TACACS+ distribution is enabled, commit the TACACS+ configuration to the fabric.

Enabling TACACS+

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

Procedure

	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.

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

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 Cisco NX-OS device. You can configure up to 64 TACACS+ servers.

Note By default, when you configure a TACACS+ server IP address or hostname on the Cisco NX-OS device, the TACACS+ server is added to the default TACACS+ server group. You can also add the TACACS+ server to another TACACS+ server group.

Before you begin

Enable TACACS+.

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

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **Example:** `switch(config)# tacacs-server host 10.10.2.2`	Specifies the IPv4 or IPv6 address or hostname for a TACACS+ server.
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.

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.

Configuring Global TACACS+ Keys

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

Before you begin

Enable TACACS+.

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

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
Step 2	**tacacs-server key** [**0** | **6** | **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**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS 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 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.

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 Cisco NX-OS device and the TACACS+ server host.

Before you begin

Enable TACACS+.

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

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
Step 2	**tacacs-server host** {*ipv4-address* | *ipv6-address* | *host-name*} **key** [**0** | **6** | **7**] *key-value* **Example:** `switch(config)# tacacs-server host 10.10.1.1 key 0 PlIjUhYg`	Specifies a secret key for a specific TACACS+ server. You can specify that the *key-value* is in clear text format (**0**), is type-6 encrypted (**6**), or is type-7 encrypted (**7**). The Cisco NX-OS 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 secret key is used instead of the global secret key.

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+.

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	

Step 2	**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 3	**server** {*ipv4-address* | *ipv6-address* | *hostname*} **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.

Step 4	**exit**
	Example:
	`switch(config-tacacs+)# exit`
	`switch(config)#`
		Exits TACACS+ server group configuration mode.

Step 5	(Optional) **show tacacs-server groups** **Example:** `switch(config)# show tacacs-server groups`	Displays the TACACS+ server group 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 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 Cisco NX-OS software uses any available interface.

Procedure

	Command or Action	Purpose

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

	Command or Action	Purpose
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.

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.

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 Cisco NX-OS 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.

Note If you enable the directed-request option, the Cisco NX-OS device uses only the TACACS+ method for authentication and not the default local method.

Note User-specified logins are supported only for Telnet sessions.

Before you begin

Enable TACACS+.

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
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** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration.

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.

Configuring the Timeout Interval for a TACACS+ Server

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

Before you begin

Enable TACACS+.

Procedure

	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* | *hostname*} **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.

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.

Configuring TCP Ports

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

Before you begin

Enable TACACS+.

Procedure

	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* | *hostname*} **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.
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.

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.

Configuring Global Periodic TACACS+ Server Monitoring

You can monitor the availability of all TACACS+ servers without having to configure the test parameters for each server individually. Any servers for which test parameters are not configured are monitored using the global level parameters.

Note Test parameters that are configured for individual servers take precedence over global test parameters.

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

Note The test parameters are distributed across switches. If even one switch in the fabric is running an older release, the test parameters are not distributed to any switch in the fabric.

Note To protect network security, we recommend that you use a username that is not the same as an existing username in the TACACS+ database.

Note The default idle timer value is 0 minutes. When the idle time interval is 0 minutes, periodic TACACS+ server monitoring is not performed.

Before you begin

Enable TACACS+.

Procedure

	Command or Action	Purpose

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

Step 2	**tacacs-server test** {**idle-time** *minutes* | **password** *password* [**idle-time** *minutes*] | **username** *name* [**password** *password* [**idle-time** *minutes*]]} **Example:** `switch(config)# tacacs-server test username user1 password Ur2Gd2BH idle-time 3`	Specifies parameters for global 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 Cisco NX-OS 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:**	Exits configuration mode.

	Command or Action	Purpose
	`switch(config)# exit` `switch#`	
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.

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 Cisco NX-OS device waits, after declaring a TACACS+ server is dead, before sending out a test packet to determine if the server is now alive.

Note When the dead-timer interval is 0 minutes, TACACS+ servers are not marked as dead even if they are not responding. You can configure the dead-timer per group.

Before you begin

Enable TACACS+.

Procedure

	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.

	Command or Action	Purpose
Step 4	(Optional) **tacacs+ commit** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration.

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.

Configuring ASCII Authentication

You can enable ASCII authentication on the TACACS+ server.

Before you begin

Enable TACACS+.

Procedure

	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.

	Command or Action	Purpose
Step 4	(Optional) **tacacs+ commit** **Example:** `switch(config)# tacacs+ commit`	Applies the TACACS+ configuration changes in the temporary database to the running configuration.

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.

Configuring AAA Authorization on TACACS+ Servers

You can configure the default AAA authorization method for TACACS+ servers.

Before you begin

Enable TACACS+.

Procedure

	Command or Action	Purpose

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

Step 2	**aaa authorization ssh-certificate default** {**group** *group-list* [**none**] | **local** | **none**}
	Example:
	`switch(config)# aaa authorization ssh-certificate`
	`default group TACACSServer1 TACACSServer2`
		Configures the default AAA authorization method for the TACACS+ servers.
		The **ssh-certificate** keyword configures TACACS+ or local authorization with certificate authentication. 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 TACACS+ server group names. Servers belonging to this group are contacted for AAA authorization. The **local**

	Command or Action	Purpose
		method uses the local database for authorization, and the **none** method specifies that no AAA authorization be used.

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

Step 4	(Optional) **show aaa authorization** [**all**] **Example:** `switch# 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# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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.

Note If you use a console to login to the server, command authorization is disabled. Authorization is available for both non-console and console sessions. By default, command authorization is disabled for console sessions even if it is configured for default (non-console) sessions. You must explicitly configure a AAA group for the console to enable command authorization for console sessions.

Note By default, context sensitive help and command tab completion show only the commands supported for a user as defined by the assigned roles. When you enable command authorization, the Cisco NX-OS software displays all commands in the context sensitive help and in tab completion, regardless of the role assigned to the user.

Before you begin

Enable TACACS+.

Procedure

	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**} {**group** *group-list* [**local**] | **local**}
	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**.

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.

	Command or Action	Purpose

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.

Testing Command Authorization on TACACS+ Servers

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

Note You must send correct commands for authorization or else the results may not be reliable.

Note The **test** command uses the default (non-console) method for authorization, not the console method.

Before you begin

Enable TACACS+.

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

Procedure

	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-string* argument if it contains spaces.

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.

Note The commands do no execute when you enable authorization verification.

Procedure

	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 Cisco NX-OS 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.

Permitting or Denying Commands for Users of Privilege Roles

As a network administrator, you can modify the privilege roles to permit users to execute specific commands or to prevent users from running those commands.

You must follow these guidelines when changing the rules of privilege roles:

- You cannot modify the priv-14 and priv-15 roles.
- You can add deny rules only to the priv-0 role.
- These commands are always permitted for the priv-0 role: **configure**, **copy**, **dir**, **enable**, **ping**, **show**, **ssh**, **telnet**, **terminal**, **traceroute**, **end**, and **exit**.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **role name priv-***n*
	Example:
	`switch(config)# role name priv-5`
	`switch(config-role)#`
		Enables or disables a privilege role and enters role configuration mode. The *n* argument specifies the privilege level and is a number between 0 and 13.

Step 3	**rule** *number* {**deny** | **permit**} **command** *command-string*	Configures a command rule for users of privilege roles. These rules permit or deny users

	Command or Action	Purpose
	Example: `switch(config-role)# rule 2 permit command pwd`	to execute specific commands. You can configure up to 256 rules for each role. The 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. The *command-string* argument can contain spaces. **Note** Repeat this command for as many rules as needed.

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

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

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+.

Procedure

	Command or Action	Purpose
Step 1	**test aaa server tacacs+** {*ipv4-address* | *ipv6-address* | *hostname*} [**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.

Disabling TACACS+

You can disable TACACS+.

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

Procedure

	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**
	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.

Monitoring TACACS+ Servers

You can monitor the statistics that the Cisco NX-OS device maintains for TACACS+ server activity.

Before you begin

Configure TACACS+ servers on the Cisco NX-OS device.

Procedure

	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.

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+ Cisco 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.
show privilege	Displays the current privilege level, username, and status of cumulative privilege support.

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) --
```

The following example shows how to enable the cumulative privilege of roles, configure a secret password for privilege level 2, and configure user3 for privilege level 2 authorization:

```
switch# configure terminal
switch(config)# feature privilege
switch(config)# enable secret def456 priv-lvl 2
switch(config)# username user3 priv-lvl 2
```

```
switch(config)# show privilege
User name: user3
Current privilege level: -2
Feature privilege: Enabled
switch(config)# copy running-config startup-config
switch(config)# exit
```

The following example shows how to change user3 from the priv-2 role to the priv-15 role. After entering the **enable 15** command, the user is prompted to enter the password that was configured by the administrator using the **enable secret** command. Privilege level 15 gives this user network-admin privileges under the enable mode.

```
User Access Verification
login: user3
Password: ******
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright ©) 2013, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
switch#
switch# enable 15
Password: def456
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright ©) 2013, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php
switch-enable#
```

The following example shows how to permit all users with roles priv-5 and above to execute the **pwd** command:

```
switch# configure terminal
switch(config)# role name priv-5
switch(config-role)# rule 1 permit command pwd
```

The following example shows how to deny the **show running-config** command to all users with roles below priv-5. First, you must remove the permission to execute this command from the priv-0 role; then you must permit the command at role priv-5 so that users with roles priv-5 and above have permission to run the command.

```
switch# configure terminal
switch(config)# role name priv-0
```

```
switch(config-role)# rule 2 deny command show running-config
switch(config-role)# exit
switch(config)# role name priv-5
switch(config-role)# rule 3 permit command show running-config
switch(config-role)# exit
```

Additional References for TACACS+

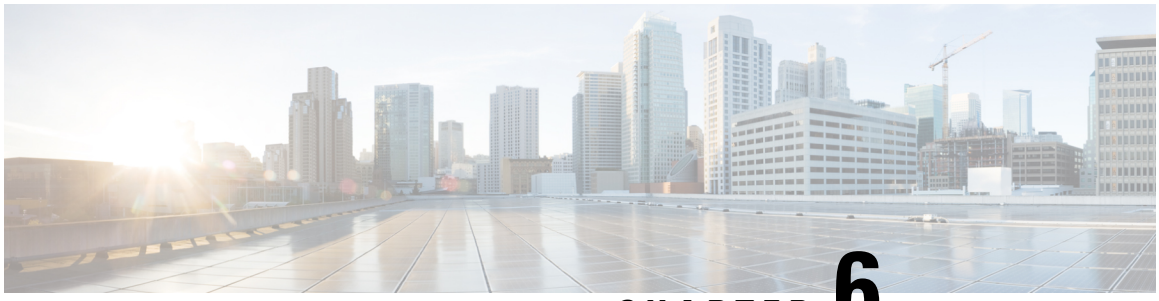
This section includes additional information related to implementing TACACS+.

Related Documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco NX-OS 3400-S NX-OS 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.	—



CHAPTER 6

Configuring LDAP

This chapter describes how to configure the Lightweight Directory Access Protocol (LDAP) on Cisco NX-OS devices and includes the following sections:

- About LDAP, on page 81
- Prerequisites for LDAP, on page 84
- Default Settings for LDAP, on page 84
- Configuring LDAP, on page 85
- Monitoring LDAP Servers, on page 94
- Clearing LDAP Server Statistics, on page 94
- Verifying the LDAP Configuration, on page 95
- Configuration Examples for LDAP, on page 95
- Additional References for LDAP, on page 96

About LDAP

The Lightweight Directory Access Protocol (LDAP) provides centralized validation of users attempting to gain access to a Cisco NX-OS 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 Cisco NX-OS 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 (port 389) for transport requirements. Cisco NX-OS devices provide centralized authentication using the LDAP protocol.

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.

Note As an alternative to the bind method, you can establish LDAP authentication using the compare method, which compares the attribute values of a user entry at the server. For example, the user password attribute can be compared for authentication. The default password attribute type is userPassword.

LDAP Operation for User Login

When a user attempts a Password Authentication Protocol (PAP) login to a Cisco NX-OS device using LDAP, the following actions occur:

1. When the Cisco NX-OS device establishes a connection, it contacts the LDAP daemon to obtain the username and password.
2. The Cisco NX-OS device eventually receives one of the following responses from the LDAP daemon:
   - ACCEPT—User authentication succeeds and service begins. If the Cisco NX-OS 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 Cisco NX-OS device. If the Cisco NX-OS device receives an ERROR response, the Cisco NX-OS 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 Cisco NX-OS device. Users must first successfully complete LDAP authentication before proceeding to LDAP authorization.
3. If LDAP authorization is required, the Cisco NX-OS 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

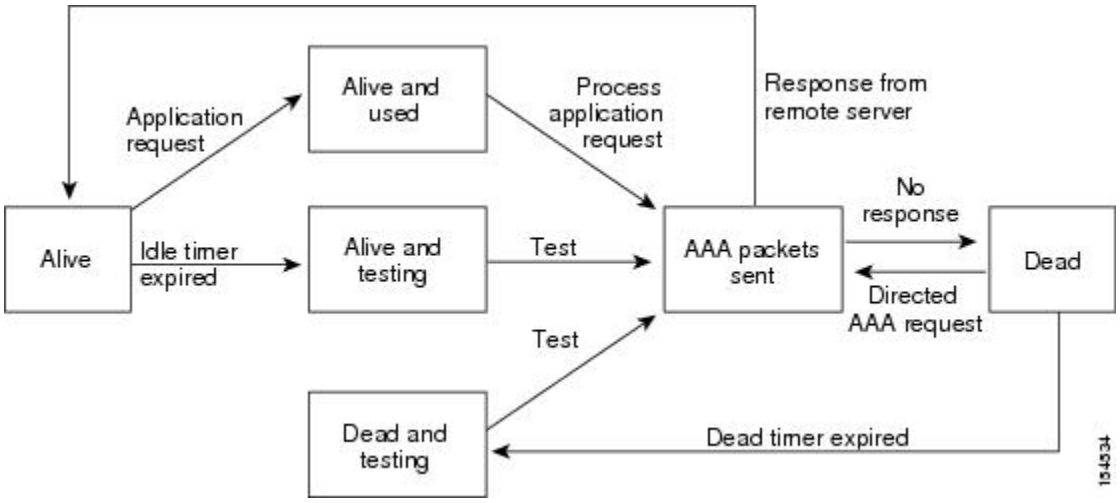
Note LDAP allows an arbitrary conversation between the daemon and the user until the daemon receives enough information to authenticate the user. This action is usually done by prompting for a username and password combination but may include prompts for other items.

Note In LDAP, authorization can occur before authentication.

LDAP Server Monitoring

An unresponsive LDAP server can delay the processing of AAA requests. A Cisco NX-OS device can periodically monitor an LDAP server to check whether it is responding (or alive) to save time in processing AAA requests. The Cisco NX-OS device marks unresponsive LDAP servers as dead and does not send AAA requests to any dead LDAP servers. A Cisco NX-OS 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 Cisco NX-OS device displays an error message that a failure is taking place before it can impact performance. The following figure shows the server states for LDAP server monitoring.

Figure 3: LDAP Server States



Note The monitoring interval for alive servers and dead servers is different and can be configured by the user. The LDAP server monitoring is performed by sending a test authentication request to the LDAP server.

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.

Cisco VSA Format for LDAP

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

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

The protocol is a Cisco attribute for a particular type of authorization, the separator is an = (equal sign) for mandatory attributes, and an * (asterisk) indicates optional attributes. When you use LDAP servers for authentication on a Cisco NX-OS 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 option is supported by the Cisco NX-OS software:

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

The Cisco NX-OS software supports the following attribute:

- 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.

Virtualization Support for LDAP

The Cisco NX-OS device uses virtual routing and forwarding instances (VRFs) to access the LDAP servers. For more information on VRFs, see the *Cisco Nexus 3400-S NX-OS Unicast Routing Configuration Guide.*

Prerequisites for LDAP

LDAP has the following prerequisites:

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

Default Settings for LDAP

This table lists the default settings for LDAP parameters.

Parameters	Default
LDAP	Disabled
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

Parameters	Default
Periodic server monitoring password	Cisco

Configuring LDAP

This section describes how to configure LDAP on a Cisco NX-OS device.

LDAP Server Configuration Process

You can configure LDAP servers by following this configuration process.

1. Enable LDAP.
2. Establish the LDAP server connections to the Cisco NX-OS device.
3. If needed, configure LDAP server groups with subsets of the LDAP servers for AAA authentication methods.
4. (Optional) Configure the TCP port.
5. (Optional) Configure the default AAA authorization method for the LDAP server.
6. (Optional) Configure an LDAP search map.
7. (Optional) If needed, configure periodic LDAP server monitoring.

Enabling or Disabling LDAP

By default, the LDAP feature is disabled on the Cisco NX-OS device. You must explicitly enable the LDAP feature to access the configuration and verification commands for authentication.

Procedure

	Command or Action	Purpose

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

Step 2	Required: [**no**] **feature ldap** **Example:** `switch(config)# feature ldap`	Enables LDAP. Use the **no** form of this command to disable LDAP. **Note** When you disable LDAP, all related configurations are automatically discarded.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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 Cisco NX-OS device. You can configure up to 64 LDAP servers.

Note By default, when you configure an LDAP server IP address or hostname on the Cisco NX-OS device, the LDAP server is added to the default LDAP server group. You can also add the LDAP server to another LDAP server group.

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 Cisco NX-OS device.

Procedure

	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**] **Example:** `switch(config)# ldap-server host 10.10.2.2 enable-ssl`	Specifies the IPv4 or IPv6 address or hostname for an LDAP server. The **enable-ssl** keyword ensures the integrity and confidentiality of the transferred data by causing the LDAP client to establish an SSL session prior to sending the bind or search request.
Step 3	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server 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.

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.

Procedure

	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* | *hostname*} **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`	Specifies the rootDN for the LDAP server database 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	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server 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.

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.

Procedure

	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.

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

	Command or Action	Purpose
Step 9	(Optional) **show ldap-server groups** **Example:** `switch(config)# show ldap-server groups`	Displays the LDAP server group configuration.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring the Global LDAP Timeout Interval

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

Before you begin

Enable LDAP.

Procedure

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

Configuring TCP Ports

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

Before you begin

Enable LDAP.

Procedure

	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* | *hostname*} **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	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server 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.

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.

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	

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 Nexus role definition.

Step 4	(Optional) **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.

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 Cisco NX-OS device sends out a test packet. You can configure this option to test servers periodically, or you can run a one-time only test.

Note To protect network security, we recommend that you use a username that is not the same as an existing username in the LDAP database.

Before you begin

Enable LDAP.

Procedure

	Command or Action	Purpose

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

Step 2	Required: [**no**] **ldap-server host** {*ipv4-address* | *ipv6-address* | *hostname*} **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 Cisco. 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 Cisco NX-OS 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	(Optional) **show ldap-server** **Example:** `switch(config)# show ldap-server`	Displays the LDAP server 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.

Configuring the Global LDAP Timeout Interval

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

Before you begin

Enable LDAP.

Procedure

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

Configuring AAA Authorization on LDAP Servers

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

Before you begin

Enable LDAP.

Procedure

	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.

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

Monitoring LDAP Servers

You can monitor the statistics that the Cisco NX-OS device maintains for LDAP server activity.

Before you begin

Configure LDAP servers on the Cisco NX-OS device.

Procedure

	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 server statistics.

Clearing LDAP Server Statistics

You can display the statistics that the Cisco NX-OS device maintains for LDAP server activity.

Before you begin

Configure LDAP servers on the Cisco NX-OS device.

Procedure

	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.
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.

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 {*hostname* | *ipv4-address* | *ipv6-address*}	Displays LDAP statistics.
show ldap-search-map	Displays information about the configured LDAP attribute maps.

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 att-name 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
```

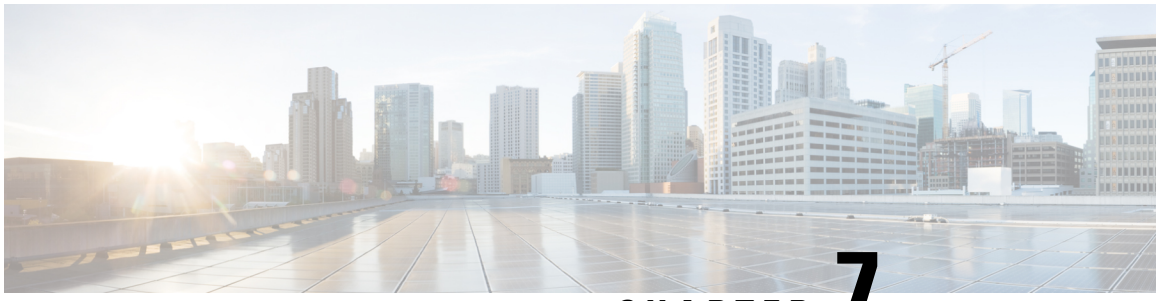
Additional References for LDAP

Related Documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus 3400-S NX-OS 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.	—



CHAPTER 7

Configuring SSH and Telnet

This chapter describes how to configure Secure Shell Protocol (SSH) and Telnet on Cisco NX-OS devices.

This chapter includes the following sections:

- About SSH and Telnet, on page 97
- Prerequisites for SSH and Telnet, on page 99
- Guidelines and Limitations for SSH and Telnet, on page 99
- Default Settings for SSH and Telnet, on page 100
- Configuring SSH , on page 100
- Configuring Telnet, on page 115
- Verifying the SSH and Telnet Configuration, on page 117
- Configuration Example for SSH, on page 117
- Configuration Example for SSH Passwordless File Copy, on page 118
- Configuration Example for X.509v3 Certificate-Based SSH Authentication, on page 120
- Additional References for SSH and Telnet, on page 121

About SSH and Telnet

This section includes information about SSH and Telnet.

SSH Server

You can use the SSH server to enable an SSH client to make a secure, encrypted connection to a Cisco NX-OS device. SSH uses strong encryption for authentication. The SSH server in the Cisco NX-OS 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.

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 Cisco NX-OS device to make a secure, encrypted connection to another Cisco NX-OS 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 Cisco NX-OS software works with publicly and commercially available SSH servers.

SSH Server Keys

SSH requires server keys for secure communications to the Cisco NX-OS 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)
- SSH version 2 using the Elliptic Curve Digital Signature Algorithm (ECDSA)

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.
- The **ecdsa** option generates the ECDSA key-pair for the SSH version 2 protocol.

By default, the Cisco NX-OS 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.

SSH Authentication Using Digital Certificates

SSH authentication on Cisco NX-OS 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 SSH authentication using an X.509 certificate. If the authentication fails, you are prompted for a password.

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 Cisco device access. The SSH client is provided by Cisco partner Pragma Systems.

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 Cisco NX-OS device.

Prerequisites for SSH and Telnet

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

Guidelines and Limitations for SSH and Telnet

SSH and Telnet have the following configuration guidelines and limitations:

- The Cisco NX-OS software supports only SSH version 2 (SSHv2).
- Due to a Poodle vulnerability, SSLv3 is no longer supported.
- You can configure your device for SSH authentication using an X.509 certificate. If the authentication fails, you are prompted for a password.
- The SFTP server feature does not support the regular SFTP **chown** and **chgrp** commands.
- When the SFTP server is enabled, only the admin user can use SFTP to access the device.
- 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 Nexus device is reloaded unless a local user account with the same name as the remote user account is configured on the device before the SSH keys are imported.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings for SSH and Telnet

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

Table 7: 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
Maximum number of SSH login attempts	3
SCP server	Disabled
SFTP server	Disabled

Configuring SSH

This section describes how to configure SSH.

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.

Procedure

	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**]] | **ecdsa** [*bits* [**force**]]}	Generates the SSH server key.

	Command or Action	Purpose
	Example: `switch(config)# ssh key rsa 2048`	The *bits* argument is the number of bits used to generate the RSA key. The range is from 768 to 2048. 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. **Note** If you configure ssh key dsa, you must do the following additional configurations: ssh keytypes all and ssh kexalgos all
Step 4	**ssh rekey max-data** *max-data* **max-time** *max-time*i **Example:** `switch(config)# ssh rekey max-data 1K max-time 1M`	Configures the rekey parameters.
Step 5	**feature ssh** **Example:** `switch(config)# feature ssh`	Enables SSH.

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

Step 7	(Optional) **show ssh key** [**dsa** | **rsa** | **ecdsa**] [] **Example:** `switch# show ssh key`	Displays the SSH server keys. This command displays the fingerprint in SHA256 format by default. SHA256 is more secure than the old default format of MD5. However, the **md5** option has been added, if you want to see the fingerprint in MD5 format for backward compatibility.
Step 8	**show run security all**	
Step 9	(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 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

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.

Procedure

	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.

Procedure

	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`
	`AAAAB3NzaC1yc2EAAAABIwAAAIEAy19bF6QsZl9G+3flXswK3OiWH7yUyuA50rv7gsEPj`
	`hOBmsi6PAVKuilnIf/IQum+lNqP/eLowb7ib0+lVRXFY/G+LNIQN3g9igG3Oc6k6+`
	`XVn+NjnIlB7ihvpVh7dLddMCKwOnXHYshXmSiH3UD/vKyziEh5S4IplxB=`
		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.

Configuring a Maximum Number of SSH Login Attempts

You can configure the maximum number of SSH login attempts. If the user exceeds the maximum number of permitted attempts, the session disconnects.

Note The total number of login attempts includes attempts through public-key authentication, certificate-based authentication, and password-based authentication. If public-key authentication is enabled, it takes priority. If only certificate-based and password-based authentication are enabled, certificate-based authentication takes priority. If you exceed the configured number of login attempts through all of these methods, a message appears indicating that too many authentication failures have occurred.

Procedure

	Command or Action	Purpose

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

Step 2	**ssh login-attempts** *number* **Example:** `switch(config)# ssh login-attempts 5`	Configures the maximum number of times that a user can attempt to log into an SSH session. The default maximum number of login attempts is 3. The range is from 1 to 10. **Note** The **no** form of this command removes the previous login attempts value and sets the maximum number of login attempts to the default value of 3.
Step 3	(Optional) **show running-config security all** **Example:** `switch(config)# show running-config security all`	Displays the configured maximum number of SSH login attempts.
Step 4	(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 Cisco NX-OS 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.

Procedure

	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.

Starting SSH Sessions from Boot Mode

You can start SSH sessions from the boot mode of the Cisco NX-OS device to connect to remote devices.

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.

Procedure

	Command or Action	Purpose
Step 1	**ssh** [*username@*]*hostname* **Example:** `switch(boot)# ssh user1@10.10.1.1`	Creates an SSH session to a remote device from the boot mode of the Cisco NX-OS device. The default VRF is always used.
Step 2	**exit** **Example:** `switch(boot)# exit`	Exits boot mode.
Step 3	**copy scp:**//[*username@*]*hostname*/*filepath directory* **Example:** `switch# copy scp://user1@10.10.1.1/users abc`	Copies a file from the Cisco NX-OS device to a remote device using the Secure Copy Protocol (SCP). The default VRF is always used.

Configuration Example for SSH Passwordless File Copy

The following example shows how to copy files from a Cisco NX-OS device to a secure copy (SCP) or secure FTP (SFTP) server without a password:

Procedure

Step 1 Generate the SSH public and private keys and store them in the home directory of the Cisco NX-OS device for the specified user.

Example:

```
switch# configure terminal
switch(config)# username admin keypair generate rsa
generating rsa key(1024 bits)......
generated rsa key
```

Step 2 Display the public key for the specified user.

Example:

```
switch(config)# show username admin keypair

**************************************

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZElTfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=

bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
```

Step 3 Export the public and private keys from the home directory of the Cisco NX-OS device to the specified bootflash directory.

Example:

```
switch(config)# username admin keypair export bootflash:key_rsa rsa
Enter Passphrase:
switch(config)# dir
.
.
.
        951     Jul 09 11:13:59 2013  key_rsa
        221     Jul 09 11:14:00 2013  key_rsa.pub
.
.
```

Step 4 After copying these two files to another Cisco NX-OS device using the **copy scp** or **copy sftp** command, import them to the home directory of the Cisco NX-OS device.

Example:

```
switch(config)# username admin keypair import bootflash:key_rsa rsa
Enter Passphrase:
```

```
switch(config)# show username admin keypair
**************************************

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZEl1TfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=

bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
switch(config)#
```

Step 5 On the SCP or SFTP server, append the public key stored in key_rsa.pub to the authorized_keys file.

Example:

```
$ cat key_rsa.pub >> $HOME/.ssh/ authorized_keys
```

You can now copy files from the Cisco NX-OS device to the server without a password using standard SSH and SCP commands.

Step 6 (Optional) Repeat this procedure for the DSA keys.

Configuring SCP and SFTP Servers

You can configure an SCP or SFTP server on the Cisco NX-OS device in order to copy files to and from a remote device. After you enable the SCP or SFTP server, you can execute an SCP or SFTP command on the remote device to copy the files to or from the Cisco NX-OS device.

Note The arcfour and blowfish cipher options are not supported for the SCP server.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **feature scp-server** **Example:**	Enables or disables the SCP server on the Cisco NX-OS device.

	Command or Action	Purpose
	`switch(config)# feature scp-server`	
Step 3	Required: [**no**] **feature sftp-server** **Example:** `switch(config)# feature sftp-server`	Enables or disables the SFTP server on the Cisco NX-OS device.

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

Step 5	(Optional) **show running-config security** **Example:** `switch# show running-config security`	Displays the configuration status of the SCP and SFTP servers.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring X.509v3 Certificate-Based SSH Authentication

You can configure SSH authentication using X.509v3 certificates.

Before you begin

Enable the SSH server on the remote device.

Procedure

	Command or Action	Purpose

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

Step 2	**username** *user-id* [**password** [**0** | **5**] *password*] **Example:** `switch(config)# username jsmith 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, hyphen (-), period (.), underscore (_), plus sign (+), and equal sign (=). The at symbol (@) is supported in remote usernames but not in local usernames.

	Command or Action	Purpose
		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** If you do not specify a password, the user might not be able to log in to the Cisco NX-OS device. **Note** If you create a user account with the encrypted password option, the corresponding SNMP user will not be created.
Step 3	**username** *user-id* **ssh-cert-dn** *dn-name* {**dsa** | **rsa**} **Example:** `switch(config)# username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1, emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith" rsa`	Specifies an SSH X.509 certificate distinguished name and DSA or RSA algorithm to use for authentication for an existing user account. The distinguished name can be up to 512 characters and must follow the format shown in the examples. Make sure the email address and state are configured as emailAddress and ST, respectively.

Step 4	[**no**] **crypto ca trustpoint** *trustpoint*
	Example:
	`switch(config)# crypto ca trustpoint winca`
	`switch(config-trustpoint)#`
		Configures a 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	**crypto ca authenticate** *trustpoint* **Example:** `switch(config-trustpoint)# crypto ca authenticate winca`	Configures a CA certificate for the trustpoint. **Note** To delete a CA certificate, enter the **delete ca-certificate** command in the trustpoint configuration mode.
Step 6	(Optional) **crypto ca crl request** *trustpoint* **bootflash:***static-crl***.crl** **Example:** `switch(config-trustpoint)# crypto ca crl request winca bootflash:crllist.crl`	This command is optional but highly recommended. Configures the certificate revocation list (CRL) for the trustpoint. 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 method.

	Command or Action	Purpose
		Note To delete the CRL, enter the **delete crl** command.
Step 7	(Optional) **show crypto ca certificates** **Example:** `switch(config-trustpoint)# show crypto ca certificates`	Displays the configured certificate chain and associated trustpoint.
Step 8	(Optional) **show crypto ca crl** *trustpoint* **Example:** `switch(config-trustpoint)# show crypto ca crl winca`	Displays the contents of the CRL list of the specified trustpoint.
Step 9	(Optional) **show user-account** **Example:** `switch(config-trustpoint)# show user-account`	Displays configured user account details.
Step 10	(Optional) **show users** **Example:** `switch(config-trustpoint)# show users`	Displays the users logged into the device.
Step 11	(Optional) **copy running-config startup-config** **Example:** `switch(config-trustpoint)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Legacy SSH Algorithm Support

You can configure support for legacy SSH security algorithms, message authentication codes (MACs), key types, and ciphers.

Procedure

	Command or Action	Purpose

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

Step 2	(Optional) **ssh kexalgos all** **Example:** `switch(config)# ssh kexalgos all`	Enables all supported KexAlgorithms which are the key exchange methods that are used to generate per-connection keys. Supported KexAlgorithmns are:

	Command or Action	Purpose
		• curve25519-sha256 • diffie-hellman-group-exchange-sha256 • diffie-hellman-group14-sha1 • diffie-hellman-group1-sha1 • ecdh-sha2-nistp256 • ecdh-sha2-nistp384 • ecdh-sha2-nistp521
Step 3	(Optional) **ssh macs all** **Example:** `switch(config)# ssh macs all`	Enables all supported MACs which are the message authentication codes used to detect traffic modification. Supported MACs are: • hmac-sha1 • hmac-sha2-256 • hmac-sha2-512
Step 4	(Optional) **ssh ciphers all** **Example:** `switch(config)# ssh ciphers all`	Enables all supported ciphers to encrypt the connection. Supported ciphers are: • aes128-cbc • aes192-cbc • aes256-cbc • aes128-ctr • aes192-ctr • aes256-ctr • aes256-gcm@openssh.com • aes128-gcm@openssh.com
Step 5	(Optional) **ssh keytypes all** **Example:** `switch(config)# ssh keytypes all`	Enables all supported PubkeyAcceptedKeyTypes which are the public key algorithms that the server can use to authenticate itself to the client. Supported key types are: • ecdsa-sha2-nistp256 • ecdsa-sha2-nistp384

	Command or Action	Purpose
		• ecdsa-sha2-nistp521 • ssh-dss • ssh-rsa

Changing the Default SSH Server Port

You can change the SSHv2 port number from the default port number 22. Encryptions used while changing the default SSH port provides you with connections that support stronger privacy and session integrity

Procedure

	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	**show sockets** *local-port-range*
	Example:
	`switch(config)# show sockets local port range (15001 - 58000)`
	`switch(config)# local port range (58001 - 63535) and nat port range (63536 - 65535)`
		Displays the available port range.

Step 4	**ssh port** *local-port* **Example:** `switch(config)# ssh port 58003`	Configures the port.
Step 5	**feature ssh** **Example:** `switch(config)# feature ssh`	Enables SSH.

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

Step 7	(Optional) **show running-config security all** **Example:**	Displays the security configuration.

	Command or Action	Purpose
	`switch# ssh port 58003`	
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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.

Procedure

	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 Cisco NX-OS device. You can disable the SSH server to prevent SSH access to the switch.

Procedure

	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:**	Displays the SSH server configuration.

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

Deleting SSH Server Keys

You can delete SSH server keys on the Cisco NX-OS device after you disable the SSH server.

Note To reenable SSH, you must first generate an SSH server key.

Procedure

	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** | **ecdsa**] **Example:** `switch(config)# no ssh key rsa`	Deletes the SSH server key. The default is to delete all the SSH keys.

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.

Clearing SSH Sessions

You can clear SSH sessions from the Cisco NX-OS device.

Procedure

	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.

Configuring Telnet

This section describes how to configure Telnet on the Cisco NX-OS device.

Enabling the Telnet Server

You can enable the Telnet server on the Cisco NX-OS device. By default, the Telnet server is disabled.

Procedure

	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.

	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.

Starting Telnet Sessions to Remote Devices

You can start Telnet sessions to connect to remote devices from the Cisco NX-OS 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 Cisco NX-OS device.

Enable the Telnet server on the remote device.

Procedure

	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*] **Example:** `switch# telnet6 2001:0DB8::ABCD:1 vrf management`	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 VRF.

Clearing Telnet Sessions

You can clear Telnet sessions from the Cisco NX-OS device.

Before you begin

Enable the Telnet server on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	Displays user session information.

	Command or Action	Purpose
Step 2	**clear line** *vty-line* **Example:** `switch(config)# clear line pts/12`	Clears a user Telnet session.

Verifying the SSH and Telnet Configuration

To display the SSH and Telnet configuration information, perform one of the following tasks:

Command	Purpose
show ssh key [**dsa** | **rsa**] [**md5**]	Displays the SSH server keys.
show running-config security [**all**]	Displays the SSH and user account configuration in the running configuration. The **all** keyword displays the default values for the SSH and user accounts.
show ssh server	Displays the SSH server configuration.
show telnet server	Displays the Telnet server configuration.
show username *username* **keypair**	Displays the public key for the specified user.
show user-account	Displays configured user account details.
show users	Displays the users logged into the device.
show crypto ca certificates	Displays the configured CA certificate and associated trustpoint for X.509v3 certificate-based SSH authentication.
show crypto ca crl *trustpoint*	Displays the contents of the CRL list of the specified trustpoint for X.509v3 certificate-based SSH authentication.

Configuration Example for SSH

The following example shows how to configure SSH with an OpenSSH key:

Procedure

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:

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
```

Configuration Example for SSH Passwordless File Copy

The following example shows how to copy files from a Cisco NX-OS device to a secure copy (SCP) or secure FTP (SFTP) server without a password:

Procedure

Step 1 Generate the SSH public and private keys and store them in the home directory of the Cisco NX-OS device for the specified user.

Example:

```
switch# configure terminal
switch(config)# username admin keypair generate rsa
generating rsa key(1024 bits)......
generated rsa key
```

Step 2 Display the public key for the specified user.

Example:

```
switch(config)# show username admin keypair

**************************************

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZElTfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=

bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
```

Step 3 Export the public and private keys from the home directory of the Cisco NX-OS device to the specified bootflash directory.

Example:

```
switch(config)# username admin keypair export bootflash:key_rsa rsa
Enter Passphrase:
switch(config)# dir
.
.
.
        951     Jul 09 11:13:59 2013  key_rsa
        221     Jul 09 11:14:00 2013  key_rsa.pub
.
.
```

Step 4 After copying these two files to another Cisco NX-OS device using the **copy scp** or **copy sftp** command, import them to the home directory of the Cisco NX-OS device.

Example:

```
switch(config)# username admin keypair import bootflash:key_rsa rsa
Enter Passphrase:
switch(config)# show username admin keypair
**************************************

rsa Keys generated: Thu Jul  9 11:10:29 2013

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEAxWmjJT+oQhIcvnrMbx2BmD0P8boZElTfJ
Fx9fexWp6rOiztlwODtehnjadWc6A+DE2DvYNvqsrU9TBypYDPQkR/+Y6cKubyFW
VxSBG/NHztQc3+QC1zdkIxGNJbEHyFoajzNEO8LLOVFIMCZ2Td7gxUGRZc+fbq
S33GZsCAX6v0=

bitcount:262144
fingerprint:
8d:44:ee:6c:ca:0b:44:95:36:d0:7d:f2:b5:78:74:7d
**************************************

could not retrieve dsa key information
**************************************
```

```
switch(config)#
```

Step 5 On the SCP or SFTP server, append the public key stored in key_rsa.pub to the authorized_keys file.

Example:

```
$ cat key_rsa.pub >> $HOME/.ssh/ authorized_keys
```

You can now copy files from the Cisco NX-OS device to the server without a password using standard SSH and SCP commands.

Step 6 (Optional) Repeat this procedure for the DSA keys.

Configuration Example for X.509v3 Certificate-Based SSH Authentication

The following example shows how to configure SSH authentication using X.509v3 certificates:

```
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

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

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

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 = cisco , OU = csg, CN =
user1; Algo: x509v3-sign-rsa
```

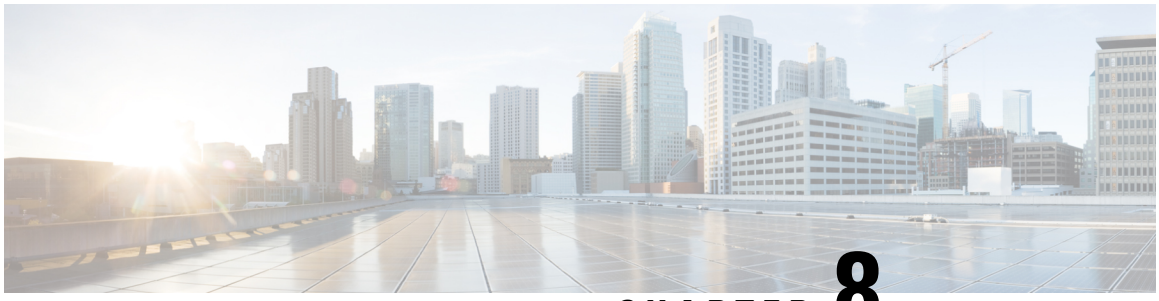
```
show users
NAME       LINE         TIME          IDLE      PID         COMMENT
user1      pts/1        Jul 27 18:43  00:03     18796      (10.10.10.1)   session=ssh
```

Additional References for SSH and Telnet

This section describes additional information related to implementing SSH and Telnet.

Related Documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus 3400-S NX-OS Unicast Routing Configuration Guide*



CHAPTER 8

Configuring User Accounts and RBAC

This chapter describes how to configure user accounts and role-based access control (RBAC) on Cisco NX-OS devices.

This chapter includes the following sections:

- About User Accounts and RBAC, on page 123
- Guidelines and Limitations for User Accounts and RBAC, on page 126
- Default Settings for User Accounts and RBAC, on page 126
- Enabling Password-Strength Checking, on page 127
- Configuring User Accounts, on page 127
- Configuring Roles, on page 129
- About No Service Password-Recovery, on page 137
- Enabling No Service Password-Recovery, on page 137
- Verifying User Accounts and RBAC Configuration, on page 138
- Configuration Examples for User Accounts and RBAC, on page 139
- Additional References for User Accounts and RBAC, on page 141

About User Accounts and RBAC

You can create and manage users accounts and assign roles that limit access to operations on the Cisco NX-OS device. RBAC allows you to define the rules for an assign role that restrict the authorization that the user has to access management operations.

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.

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.

Note User passwords are not displayed in the configuration files.

Caution Usernames must begin with an alphanumeric character and 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.

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:

- If2CoM18
- 2004AsdfLkj30
- Cb1955S21

Note Clear text passwords cannot contain dollar signs ($) or spaces anywhere in the password. Also, they cannot include these special characters at the beginning of the password: quotation marks (" or '), vertical bars (|), or right angle brackets (>).

Note All printable ASCII characters are supported in the password string if they are enclosed in quotation marks.

If a password is trivial (such as a short, easy-to-decipher password), the Cisco NX-OS 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.

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 virtual routing and forwarding instances (VRFs), VLANs, and interfaces.

The Cisco NX-OS software provides the following user roles:

- network-admin—Complete read-and-write access to the entire Cisco NX-OS device
- network-operator or vdc-operator—Complete read access to the entire Cisco NX-OS device

Note You cannot change the user roles.

Note Some **show** commands may be hidden from network-operator users. In addition, some non-**show** commands (such as **telnet**) may be available for this user role.

By default, the user accounts without an administrator role can access only the **show**, **exit**, **end**, and **configure terminal** commands. You can add rules to allow users to configure features.

Note If you belong to multiple roles, you can execute a combination of all the commands permitted by these roles. Access to a command takes priority over being denied access to a command. For example, suppose a user has RoleA, which denied access to the configuration commands. However, the user also has RoleB, which has access to the configuration commands. In this case, the user has access to the configuration commands.

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 Cisco NX-OS 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.

Guidelines and Limitations for User Accounts and RBAC

User accounts and RBAC have the following configuration guidelines and limitations:

- You can add up to 256 rules to a user role.
- You can add up to 64 user-defined feature groups in addition to the default feature group, L3.
- You can configure up to 256 users.
- You can assign a maximum of 64 user roles to a user account.
- If you have a user account configured on the local Cisco NX-OS device that has the same name as a remote user account on an AAA server, the Cisco NX-OS 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 role cannot run the **show running-config** and **show startup-config** commands.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings for User Accounts and RBAC

This table lists the default settings for user accounts and RBAC parameters.

Table 8: Default User Accounts and RBAC Parameters

Parameters	Default
User account password	Undefined
User account expiry date	None
User account role	Network-operator if the creating user has the network-admin role
Default user role	Network-operator
Interface policy	All interfaces are accessible
VLAN policy	All VLANs are accessible
VRF policy	All VRFs are accessible
Feature group	L3

Enabling Password-Strength Checking

You can enable password-strength checking which prevents you from creating weak passwords for user accounts.

Note When you enable password-strength checking, the Cisco NX-OS software does not check the strength of existing passwords.

Procedure

	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.

Configuring User Accounts

You can create a maximum of 256 user accounts on a Cisco NX-OS 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 Cisco NX-OS password encrypts clear text passwords before saving them to the running configuration. Encrypted format passwords are saved to the running configuration without further encryption.

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.

Note Changes to user account attributes do not take effect until the user logs in and creates a new session.

Procedure

	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, 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** If you do not specify a password, the user might not be able to log in to the Cisco NX-OS device.

	Command or Action	Purpose
		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	**username** *user-id* **ssh-cert-dn** *dn-name* {**dsa** | **rsa**}
	
	Example:
	`switch(config)# username NewUser ssh-cert-dn "/CN = NewUser, OU = Cisco Demo, O = Cisco, C = US" rsa`
	
	Example:
	`switch(config)# username jsmith ssh-cert-dn "/O = ABCcompany, OU = ABC1, emailAddress = jsmith@ABCcompany.com, L = Metropolis, ST = New York, C = US, CN = jsmith" rsa`
		Specifies an SSH X.509 certificate distinguished name and DSA or RSA algorithm to use for authentication for an existing user account. The distinguished name can be up to 512 characters and must follow the format shown in the examples. Make sure the email address and state are configured as emailAddress and ST, respectively.

Step 5	**exit** **Example:** `switch(config)# exit switch#`	Exits global configuration mode.
Step 6	(Optional) **show user-account** **Example:** `switch# show user-account`	Displays the role configuration.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring Roles

This section describes how to configure user roles.

Creating User Roles and Rules

You can configure up to 64 user roles. 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.

Note Regardless of the read-write rule configured for a user role, some commands can be executed only through the predefined network-admin role.

Before you begin

If you want to distribute the user role configuration, enable user role configuration distribution on all Cisco NX-OS devices to which you want the configuration distributed.

Procedure

	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.
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.

	Command or Action	Purpose
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, MAC address tables, specific MIBs, and so on. **Note** The deepest OID can be at the scalar level or at the table root level. 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.

	Command or Action	Purpose
Step 13	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Creating Feature Groups

You can create custom feature groups to add to the default list of features provided by the Cisco NX-OS software. These groups contain one or more of the features. You can create up to 64 feature groups.

Note You cannot change the default feature group L3.

Before you begin

If you want to distribute the user role configuration, enable user role configuration distribution on all Cisco NX-OS devices to which you want the configuration distributed.

Procedure

	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 radius`	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.

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

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.

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 Cisco NX-OS devices to which you want the configuration distributed.

Procedure

	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:**	Enters role interface policy configuration mode.

	Command or Action	Purpose
	`switch(config-role)# interface policy deny` `switch(config-role-interface)#`	
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**} **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.
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.

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.

Before you begin

Create one or more user roles.

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	

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	**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.
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.

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.

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 Cisco NX-OS devices to which you want the configuration distributed.

Procedure

	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.
Step 9	(Optional) **copy running-config startup-config** **Example:**	Copies the running configuration to the startup configuration.

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

About No Service Password-Recovery

The No Service Password-Recovery feature enables anyone with console access, the ability to access the router and its network. The No Service Password-Recovery feature prevents the password recovery with standard procedure as described in the Cisco Nexus 3400-S NX-OS Troubleshooting Gude..

Enabling No Service Password-Recovery

If the no service password-recovery feature is enabled, then none except the administrator with network privileges will be able to modify the administrator password.

Before you begin

If you plan to enter the no service password-recovery command, Cisco recommends that you save a copy of the system configuration file in a location away from the device.

Procedure

	Command or Action	Purpose

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

Step 2	**no service password-recovery**
	Example:
	`switch(config)# no service password-recovery`
	`WARNING: Executing this command will disable the password recovery mechanism. Do not execute this command without another plan for password recovery. Are you sure you want to continue? (y/n) : [y] y`
	`switch(config)# copy run start`
	`[########################################] 100%`
	`Copy complete, now saving to disk (please wait)...`
	`Copy complete.`
		Disables the password recovery mechanism.

Step 3	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

	Command or Action	Purpose

Step 4	**Reload**
	Example:
	`switch(config)# Reload`
	`This command will reboot the system. (y/n)?  [n] y`
	`2018 Jun 26 16:23:19 BAR %$ VDC-1 %$ %PLATFORM-2-PFM_SYSTEM_RESET: Manual system restart from Command Line Interface`
	
	`CISCO SWITCH Ver 8.34`
	
	`CISCO SWITCH Ver 8.34`
	`Manual system restart from Command Line Interface`
	`writing reset reason 9,`
	`..`
	`..`
	
	`switch(boot)# config t`
	`Enter configuration commands, one per line.  End with CNTL/Z.`
	`switch(boot)(config)# admin-password Abcd!123$`
	`ERROR: service password-recovery disabled. Cannot change password!`
	`switch(boot)(config)#`	

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

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

Verifying User Accounts and RBAC Configuration

To display user account and RBAC configuration information, perform one of the following tasks:

Command	Purpose
show cli syntax roles network-admin	Displays the syntax of the commands that the network-admin role can use.
show cli syntax roles network-operator	Displays the syntax of the commands that the network-operator role can use.
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.

Configuration Examples for User Accounts and RBAC

The following example shows how to configure a user role:

```
role name User-role-A
  rule 2 permit read-write feature bgp
  rule 1 deny command clear *
```

The following example shows how to create a user role that can configure an interface to enable and show BGP and show EIGRP:

```
role name iftest
   rule 1 permit command config t; interface *; bgp *
   rule 2 permit read-write feature bgp
   rule 3 permit read feature eigrp
```

In the above example, rule 1 allows you to configure BGP on an interface, rule 2 allows you to configure the **config bgp** command and enable the exec-level **show** and **debug** commands for BGP, and rule 3 allows you to enable the exec-level **show** and **debug eigrp** 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 aaa
  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
```

Additional References for User Accounts and RBAC

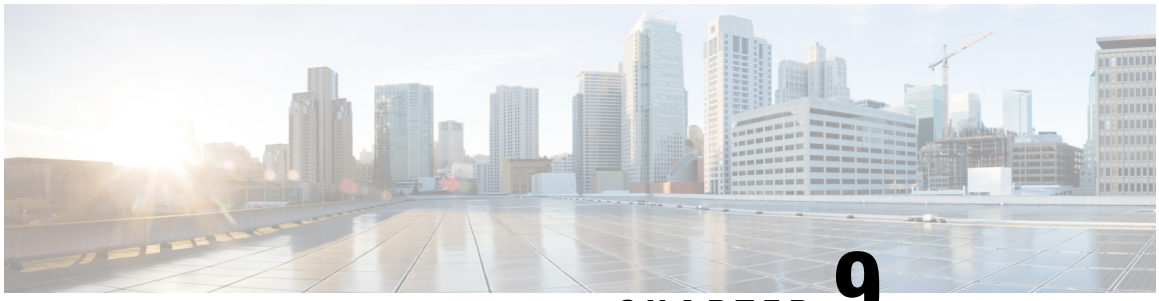
This section includes additional information related to implementing user accounts and RBAC.

Related Documents

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
VRF configuration	*Cisco Nexus 3400-S NX-OS 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.	—



CHAPTER 9

Configuring IP ACLs

This chapter describes how to configure IP access control lists (ACLs) on Cisco NX-OS devices.

Unless otherwise specified, the term IP ACL refers to IPv4 and IPv6 ACLs.

This chapter includes the following sections:

- About ACLs, on page 143
- Prerequisites for IP ACLs, on page 154
- Guidelines and Limitations for IP ACLs, on page 154
- Default Settings for IP ACLs, on page 157
- Configuring IP ACLs, on page 157
- Verifying the IP ACL Configuration, on page 171
- Monitoring and Clearing IP ACL Statistics, on page 172
- Configuration Examples for IP ACLs, on page 172
- Configuring Object Groups, on page 173
- Verifying the Object-Group Configuration, on page 177
- Configuring Time-Ranges, on page 178
- Verifying the Time-Range Configuration, on page 182

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.

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.

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
VTY ACL
Filters virtual teletype (VTY) traffic

This table summarizes the applications for security ACLs.

Table 9: 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 • IPv4 ACLs with UDF-based match • IPv6 ACLs • IPv6 ACLs with UDF-based match • MAC ACLs
Router ACL	• VLAN interfaces • Physical Layer 3 interfaces • Layer 3 Ethernet subinterfaces • Layer 3 Ethernet port-channel interfaces • Management interfaces **Note** You must enable VLAN interfaces globally before you can configure a VLAN interface.	• IPv4 ACLs • IPv6 ACLs
VLAN ACL	• VLANs	• IPv4 ACLs • IPv6 ACLs • MAC ACLs

Application	Supported Interfaces	Types of ACLs Supported
VTY ACL	• VTYs	• IPv4 ACLs • IPv6 ACLs

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. Ingress VTY ACL
5. Egress VTY ACL
6. Egress router ACL
7. Egress VACL

If the packet is bridged within the ingress VLAN, the device does not apply router ACLs.

Figure 4: Order of ACL Application

The following figure shows the order in which the device applies ACLs.

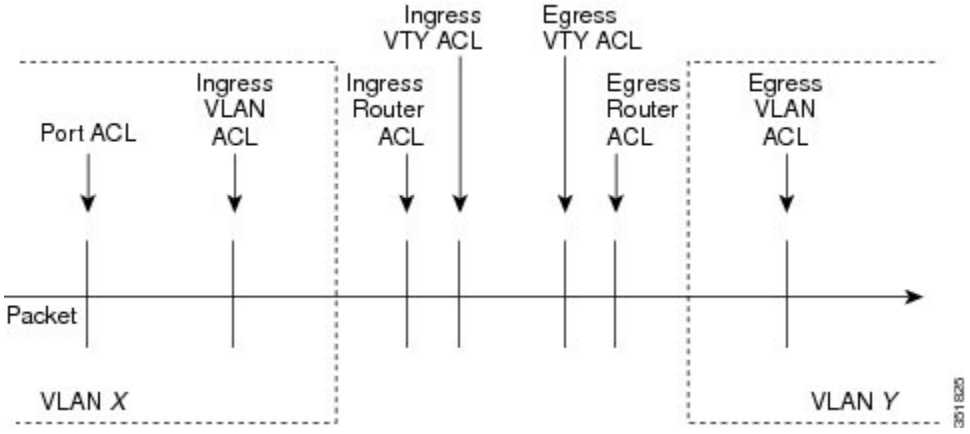
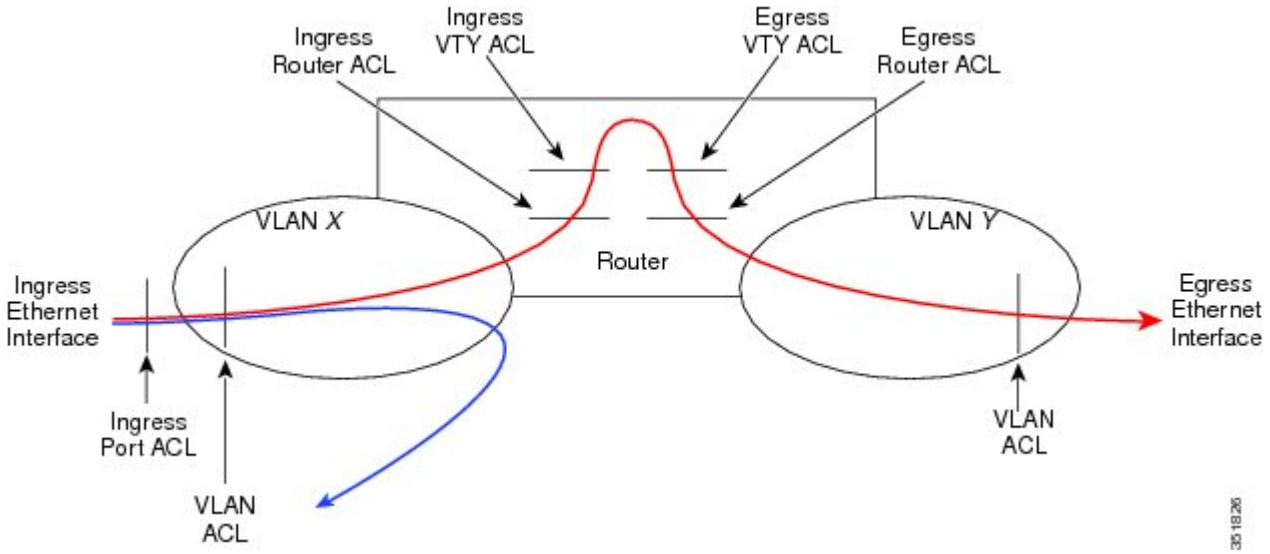


Figure 5: ACLs and Packet Flow

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.



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.

Protocols for IP ACLs and MAC 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.

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 ACLs, IPv6 ACLs, or MAC ACLs.

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 rule:

```
deny ipv6 any any
```

This implicit rule ensures that the device denies unmatched IPv6 traffic.

Note IPv6 nd-na, nd-ns, router-advertisement, and router-solicitation packets will not be permitted as the implicit permit rules on IPv6 ACL. You must add the following rules explicitly to allow them:

- **permit icmp any any nd-na**
- **permit icmp any any nd-ns**
- **permit icmp any any router-advertisement**
- **permit icmp any any router-solicitation**

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:
  - Differentiated Services Code Point (DSCP) value
  - Established TCP connections
  - Layer 4 protocol
  - TCP and UDP ports
  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set
- IPv6 ACLs support the following additional filtering options:
  - Differentiated Services Code Point (DSCP) value
  - Encapsulating Security Payload

- Established TCP connections
- Layer 4 protocol
- Payload Compression Protocol
- Stream Control Transmission Protocol (SCTP)
- SCTP, TCP, and UDP ports
- TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set

- MAC ACLs support the following additional filtering options:
  - Class of Service (CoS)
  - Layer 3 protocol (Ethertype)
  - VLAN ID

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, Cisco NX-OS 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. Cisco NX-OS supports logical operators in only the ingress direction.

The device stores operator-operand couples in registers called logical operator units (LOUs). The LOU usage for each type of operator is as follows:

eq
Is never stored in an LOU

gt
Uses 1 LOU

lt
Uses 1 LOU

neq
Uses 1 LOU

range
Uses 1 LOU

Time Ranges

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.

Note The order of rules in a time range does not affect how a device evaluates whether a time range is active. Cisco NX-OS includes sequence numbers in time ranges to make editing the time range easier.

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.

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, policy-based routing (PBR), 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.

Note Policy-based routing (PBR) ACLs do not support deny access control entries (ACEs) or **deny** commands to configure a rule.

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.

Note The device does not support interface-level ACL statistics.

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.

About Per-Port Stats

Beginning Cisco NX-OS Release 9.2(2v), if required, you can get generate per-port stats even when you apply the same IPv4 or an IPv6 ACL to multiple interfaces.

Per-port stats have the following guidelines and limiations:

- Per-port stats for ACLs are only applicable for physical ports.

- Maximum three ingress TCAMS can be carved as per-port stats.
- Maximum two egress TCAMS can be carved as per-port stats.
- The maximum TCAM entries with per-port stats is 240 per IB.
- Per-port stats are not supported on sub interfaces.
- Per-port stats are always atomic.

Atomic ACL Updates

By default, when a supervisor module of a Cisco NX-OS 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
```

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.

ACL TCAM Regions

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

On Cisco Nexus 3400-S Series switches, the egress TCAM size is 1.5K and is divided into two 256 slices and two 512 slices. The ingress TCAM size is 3.5K and is divided into six 256 slices and four 512 slices. A slice is the unit of allocation. A slice can be allocated to one region only.

You can create IPv4, IPv6, port ACLs, VLAN ACLs, and router ACLs, and you can match IPv4, IPv6 and MAC addresses for QoS. However, Cisco NX-OS cannot support all of them simultaneously. You must remove or reduce the size of the existing TCAM regions (TCAM carving) to enable the IPv6, MAC, or other desired TCAM regions. For every TCAM region configuration command, the system evaluates if the new change can be fit in the TCAM. If not, it reports an error, and the command is rejected. You must remove or reduce the size of existing TCAM regions to make room for new requirements.

ACL TCAM region sizes have the following guidelines and limitations:

- When a VACL region is configured, it is configured with the same size in both the ingress and egress directions. If the region size cannot fit in either direction, the configuration is rejected.
- The SUP region occupies 256 entries of 320 bits width.

The following table summarizes the regions that need to be configured for a given feature to work. The region sizes should be selected based on the scale requirements of a given feature.

Table 10: Features per ACL TCAM Region

Feature Name	Region Name
Port ACL	ifacl: For IPv4 port ACLs ipv6-ifacl: For IPv6 port ACLs mac-ifacl: For MAC port ACLs
Port QoS (QoS classification policy applied on Layer 2 ports or port channels)	ing-l2-qos: For classifying ingress Layer 2 packets
VACL (can be carved in both directions)	vacl: For IPv4 packets ipv6-vacl: For IPv6 packets mac-vacl: For non-IP packets
RACL	racl: For IPv4 RACLs ipv6-racl: For IPv6 RACLs e-racl: For egress IPv4 RACLs e-ipv6-racl: For egress IPv6 RACLs
Layer 3 QoS (QoS classification policy applied on Layer 3 ports or port channels)	ing-l3-vlan-qos: For classifying IPv4 packets
Rx SPAN on 40G ports	span
SPAN filters	span
BFD, DHCP relay, or DHCPv6 relay	ing-sup
CoPP	ing-sup

Feature Name	Region Name
System-managed ACLs	ing-sup
vPC convergence **Note** This region boosts the convergence times when a vPC link goes down and traffic needs to be redirected to the peer link.	vpc-convergence **Note** Setting this region size to 0 might affect the convergence times of vPC link failures.

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.

Guidelines and Limitations for IP ACLs

IP ACLs have the following configuration guidelines and limitations:

- We recommend that you perform ACL configuration using the Session Manager. This feature allows you to verify ACL configuration and confirm that the resources that are required by the configuration are available before committing them to the running configuration. This recommendation is especially useful for ACLs that include more than 1000 rules. For more information about Session Manager, see the *Cisco Nexus 3400-S NX-OS System Management Configuration Guide*.
- Duplicate ACL entries with different sequence numbers are allowed in the configuration. However, these duplicate entries are not programmed in the hardware access-list.
- Only 62 unique ACLs can be configured. Each ACL takes one label. If the same ACL is configured on multiple interfaces, the same label is shared. If each ACL has unique entries, the ACL labels are not shared, and the label limit is 62. The combinations are as follows:
  - 31 unique PACLs of 5 bits and 31 unique Layer 2 QoS of 5 bits
  - 15 unique egress RACLs of 4 bits
  - 7 unique RACLs of 3 bits and 31 unique Layer 3 QoS of 5 bits
  - 15 unique VACLs of 4 bits
- TCAM carving region can be either 128 or multiples of 128 for RACL + VACL.
- VLAN QoS and egress QoS are not supported.
- UDF with odd offset and 2-byte match is not supported.
- ICMP type and code match are not supported.
- Packet length match is not supported.

- ACL statistics are not supported for CRC packets.
- ACL log options are not supported.
- TCP flags are not supported on an egress RACL in Cisco NX-OS Release 9.2(2t).

  Beginning Cisco NX-OS release 9.2(2v), TCP flags are supported on egress RACL.
- ACLs with match DSCP are supported only in the pacl-all regions.
- RACL does not affect sup-traffic.
- VACL redirects are not supported.
- Set COS and set DSCP combination is not supported for Layer 3 QoS.
- Usually, 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 many 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.

  Policers prevent redirected packets from overwhelming the supervisor module.
- When you apply an ACL that uses time ranges, the device updates the ACL entries whenever a time range that is 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 *Cisco Nexus 3400-S NX-OS Interfaces Configuration Guide*.
- 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.
- When you apply an undefined ACL to an interface, the system treats the ACL as empty and permits all traffic.
- IP tunnels do not support ACLs or QoS policies.
- IPv4 and IPv6 ACL logging is not supported.
- ACL logging for VACLs is not supported.
- For Network Forwarding Engine (NFE)-enabled switches, ingress RACLs matching the outer header of the tunnel interface are not supported.
- The switch hardware does not support range checks (Layer 4 operations) in the egress TCAM. Therefore, ACL and QoS policies with a Layer 4 operations-based classification must be expanded to multiple entries in the egress TCAM. Make sure to consider this limitation for egress TCAM space planning.
- TCAM resources are shared in the following scenarios:

  - When a routed ACL is applied to multiple switched virtual interfaces (SVIs) in the ingress direction.
  - VACL (VLAN ACL) is applied to multiple VLANs.

- Atomic ACL update is supported for all the ingress and egress ACL features except for the Multihop BFD and CoPP features.
- Label sharing is supported only for the same policy on different interfaces within the same ASIC.
- ACL label sharing is not supported for egress RACL and SPAN.
- ACL statistics are not supported for the following:
  - BFD
  - DHCP - IPv4 and IPv6
- Cisco Nexus 3400-S Series switches support the following on the ACLs:
  - Statistics support
  - Label sharing
- When you enable the counters for the ACL TCAM entries using the hardware profile acl-stats module*xx* command, the input discard field in the show interface is always zero.
- IPv6 wildcard mask is not supported on Cisco Nexus 3400-S Series switches.
- UDF is supported for both, IPv4 and IPv6 RACL and SPAN.
- UDF is not supported on PACL.
- TCAM regions for traffic storm control are carved by default.
- Beginning with Cisco NX-OS Release 9.2(2v), UDF support is extended to IPv6 RACL and PACL.
- Beginning Cisco NX-OS Release 9.2(2v), Remote Directory Memory Access (RDMA) and Explicit Congestion Notification (ECN) bits can be matched with ACLs only over UDF.
- Beginning with Cisco NX-OS Release 9.2(2v), the following ACL TCAM regions are introduced:
  - Ingress PACL IPv4 and IPv6 (ifacl-all)
  - Ingress RACL IPv4 and IPv6 (racl-all)
- ACEs with the same IPv4 or IPv6 addresses and different masks are not supported.
- Beginning with Cisco NX-OS Release 9.3(5), the following ACL TCAM region is introduced:
  - Egress PACL IPv4 and IPv6 (e-ifacl-all)
- Beginning with Cisco NX-OS Release 9.3(5), egress PACL is supported on Cisco Nexus 3408-S and 3432D-S switches. The following limitations are applicable:
  - ACL logging and redirect are not supported in the egress direction.
  - IP compression is not supported in the egress direction.
  - Egress PACL does not have per-interface label and is applied as per global policy. We recommend that the same egress PACL policy is applied on all Layer 2 interfaces.

- If different egress PACL policies are applied to different Layer 2 ports, the TCAM programming behavior depends on the order in which the egress PACLs are configured.
- TCP/UDP port range options are supported by ACE expansion. There is no LOU support.

- Only a host match is supported when the IPv6 ACL is applied to the Layer 2 and Layer 3 interface for the traffic flowing in the specified direction.

Default Settings for IP ACLs

This table lists the default settings for IP ACL parameters.

Table 11: Default IP ACL Parameters

Parameters	Default
IP ACLs	No IP ACLs exist by default
IP ACL entries	1024
ACL rules	Implicit rules apply to all ACLs
Object groups	No object groups exist by default
Time ranges	No time ranges exist by default

Configuring IP ACLs

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 that are required by the configuration are available before committing them to the running configuration. This feature is especially useful for ACLs that include more than about 1000 rules.

Procedure

	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:
	• **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-ip-prefix* | *source-ip-mask*} {*destination-ip-prefix* | *destination-ip-mask*} **Example:** `switch(config-acl)# 10 permit ipv6 1::1 2::2 3::3 4::4`	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 IPv4 and IPv6 access lists, you can specify a source and destination IPv4 or IPv6 prefix, which matches only on the first contiguous bits, or you can specify a source and destination IPv4 or IPv6 wildcard mask, which matches on any bit in the address.
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	**reload module** *xx* **Example:** `switch(config)# reload module 10`	Reloads the switch.
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:**	Copies the running configuration to the startup configuration.

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

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.

Procedure

	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. The **permit** and **deny** commands support many ways of identifying traffic.
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.

	Command or Action	Purpose
		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.
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.

Creating a VTY ACL

You can configure a VTY ACL to control access to all IPv4 or IPv6 traffic over all VTY lines in the ingress or egress direction.

Before you begin

Set identical restrictions on all the virtual terminal lines because a user can connect to any of them.

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, which is especially useful for ACLs that include more than about 1000 rules.

Procedure

	Command or Action	Purpose

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

Step 2	{**ip** | **ipv6**} **access-list** *name* **Example:** `switch(config)# ip access-list vtyacl`	Creates an ACL and enters IP access list configuration mode for that ACL. The maximum length for the *name* argument is 64 characters.
Step 3	{**permit** | **deny**} *protocol source destination* [**log**] [**time-range** *time*] **Example:** `switch(config-ip-acl)# permit tcp any any`	Creates an ACL rule that permits TCP traffic from and to the specified sources.

Step 4	**exit**
	Example:
	`switch(config-ip-acl)# exit`
	`switch(config)#`
		Exits IP access list configuration mode.

Step 5	**line vty**
	Example:
	`switch(config)# line vty`
	`switch(config-line)#`
		Specifies the virtual terminal and enters line configuration mode.

Step 6	{**ip** | **ipv6**} **access-class** *name* {**in** | **out**} **Example:** `switch(config-line)# ip access-class vtyacl out`	Restricts incoming or outgoing connections to and from all VTY lines using the specified ACL. The maximum length for the *name* argument is 64 characters.
Step 7	(Optional) **show** {**ip** | **ipv6**} **access-lists** **Example:** `switch# show ip access-lists`	Displays the configured ACLs, including any VTY ACLs.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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.

Procedure

	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.

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.

Procedure

	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.

Configuring ACL TCAM Region Sizes

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware.

Note

- Once you apply a template , the **hardware access-list tcam region** command in this section will not work. You must uncommit the template in order to use the command.
- For information on configuring QoS TCAM carving, see the *Cisco Nexus 3400-S NX-OS Quality of Service Configuration Guide*.

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
Step 2	[**no**] **hardware access-list tcam region** *region tcam-size* **Example:** `switch(config)# hardware access-list tcam region ifacl 256`	Changes the ACL TCAM region size. You can use the **no** form of this command to revert to the default TCAM region size.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 4	(Optional) **show hardware access-list tcam region** **Example:** `switch(config)# show hardware access-list tcam region`	Displays the TCAM sizes that will be applicable on the next reload of the device.
Step 5	**reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** The new size values are effective only after you enter **copy running-config startup-config** + **reload** or reload all line card modules.

Example

The following example shows how to change the size of the RACL TCAM region on a Cisco Nexus 3400-S Series switch:

```
switch(config)# hardware access-list tcam region racl 256
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'
switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

This example shows how to display the TCAM region sizes to verify your changes:

```
switch(config)# show hardware access-list tcam region
IPV4 PACL [ifacl] size = 0
IPV6 PACL [ipv6-ifacl] size = 0
MAC PACL [mac-ifacl] size = 0
IPV4 VACL [vacl] size = 0
IPV6 VACL [ipv6-vacl] size = 0
MAC VACL [mac-vacl] size = 0
IPV4 RACL [racl] size = 256
IPV6 RACL [ipv6-racl] size = 0
Egress IPV4 RACL [e-racl] size = 0
Egress IPV6 RACL [e-ipv6-racl] size = 0
```

```
SPAN [span] size = 0
VPC Convergence/ES-Multi Home [vpc-convergence] size = 0
Ingress L2 QOS [ing-l2-qos] size = 0
Ingress L3/VLAN QOS [ing-l3-vlan-qos] size = 128
Ingress SUP [ing-sup] size = 256
Egress L2 QOS [egr-l2-qos] size = 0
Egress L3/VLAN QOS [egr-l3-vlan-qos] size = 0
```

This example shows how to revert to the default RACL TCAM region size:

```
switch(config)# no hardware profile tcam region racl 512
[SUCCESS] New tcam size will be applicable only at boot time.
You need to 'copy run start' and 'reload'
switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

Configuring TCAM Carving

The default TCAM region configuration varies by platform and does not accommodate all TCAM regions. To enable any desired regions, you must decrease the TCAM size of one region and then increase the TCAM size for the desired region.

Note For information on configuring QoS TCAM carving, see the *Cisco Nexus 3400-S NX-OS Quality of Service Configuration Guide*.

The following tables list the default sizes for the ingress and egress TCAM regions.

Table 12: Default TCAM Region Configuration (Ingress)

Region Name	Size	Width	Total Size
IPv4 RACL	256	160 bits	512 (80 bits)
Layer 3 QoS	128	320 bits	512 (80 bits)
System	256	320 bits	1024 (80 bits)

Attention To keep all modules synchronized, you must reload all line card modules or enter **copy running-config startup-config** + **reload** to reload the device. Multiple TCAM region configurations require only a single reload. You can wait until you complete all of your TCAM region configurations before you reload the device.

Depending on the configuration, you might exceed the TCAM size or run out of slices.

If you exceed the 1K ingress limit for all TCAM regions when you configure a TCAM region, the following message appears:

```
ERROR: Aggregate TCAM region configuration exceeded the available Ingress TCAM space. Please
 re-configure.
```

If you exceed the 1K egress limit for all TCAM regions when you configure a TCAM region, the following message appears:

```
ERROR: Aggregate TCAM region configuration exceeded the available Egress TCAM space. Please
 re-configure.
```

If TCAM for a particular feature is not configured and you try to apply a feature that requires TCAM carving, the following message appears:

```
 ERROR: Module x returned status: TCAM region is not configured. Please configure TCAM
region and retry the command.
```

Configuring UDF-Based Router ACLs

This feature enables the device to match on user-defined fields (UDFs) and to apply the matching packets to IPv4 and IPv6 RACLs.

Procedure

	Command or Action	Purpose

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

Step 2	**udf** *udf-name offset-base offset length*
	Example:
	`switch(config)# udf pktoff10 packet-start 10 2`
	Example:
	`switch(config)# udf pktoff10 header outer l3 20 2`
		Defines the UDF as follows:
		• *udf-name*—Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name.
		• *offset-base*—Specifies the UDF offset base as follows, where **header** is the packet header to consider for the offset: {**packet-start** | **header** {**outer** | **inner** {**l3** | **l4**}}}.
		• *offset*—Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0.
		• *length*—Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs.
		You can define multiple UDFs, but Cisco recommends defining only required UDFs.

Step 3	**hardware access-list tcam region racl qualify** {**udf** *udf-name*} | **v6udf** *v6udf-name*	Attaches the UDFs to the racl TCAM region, which applies to IPv4 or IPv6 router ACLs.

	Command or Action	Purpose
	Example: `switch(config)# hardware access-list tcam region racl qualify udf pktoff10`	The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide. **Note** When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information, see Configuring ACL TCAM Region Sizes, on page 163.
Step 4	**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.
Step 5	Required: **reload** **Example:** `switch(config)# reload`	Reloads the device. **Note** Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload**.

Step 6	**ip access-list** *udf-acl*
	Example:
	`switch(config)# ip access-list udfacl`
	`switch(config-acl)#`
		Creates an IPv4 access control list (ACL) and enters IP access list configuration mode.

Step 7	Enter one of the following commands:
	• **permit udf** *udf-name value mask*
	• **permit ip** *source destination* **udf** *udf-name value mask*
	Example:
	`switch(config-acl)# permit udf pktoff10 0x1234 0xffff`
	Example:
	`switch(config-acl)# permit ip any any udf pktoff10 0x1234 0xffff`
		Configures the ACL to match only on UDFs (example 1) or to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2). The range for the *value* and *mask* arguments is from 0x0 to 0xffff.
		A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs.

Step 8	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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
- VLAN interfaces
- Management interfaces

ACLs applied to these interface types are considered router ACLs.

Note Egress router ACLs are not supported on subinterfaces.

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.

Procedure

	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*[**.** *number*]
	• **interface port-channel** *channel-number*
	• **interface vlan** *vlan-id*
	• **interface mgmt** *port*
	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 access-group** *access-list* {**in** | **out**} • **ipv6 traffic-filter** *access-list* {**in** | **out**} **Example:** `switch(config-if)# ip access-group acl1 in`	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) **show running-config aclmgr** **Example:**	Displays the ACL configuration.

	Command or Action	Purpose
	`switch(config-if)# show running-config aclmgr`	
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.

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.

Procedure

	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** **Example:** `switch(config-if)# ip port access-group acl-l2-marketing-group 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.
Step 4	(Optional) **show running-config aclmgr** **Example:** `switch(config-if)# show running-config aclmgr`	Displays the ACL configuration.

	Command or Action	Purpose
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.

Applying an IP ACL as a VACL

You can apply an IP ACL as a VACL.

Configuring Per-Port Stats

Procedure

	Command or Action	Purpose

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

Step 2	**hardware access-list tcam region racl 128 per-port-stats** **Example:** `switch(config)# hardware access-list tcam region racl 128 per-port-stats`	Configures the per-port status for ingress RACL.
Step 3	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.
Step 4	**reload** **Example:** `switch(config)# reload`	Reloads the device.

Verifying the IP ACL Configuration

To display IP ACL configuration information, perform one of the following tasks.

Command	Purpose
show hardware access-list tcam region	Displays the TCAM sizes that will be applicable on the next reload of the device.
show ip access-lists	Displays the IPv4 ACL configuration.
show ipv6 access-lists	Displays the IPv6 ACL configuration.
show running-config aclmgr [**all**]	Displays the ACL running configuration, including the IP ACL configuration and the interfaces to which IP ACLs are applied. **Note** This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config acllog	Displays the ACL log startup configuration.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration. **Note** This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Monitoring and Clearing IP ACL Statistics

To monitor or clear IP ACL statistics, use one of the commands in this table.

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.

Configuration Examples for IP ACLs

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
```

The following example shows how to create a VTY ACL named single-source and apply it on input IP traffic over the VTY line. This ACL allows all TCP traffic through and drops all other IP traffic:

```
ip access-list single-source
  permit tcp 192.168.7.5/24 any
  exit
  line vty
  ip access-class single-source in
  show ip access-lists
```

The following example shows how to configure a UDF-based port ACL:

```
switch# configure terminal
switch(config)# hardware access-list tcam region racl 256
switch(config)# udf pktoff10 packet-start 10 2
```

```
switch(config)# udf pktoff20 packet-start 10 1
switch(config)# hardware access-list tcam region racl qualify udf pktoff10 pktoff20

switch# configure terminal
switch(config)# ip access-list udfacl
switch(config-acl)# statistics per-entry
switch(config-acl)# 10 permit ip any any udf pktoff10 0x1234 0xffff

switch# configure terminal
switch(config)# interface Ethernet1/1
switch(config-if)# ip access-group udfacl in
switch(config-if)# no switchport
switch(config-if)# no shutdown
```

The following example shows how to configure an egress PACL:

```
switch# configure terminal
switch(config)# hardware access-list tcam region e-ifacl-all 256
switch(config)# copy running-config startup-config
switch(config)# reload
switch(config)# ip access-list acl-01
switch(config-acl)# statistics per-entry
 10 permit ip 30.1.1.2/32 60.1.1.2/32
 20 permit ip 30.1.1.3/32 60.1.1.3/32
 30 permit ip 30.1.1.4/32 60.1.1.4/32
 40 permit ip 30.1.1.5/32 60.1.1.5/32
 50 permit ip 30.1.1.6/32 60.1.1.6/32
switch(config)# interface ethernet 2/2
switch(config-if)# witchport mode trunk
switch(config-if)# switchport trunk allowed vlan 20,120
switch(config-if)# ip port access-group EP_30 out
```

Configuring Object Groups

You can use object groups to specify source and destination addresses and protocol ports in IPv4 ACL and IPv6 ACL rules.

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 *Cisco Nexus 3400-S NX-OS System Management Configuration Guide*.

Creating and Changing an IPv4 Address Object Group

You can create and change an IPv4 address group object.

Procedure

	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 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*/*prefix-len* • [*sequence-number*] *IPv4-address network-wildcard* **Example:** `switch(config-ipaddr-ogroup)# host 10.99.32.6`	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. You can specify a prefix length for an IPv4 object group, which matches only on the first contiguous bits, or you can specify a wildcard mask, which matches on any bit in the address.
Step 4	Enter one of the following commands: • **no** [*sequence-number*] • **no host** *IPv4-address* • **no** *IPv4-address*/*prefix-len* • **no** *IPv4-address network-wildcard* **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.

Creating and Changing an IPv6 Address Object Group

You can create and change an IPv6 address group object.

Procedure

	Command or Action	Purpose

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

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 to specify a network of hosts. You can specify a prefix length for an IPv6 object group, which matches only on the first contiguous bits.
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.

Creating and Changing a Protocol Port Object Group

You can create and change a protocol port object group.

Procedure

	Command or Action	Purpose

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

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 only the port number that you specify. • **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 numbers 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.

	Command or Action	Purpose
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.

Removing an Object Group

You can remove an IPv4 address object group, an IPv6 address object group, or a protocol port object group.

Procedure

	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 specified object group.
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.

Verifying the Object-Group Configuration

To display object-group configuration information, enter one of the following commands:

Command	Purpose
show object-group	Displays the object-group configuration.
show {**ip** | **ipv6**} **access-lists** *name* [**expanded**]	Displays expanded statistics for the ACL configuration.
show running-config aclmgr	Displays the ACL configuration, including object groups.

Configuring Time-Ranges

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 *Cisco Nexus 3400-S NX-OS System Management Configuration Guide*.

Creating a Time-Range

You can create a time range on the device and add rules to it.

Procedure

	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

	Command or Action	Purpose
	Example: `switch(config-time-range)# absolute start 1:00 15 march 2013`	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 may 2013`	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.

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.

Procedure

	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)#`
		Enters time-range configuration mode for the specified time range.

Step 3	(Optional) [*sequence-number*] **periodic** *weekday time* **to** [*weekday*] *time* **Example:**	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.

	Command or Action	Purpose
	`switch(config-time-range)# periodic monday 00:00:00 to friday 23:59:59`	
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 2013`	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 may 2013`	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** **Example:** `switch(config-time-range)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Removing a Time-Range

You can remove a time range from the device.

Before you begin

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.

Procedure

	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.

Changing Sequence Numbers in a Time Range

You can change all the sequence numbers assigned to rules in a time range.

Procedure

	Command or Action	Purpose

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

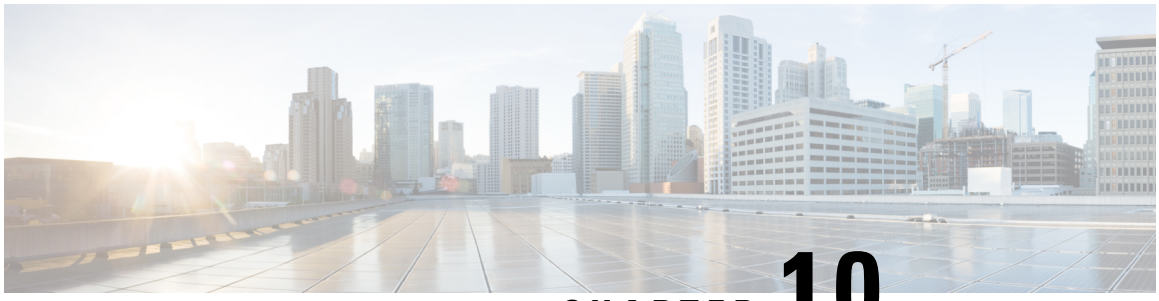
Step 2	**resequence time-range** *name starting-sequence-number increment* **Example:**	Assigns sequence numbers to the rules contained in the time range, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number

	Command or Action	Purpose
	`switch(config)# resequence time-range daily-workhours 100 10` `switch(config)#`	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.

Verifying the Time-Range Configuration

To display time-range configuration information, perform one of the following tasks.

Command	Purpose
show time-range	Displays the time-range configuration.
show running-config aclmgr	Displays ACL configuration, including all time ranges.



CHAPTER 10

Configuring MAC ACLs

This chapter describes how to configure MAC access lists (ACLs) on Cisco NX-OS devices.

This chapter contains the following sections:

- About MAC ACLs, on page 183
- Guidelines and Limitations for MAC ACLs, on page 183
- Default Settings for MAC ACLs, on page 184
- Configuring MAC ACLs, on page 184
- Verifying the MAC ACL Configuration, on page 189
- Monitoring and Clearing MAC ACL Statistics, on page 189
- Configuration Example for MAC ACLs, on page 189

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.

Guidelines and Limitations for MAC ACLs

MAC ACLs have the following configuration guidelines and limitations:

- MAC ACLs apply to ingress traffic only.
- If you try to apply too many ACL entries, the configuration might be rejected.
- MAC packet classification is not supported.

Default Settings for MAC ACLs

This table lists the default settings for MAC ACL parameters.

Table 13: Default MAC ACLs Parameters

Parameters	Default
MAC ACLs	No MAC ACLs exist by default
ACL rules	Implicit rules apply to all ACLs

Configuring MAC ACLs

Creating a MAC ACL

You can create a MAC ACL and add rules to it.

Procedure

	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)# 100 permit mac 00c0.4f00.0000 0000.00ff.ffff any 0x0806`	Creates a rule in the MAC ACL. The **permit** and **deny** commands support many ways of identifying traffic.
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:**	Displays the MAC ACL configuration.

	Command or Action	Purpose
	`switch(config-mac-acl)# show mac access-lists acl-mac-01`	
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.

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 on which a MAC ACL is configured.

Procedure

	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 00c0.4f00.0000 0000.00ff.ffff any 0x0806`	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.
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.

	Command or Action	Purpose
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.

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.

Procedure

	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.

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

Removing a MAC ACL

You can remove a MAC ACL from the device.

Procedure

	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.

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.

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 Ethernet interfaces
- Layer 2 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.

Procedure

	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.

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 the 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.

Applying a MAC ACL as a VACL

You can apply a MAC ACL as a VACL.

Verifying the MAC ACL Configuration

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

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. **Note** This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration. **Note** This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Monitoring and Clearing MAC ACL Statistics

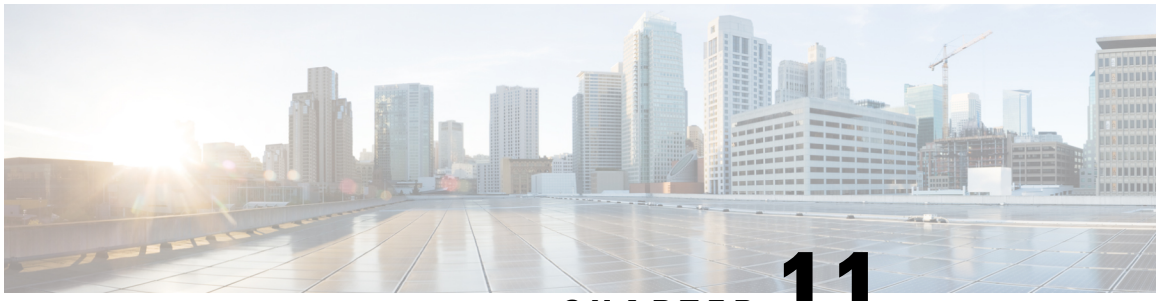
To monitor or clear MAC ACL statistics, use one of the commands in this table.

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 MAC ACLs.

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 0x0806
interface ethernet 2/1
  mac port access-group acl-mac-01
```



CHAPTER 11

Configuring VLAN ACLs

This chapter describes how to configure VLAN access lists (ACLs) on Cisco NX-OS devices.

This chapter includes the following sections:

- About VLAN ACLs, on page 191
- Prerequisites for VACLs, on page 192
- Guidelines and Limitations for VACLs, on page 192
- Default Settings for VACLs, on page 193
- Configuring VACLs, on page 193
- Verifying the VACL Configuration, on page 196
- Monitoring and Clearing VACL Statistics, on page 196
- Configuration Example for VACLs, on page 196
- Additional References for VACLs, on page 197

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).

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.

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 device.

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.

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.

Note The device does not support interface-level VACL statistics.

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.

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.

Guidelines and Limitations for VACLs

VACLs have the following configuration guidelines:

- Cisco recommends using the Session Manager to configure ACLs. 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. For more information about Session Manager, see the *Cisco Nexus 3400-S NX-OS System Management Configuration guide*.
- If you try to apply too many ACL entries, the configuration might be rejected.
- VACL redirects to SPAN destination ports are not supported.
- VACL logging is not supported.
- Deny statements are not supported on VACLs. Alternatively, you can use permit statements with the action 'drop' to achieve a similar outcome.
- To clear VACL counters, you must ensure that you have active VLAN filters configured.

Default Settings for VACLs

This table lists the default settings for VACL parameters.

Table 14: Default VACL Parameters

Parameters	Default
VACLs	No IP ACLs exist by default
ACL rules	Implicit rules apply to all ACLs

Configuring VACLs

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 exist and are configured to filter traffic in the manner that you need for this application.

Procedure

	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*	Specifies an ACL for the access-map entry.

	Command or Action	Purpose

	Example:
	`switch(config-access-map)# match mac address acl-ip-lab`
	Example:
	`switch(config-access-map)# match mac address acl-mac-01`	

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 the **drop**, **forward**, and **redirect** options.
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.

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.

Procedure

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

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
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.

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.

Procedure

	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.

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

Verifying the VACL Configuration

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

Command	Purpose
show running-config aclmgr [**all**]	Displays the ACL configuration, including the VACL-related configuration. **Note** This command displays the user-configured ACLs in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show startup-config aclmgr [**all**]	Displays the ACL startup configuration. **Note** This command displays the user-configured ACLs in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the 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.

Monitoring and Clearing VACL Statistics

To monitor or clear VACL statistics, use one of the commands in this table.

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 VACLs.

Configuration Example for VACLs

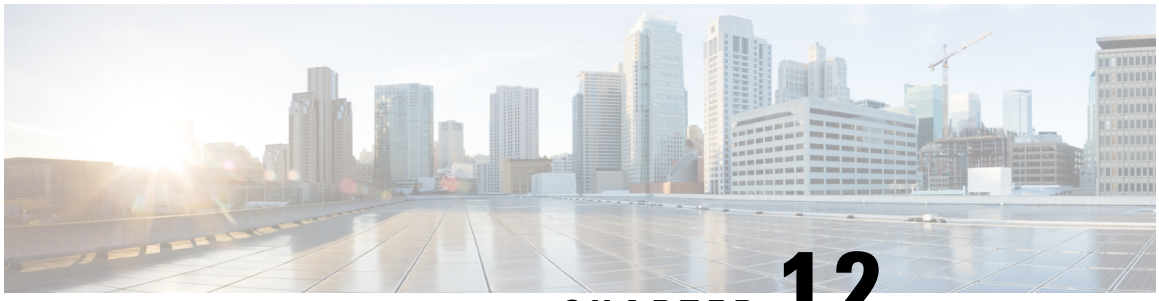
The following example shows how to configure a VACL to forward traffic permitted by a MAC ACL named acl-mac-01 and how to apply the VACL to VLANs 50 through 82:

```
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
```

Additional References for VACLs

Related Documents

Related Topic	Document Title
QoS configuration	*Cisco Nexus 3400-S NX-OS Quality of Service Configuration Guide*



CHAPTER 12

Configuring DHCP

This chapter describes how to configure the Dynamic Host Configuration Protocol (DHCP) on a Cisco NX-OS device.

This chapter includes the following sections:

- About DHCP Snooping, on page 199
- About the DHCP Relay Agent, on page 203
- About the DHCPv6 Relay Agent, on page 206
- About DHCP Client, on page 207
- Prerequisites for DHCP, on page 207
- Guidelines and Limitations for DHCP, on page 207
- Default Settings for DHCP, on page 208
- Configuring DHCP, on page 209
- Configuring DHCPv6, on page 226
- Enabling DHCP Client, on page 231
- Verifying the DHCP Configuration, on page 232
- Displaying IPv6 RA Guard Statistics, on page 233
- Displaying DHCP Snooping Bindings, on page 233
- Clearing the DHCP Snooping Binding Database, on page 233
- Monitoring DHCP, on page 234
- Clearing DHCP Snooping Statistics, on page 234
- Clearing DHCP Relay Statistics, on page 234
- Clearing DHCPv6 Relay Statistics, on page 234
- Configuration Examples for DHCP, on page 234
- Configuration Examples for DHCP Client, on page 235
- Additional References for DHCP, on page 235

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.

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 Cisco NX-OS 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.

Note For DHCP snooping to function properly, all DHCP servers must be connected to the device through trusted interfaces.

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.

Note The DHCP snooping binding database is also referred to as the DHCP snooping binding table.

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.

DHCP Snooping in a vPC Environment

A virtual port channel (vPC) allows two Cisco NX-OS switches to appear as a single logical port channel to a third device. The third device can be a switch, a server, or any other networking device that supports port channels.

In a typical vPC environment, DHCP requests can reach one vPC peer switch, and the responses can reach the other vPC peer switch, resulting in a partial DHCP (IP-MAC) binding entry in one switch and no binding entry in the other switch. As a result, DHCP snooping and associated features such as dynamic ARP inspection (DAI) and IP Source Guard are disrupted. This issue is addressed by using Cisco Fabric Service over Ethernet (CFSoE) distribution to ensure that all DHCP packets (requests and responses) appear on both switches, which helps in creating and maintaining the same binding entry on both switches for all clients behind the vPC link.

CFSoE distribution also allows only one switch to forward the DHCP requests and responses on the vPC link. In non-vPC environments, both switches forward the DHCP packets.

Synchronizing DHCP Snooping Binding Entries

The dynamic DHCP binding entries should be synchronized in the following scenarios:

- When the remote vPC is online, all the binding entries for that vPC link should be synchronized with the peer.
- When DHCP snooping is enabled on the peer switch, the dynamic binding entries for all vPC links should be synchronized with the peer.

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.

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 Cisco NX-OS 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 Cisco NX-OS 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-ifindex (for non-vPCs) or vlan-vpcid (for vPCs), from which the packet is received (the circuit ID suboption).

Note For vPC peer switches, the remote ID suboption contains the vPC switch MAC address, which is unique in both switches. This MAC address is computed with the vPC domain ID. The Option 82 information is inserted at the switch where the DHCP request is first received before it is forwarded to the other vPC peer switch.

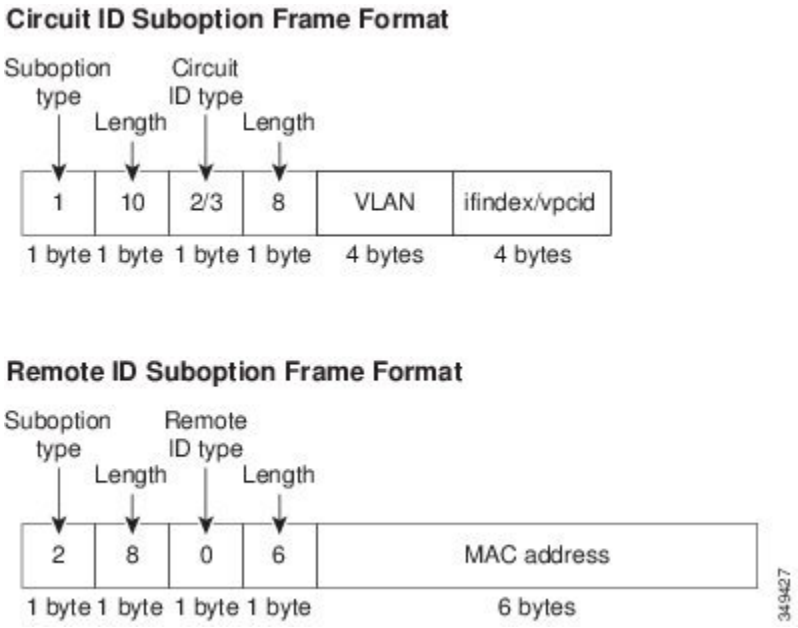
3. The device forwards the DHCP request that includes the Option 82 field to the DHCP server.
4. The DHCP server receives the packet. If the server is Option 82 capable, it can use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of 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 Cisco NX-OS device. The Cisco NX-OS device verifies that it originally inserted the Option 82 data by inspecting the remote ID and possibly the circuit ID fields. The Cisco NX-OS 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

This figure shows the packet formats for the remote ID suboption and the circuit ID suboption. The Cisco NX-OS 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 6: Suboption Packet Formats



About the DHCP Relay Agent

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.

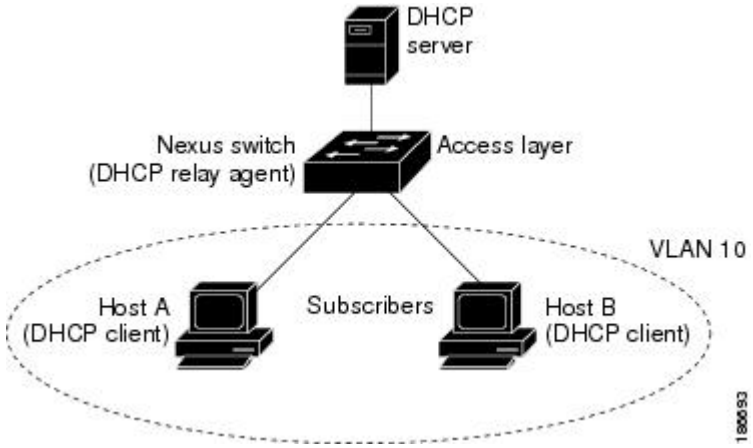
Note When the device relays a DHCP request that already includes Option 82 information, the device forwards the request with the original Option 82 information without altering it.

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 7: 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.

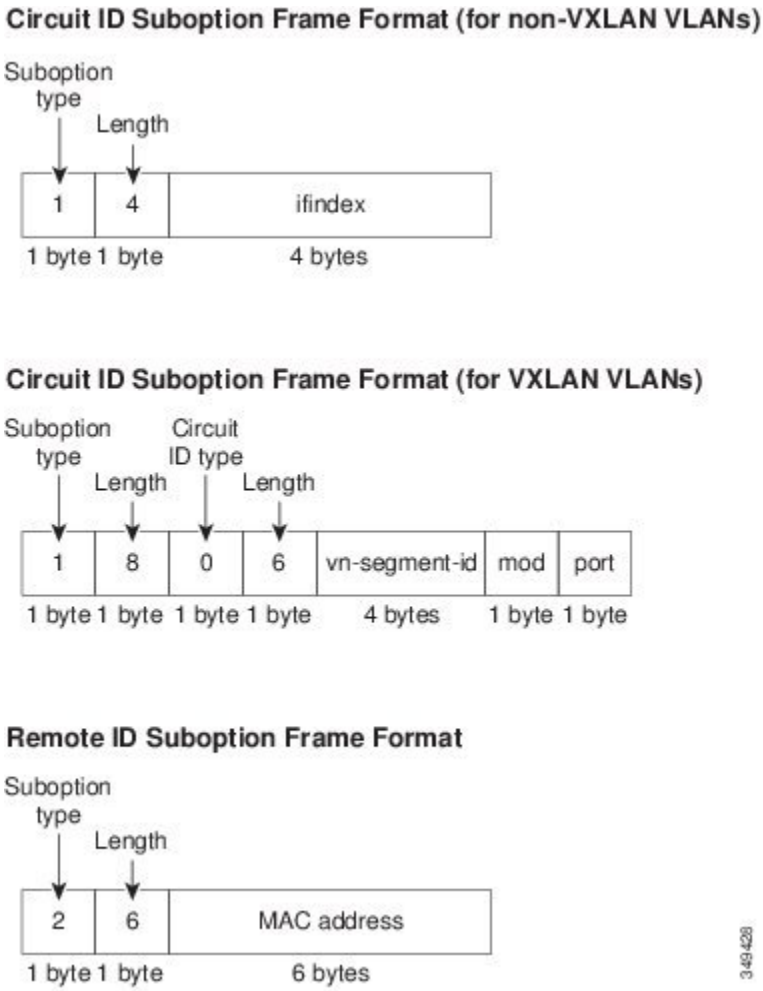


When you enable Option 82 for the DHCP relay agent on the Cisco NX-OS device, the following sequence of events occurs:

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.
2. When the Cisco NX-OS 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 ifindex (for non-VXLAN VLANs) or vn-segment-id-mod-port (for VXLAN VLANs), from which the packet is received (the circuit ID suboption). In DHCP relay, the circuit ID is filled with the ifindex 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 Cisco NX-OS device if the request was relayed to the server by the device. The Cisco NX-OS device verifies that it originally inserted the Option 82 data by inspecting the remote ID and possibly the circuit ID fields. The Cisco NX-OS device removes the Option 82 field and forwards the packet to the interface that connects to the DHCP client that sent the DHCP request.

This figure shows the packet formats for the circuit ID suboption and the remote ID suboption.

Figure 8: Suboption Packet Formats



VRF Support for the DHCP Relay Agent

You can configure the DHCP relay agent to forward DHCP broadcast messages from clients in a virtual routing and forwarding (VRF) instance to DHCP servers in a different VRF. By using a single DHCP 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 *Cisco Nexus 3400-S NX-OS Unicast Routing Configuration Guide*.

Enabling VRF support for the DHCP relay agent requires that you enable Option 82 for the DHCP relay agent.

If a DHCP request arrives on an interface that you have configured with a DHCP relay address and VRF information and the address of the DCHP server belongs to a network on an interface that is a member of a different VRF, the device inserts Option 82 information in the request and forwards it to the DHCP server in the server VRF. The Option 82 information includes the following:

VPN identifier

Name of the VRF that the interface that receives the DHCP request is a member of.

Link selection

Subnet address of the interface that receives the DHCP request. When DHCP smart relay is enabled, the link selection is filled with the subnet of the active giaddr.

Server identifier override

IP address of the interface that receives the DHCP request. When DHCP smart relay is enabled, the server identifier is filled with the active giaddr.

Note The DHCP server must support the VPN identifier, link selection, and server identifier override options.

When the device receives the DHCP response message, it strips off the Option 82 information and forwards the response to the DHCP client in the client VRF.

DHCP Smart Relay Agent

When the DHCP relay agent receives broadcast DHCP request packets from a host, it sets giaddr to the primary address of the inbound interface and forwards the packets to the server. The server allocates IP addresses from the giaddr subnet pool until the pool is exhausted and ignores further requests.

You can configure the DHCP smart relay agent to allocate IP addresses from the secondary IP address subnet pool if the first subnet pool is exhausted or the server ignores further requests. This enhancement is useful if the number of hosts is greater than the number of IP addresses in the pool or if multiple subnets are configured on an interface using secondary addresses.

About the DHCPv6 Relay Agent

DHCPv6 Relay Agent

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.

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 *Cisco Nexus 3400-S NX-OS Unicast Routing Configuration Guide*.

About DHCP Client

The DHCP client feature enables the configuration of an IPv4 or IPv6 address on an interface. Interfaces can include routed ports, the management port, and switch virtual interfaces (SVIs).

Prerequisites for DHCP

DHCP has the following prerequisite:

- You should be familiar with DHCP before you configure DHCP snooping or the DHCP relay agent.

Guidelines and Limitations for DHCP

DHCP has the following configuration guidelines and limitations:

- For secure POAP, make sure that DHCP snooping is enabled and firewall rules are set to block unintended or malicious DHCP servers.
- Cisco Nexus 3400-S Series switches do not support the relaying of bootp packets. However, the switches do support bootp packets that are Layer 2 switched.
- DHCP subnet broadcast is not supported.
- You must enable the insertion of Option 82 information for DHCP packets to support the highest DHCP snooping scale.
- Before you globally enable 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 should not be followed by DHCP relay in the network (DHCP snooping does not work when the DHCP relay is configured on the same Nexus device).
- The DHCP snooping, DAI, and FHS are not supported on Cisco Nexus 3400 Series switches.
- DHCP snooping is not supported on VXLAN VLANs.
- DHCP snooping supports multiple IP addresses with the same MAC address and VLAN in static binding entries.
- VXLAN supports DHCP relay when the DHCP server is reachable through a default VRF.
- If a VLAN ACL (VACL) is configured on a VLAN that you are configuring with DHCP snooping, make sure 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, make sure that the router ACL permits DHCP traffic between DHCP servers and DHCP hosts.
- If you use DHCP relay where DHCP clients and servers are in different VRFs, use only one DHCP server within a VRF.

- Access-control list (ACL) statistics are not supported if the DHCP snooping feature is enabled.
- Make sure that the DHCP configuration is synchronized across the switches in a vPC link. Otherwise, a run-time error can occur, resulting in dropped packets.
- DHCP smart relay is limited to the first 100 IP addresses of the interface on which it is enabled.
- You must configure a helper address on the interface in order to use DHCP smart relay.
- In a vPC environment with DHCP smart relay enabled, the subnet of the primary and secondary addresses of an interface should be the same on both Cisco NX-OS devices.
- When you configure DHCPv6 server addresses on an interface, a destination interface cannot be used with global IPv6 addresses.
- The following guidelines and limitations apply to the DHCP client feature:
  - You can configure multiple SVIs, but each interface VLAN should be in a different subnet. The DHCP client feature cannot configure different IP addresses with the same subnet on different interface VLANs on the same device.
  - DHCP client and DHCP relay are not supported on the same switch.
  - DHCP client is not supported for Layer 3 subinterfaces.

Note For DHCP configuration limits, see the *Cisco Nexus 3400-S NX-OS Verified Scalability Guide*.

Default Settings for DHCP

This table lists the default settings for DHCP parameters.

Table 15: Default DHCP Parameters

Parameters	Default
DHCP feature	Disabled
DHCP relay agent	Enabled
DHCPv6 relay agent	Enabled
VRF support for the DHCP relay agent	Disabled
VRF support for the DHCPv6 relay agent	Disabled
DHCP Option 82 for relay agent	Disabled
DHCP smart relay agent	Disabled
DHCP server IP address	None

Configuring DHCP

Minimum DHCP Configuration

Procedure

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 Make sure that the DHCP server is connected to the device using a trusted interface.

Step 5 (Optional) Enable the DHCP relay agent.

Step 6 (Optional) If DHCP servers and clients are in different VRFs, do the following:

a) Enable Option 82 for the DHCP relay agent.

b) Enable VRF support for the DHCP relay agent.

Step 7 (Optional) Configure an interface with the IP address of the DHCP server.

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 the DHCP relay agent, DHCP snooping, or any of the features that depend on DHCP. In addition, all DHCP configuration is removed from the device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`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:**	Displays the DHCP configuration.

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

Configuring DHCP Snooping

Enabling or Disabling DHCP Snooping Globally

You can enable or disable DHCP snooping globally on the device.

Before you begin

Make sure that you have enabled the DHCP feature.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ip dhcp snooping** **Example:** `switch(config)# ip dhcp snooping`	Enables DHCP snooping globally. The **no** form of this command 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.

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

Make sure that the DHCP feature is enabled.

Note If a VACL is configured on a VLAN that you are configuring with DHCP snooping, make sure that the VACL permits DHCP traffic between DHCP servers and DHCP hosts.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ip dhcp snooping vlan** *vlan-list* **Example:** `switch(config)# ip dhcp snooping vlan 100,200,250-252`	Enables DHCP snooping on the VLANs specified by *vlan-list*. The **no** form of this command disables DHCP 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.

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

Make sure that the DHCP feature is enabled.

Procedure

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

	Command or Action	Purpose
	`switch# config t` `switch(config)#`	
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** form of this command 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.

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.

Note DHCP relay agent support for Option 82 is configured separately.

Note To support a higher DHCP pps scale, you must enable the insertion of Option 82 information for DHCP packets.

Note You must add Option82 as specified in the format string in the command configuration.

- The length of the Option82 string increases based on the length of the format string.
- The circuit-id must include the ascii value of the format string.

Before you begin

Make sure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`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** form of this command disables the insertion and removal of Option 82 information.

Step 3	(Optional) [**no**] **ip dhcp option82 sub-option circuit-id** *format_type string format*
	Example:
	`switch(config)# ip dhcp snooping sub-option circuit-id format-type string format`
	Example:
	`switch(config)# ip dhcp snooping sub-option circuit-id format-type string format?`
	`WORD Format string (Max Size 64)`
		Configures Option 82 as follows:
		• If you do not specify *format-type*, the *circuit-id* displays the incoming port, for example, *ethernet1/1*.
		• If you specify format <*word*>, the *circuit-id* displays the specified word
		• If you specify *%h* instead of <*word*>, the *circuit-id* displays the host name.
		• If you specify *%p* instead of <*word*>, the *circuit-id* displays the port name.
		• If you specify *%h:%p* instead of <*word*>, the *circuit-id* displays both host and port name.
		Note The *no* option disables this behavior.

Step 4	**interface** *interface slot/port*
	Example:
	`switch(config)# interface ethernet 2/2`
	`switch(config-if)#`
		Enters the interface configuration mode, where slot/port is the interface where you want to enable or disable snooping.

Step 5	(Optional) **ip dhcp option82 sub-option circuit-id**
	Example:
	`switch(config-if)# ip dhcp option82 sub-option circuit-id?`
	`WORD Format string (Max Size 64)`
	Example:
	`switch(config-if)# ip dhcp option82 sub-option circuit-id test`
	`switch(config-if)#`
		Configures Option 82 at the interface.
		Note The *no* option disables this behavior

	Command or Action	Purpose

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

Step 7	(Optional) **show ip dhcp option82 info interface** *intf_name*	Displays the DHCP configuration. It shows whether option82 is enabled or disabled on that interface and the transmitted packets for an interface that is option82 enabled.
Step 8	(Optional) **show running-config dhcp** **Example:** `switch(config)# show running-config dhcp`	Displays the DHCP 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.

Enabling or Disabling Strict DHCP Packet Validation

You can enable or disable the strict validation of DHCP packets. By default, strict validation of DHCP packets is disabled.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ip dhcp packet strict-validation** **Example:** `switch(config)# ip dhcp packet strict-validation`	Enables the strict validation of DHCP packets. The **no** form of this command disables strict DHCP packet validation.
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.

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

Make sure that the DHCP feature is enabled.

Make sure that the interface is configured as a Layer 2 interface.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`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** form of this command 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.

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

Make sure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure 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** form of this command 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.

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 subinterfaces
- Layer 3 port-channel interfaces

Before you begin

Make sure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose

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

Step 2	**interface** [**ethernet** *slot/port*[*.number*] | **port-channel** *channel-number*]
	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 or *channel-number* is the Layer 3 port-channel 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** form of this command configures the port as an untrusted interface. **Note** For any Layer 3 interface, if the interface is configured as trusted either through a global command or an interface-level command, the interface is considered as a trusted interface. Hence, when the trusted-port command is enabled at the global level, any Layer 3 interface cannot be considered as untrusted irrespective of the interface-level configuration.
Step 4	(Optional) **show ip dhcp relay information trusted-sources** **Example:** `switch(config-if)# show ip dhcp relay information trusted-sources`	Displays the DHCP relay trusted ports configuration.
Step 5	(Optional) **show running-config dhcp** **Example:** `switch(config-if)# show running-config dhcp`	Displays the DHCP configuration.

	Command or Action	Purpose
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.

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 subinterfaces
- Layer 3 port-channel interfaces

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

Make sure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`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** form of this command configures the ports as untrusted interfaces.
Step 3	(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 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:**	Copies the running configuration to the startup configuration.

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

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.

Procedure

	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 agent. The **no** option disables the DHCP 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.

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.

Before you begin

Ensure that the DHCP feature is enabled.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **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 3	(Optional) switch(config)# [**no**] **ip dhcp relay sub-option circuit-id customized**	Programs Option 82 with the VLAN + slot + port format. This command is applicable only for SVIs. The **no** option disables this behavior.
Step 4	(Optional) switch(config)# [**no**] **ip dhcp relay sub-option circuit-id format-type string**	Configures Option 82 to use encoded string format instead of the default binary ifindex format. The **no** option disables this behavior. For VLANs and SVIs: • When this command and the **ip dhcp relay sub-option circuit-id customized** command are both configured, the **ip dhcp relay sub-option circuit-id format-type string** command is programmed. • When the **ip dhcp relay sub-option circuit-id format-type string** command is removed, the **ip dhcp relay sub-option circuit-id customized** command is programmed. • When both commands are removed, the ifindex is programmed. For other interfaces, if the **ip dhcp relay sub-option circuit-id format-type string** command is configured, it is used. Otherwise, the default ifindex is programmed.
Step 5	(Optional) switch(config)# **show ip dhcp relay**	Displays the DHCP relay configuration.
Step 6	(Optional) switch(config)# **show running-config dhcp**	Displays the DHCP configuration.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

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.

Procedure

	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 cisco** **Example:** `switch(config)# ipv6 dhcp relay option type cisco`	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.

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 than the interface, ensure that you have enabled VRF support.

Note If an ingress router ACL is configured on an interface that you are configuring with a DHCP server address, ensure that the router ACL permits DHCP traffic between DHCP servers and DHCP hosts.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	Example:
	`switch# configure terminal`
	`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* [**use-vrf** *vrf-name*]	Configures an IP address for a DHCP server to which the relay agent forwards

	Command or Action	Purpose
	Example: `switch(config-if)# ip dhcp relay address 10.132.7.120 use-vrf red`	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 startup-config`	Copies the running configuration to the startup configuration.

Configuring the DHCP Relay Source Interface

You can configure the source interface for the DHCP relay agent. By default, the DHCP 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 DHCP relay agent is enabled.

Procedure

	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* **Example:**	Configures the source interface for the DHCP relay agent.

	Command or Action	Purpose
	`switch(config)# ip dhcp relay source-interface loopback 2`	**Note** The DHCP relay source interface can be configured globally, per interface, or both. When both the global and interface levels are configured, the interface-level configuration overrides the global configuration.
Step 3	(Optional) **show ip dhcp relay** [**interface** *interface*] **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.

Enabling or Disabling DHCP Smart Relay Globally

You can enable or disable DHCP smart relay globally on the device.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCP relay agent is enabled.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **ip dhcp smart-relay global** **Example:** `switch(config)# ip dhcp smart-relay global`	Enables DHCP smart relay globally. The **no** form of this command disables DHCP smart relay.
Step 3	(Optional) **show ip dhcp relay** **Example:**	Displays the DHCP smart relay configuration.

	Command or Action	Purpose
	`switch(config)# show ip dhcp relay`	
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 DHCP Smart Relay on a Layer 3 Interface

You can enable or disable DHCP smart relay on Layer 3 interfaces.

Before you begin

Ensure that the DHCP feature is enabled.

Ensure that the DHCP relay agent is enabled.

Procedure

	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 2/3`
	`switch(config-if)#`
		Enters interface configuration mode, where *slot*/*port* is the interface for which you want to enable or disable DHCP smart relay.

Step 3	[**no**] **ip dhcp smart-relay** **Example:** `switch(config-if)# ip dhcp smart-relay`	Enables DHCP smart relay on the interface. The **no** form of this command disables DHCP smart relay on the interface.

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

Step 5	**exit** **Example:**	Exits global configuration mode.

	Command or Action	Purpose
	`switch(config)# exit` `switch#`	
Step 6	(Optional) **show ip dhcp relay** **Example:** `switch# show ip dhcp relay`	Displays the DHCP smart relay configuration.
Step 7	(Optional) **show running-config dhcp** **Example:** `switch# show running-config dhcp`	Displays the DHCP configuration.
Step 8	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring DHCPv6

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.

Procedure

	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.

	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 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.

Procedure

	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 cisco** **Example:** `switch(config)# ipv6 dhcp relay option type cisco`	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:**	Displays the DHCPv6 relay configuration.

	Command or Action	Purpose
	`switch(config)# show ipv6 dhcp relay`	
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.

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.

Note If an ingress router ACL is configured on an interface that you are configuring with a DHCPv6 server address, ensure that the router ACL permits DHCP traffic between DHCPv6 servers and DHCP hosts.

Procedure

	Command or Action	Purpose

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

Step 2	Do one of the following options:
	• **interface ethernet** *slot*/*port*
	• **interface port-channel** *channel-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.
		• Enters interface configuration mode, where *channel-id* is the ID of the port channel

	Command or Action	Purpose
		that you want to configure with a DHCPv6 server IP address.
Step 3	[**no**] **ipv6 dhcp relay address** *IPv6-address* [**use-vrf** *vrf-name*] [**interface** *interface*] **Example:** `switch(config-if)# ipv6 dhcp relay address FF02:1::FF0E:8C6C use-vrf red`	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 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.

Configuring the DHCP Relay Source Interface

You can configure the source interface for the DHCP relay agent. By default, the DHCP 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 DHCP relay agent is enabled.

Procedure

	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* **Example:** `switch(config)# ip dhcp relay source-interface loopback 2`	Configures the source interface for the DHCP relay agent. **Note** The DHCP relay source interface can be configured globally, per interface, or both. When both the global and interface levels are configured, the interface-level configuration overrides the global configuration.
Step 3	(Optional) **show ip dhcp relay** [**interface** *interface*] **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.

Configuring IPv6 RA Guard

You can configure the IPv6 router advertisement (RA) guard feature for Cisco Nexus 3400-S Series switches. This feature is used to drop all incoming IPv6 RA packets on a Layer 2 interface.

Before you begin

You must enable DHCP (using the **feature dhcp** command).

To enable DHCP relay on any interface, you must disable DHCP on interfaces that have an IPv4 or IPv6 address assigned using DHCP (dynamic IP addressing).

Make sure that both PTP (**feature ptp**) and NV overlay (**feature nv overlay**) are not already configured. A dynamic ifacl label is reserved when these features are configured. However, only two dynamic ifacl label bits are available. If both of these features are already configured, a dynamic ifacl label will not be available for IPv6 RA guard, and the feature cannot be enabled.

Procedure

	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)#`
		Enters interface configuration mode.

Step 3	[**no**] **ipv6 nd raguard** **Example:** `switch(config-if)# ipv6 nd raguard`	Enables the IPv6 RA guard feature on the specified interface.
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.

Enabling DHCP Client

You can use the DHCP client feature to enable the configuration of an IPv4 or IPv6 address on an interface. Interfaces can include routed ports, the management port, and switch virtual interfaces (SVIs). Layer 3 subinterfaces are not supported.

Note DHCP client is independent of the DHCP relay and DHCP snooping processes, so it does not require that the **feature dhcp** command be enabled.

Procedure

	Command or Action	Purpose

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

Step 2	Do one of the following options: • **interface ethernet** *slot*/*port* • **interface mgmt 0** • **interface vlan** *vlan-id*	• Enters interface configuration mode, where *slot*/*port* is the physical Ethernet interface for which you want to enable the DHCP client feature.

	Command or Action	Purpose

	Example:
	`switch(config)# interface vlan 3`
	`switch(config-if)#`
		• Enters interface configuration mode and specifies the management interface as the interface for which you want to enable the DHCP client feature.
		• Enters interface configuration mode, where *vlan-id* is the ID of the VLAN for which you want to enable the DHCP client feature.

Step 3	**ipv6 address use-link-local-only** **Example:** `switch(config-if)# ipv6 address use-link-local-only`	You must enter this command before assigning an IPv6 address to the interface in the next step. This command is not required if you will assign an IPv4 address to the interface.
Step 4	[**no**] {**ip** | **ipv6**} **address dhcp** **Example:** `switch(config-if)# ip address dhcp`	Assigns an IPv4 or IPv6 address to the interface. The **no** form of this command releases the IP address.
Step 5	(Optional) Do one of the following options: • **show running-config interface ethernet** *slot*/*port* • **show running-config interface mgmt 0** • **show running-config interface vlan** *vlan-id* **Example:** `switch(config-if)# show running-config interface vlan 3`	Displays the IPv4 or IPv6 address assigned to the interface in the running 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. Only the {**ip** | **ipv6**} **address dhcp** command is saved. The assigned IP address is not saved even though it shows in the running configuration.

Verifying the DHCP Configuration

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

Command	Purpose
show ip dhcp relay	Displays the DHCP relay configuration.

Command	Purpose
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 running-config dhcp [**all**]	Displays the DHCP configuration in the running configuration.
show startup-config dhcp [**all**]	Displays the DHCP configuration in the startup configuration.

Displaying IPv6 RA Guard Statistics

To display IPv6 RA guard statistics, perform one of the following tasks:

Command	Purpose
show ipv6 raguard statistics	Displays IPv6-related RA guard statistics.

The following example shows sample statistics:

```
switch# show ipv6 raguard statistics
-------------------------------------
Interface     Rx         Drops
------------- ---------- ------------
Ethernet1/53  4561102    4561102
```

Displaying DHCP Snooping Bindings

Use the **show ip dhcp snooping binding** [*ip-address* | *mac-address* | **dynamic** | **static** | **vlan** *vlan-id* | **interface** *interface-type interface-number*] command to display all entries from the DHCP snooping binding database.

```
MacAddress        IpAddress LeaseSec Type   VLAN Interface
----------------- --------- -------- ------ ---- ------------
0f:00:60:b3:23:33 10.3.2.2  infinite static 13   Ethernet2/46
0f:00:60:b3:23:35 10.2.2.2  infinite static 100  Ethernet2/10
```

Clearing the DHCP Snooping Binding Database

Use the **clear ip dhcp snooping binding** command to clear all entries from the DHCP snooping binding database.

Use the **clear ip dhcp snooping binding interface ethernet** *slot/port* command to clear entries associated with a specific Ethernet interface from the DHCP snooping binding database.

Use the **clear ip dhcp snooping binding interface port-channel** *channel-number* command to clear entries associated with a specific port-channel interface from the DHCP snooping binding database.

Use the **clear ip dhcp snooping binding vlan** *vlan-id* [**mac** *mac-address* **ip** *ip-address* **interface** {**ethernet** *slot /port* | **port-channel** *channel-number*}] command to clear a single specific VLAN entry from the DHCP snooping binding database.

Monitoring DHCP

Use the **show ip dhcp relay statistics** [**interface** *interface*] command to monitor DHCP relay statistics at the global or interface level.

Use the **show ipv6 dhcp relay statistics** [**interface** *interface*] command to monitor DHCPv6 relay statistics at the global or interface level.

Clearing DHCP Snooping Statistics

Use the **clear ip dhcp snooping statistics** [**vlan** *vlan-id*] command to clear the DHCP snooping statistics.

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.

Use the **clear ip dhcp global statistics** command to clear the DHCP statistics globally.

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.

Configuration Examples for DHCP

This example shows how to enable the DHCP relay agent and configure the DHCP server IP address for Ethernet interface 2/3, where the DHCP server IP address is 10.132.7.120 and the DHCP server is in the VRF instance named red:

```
feature dhcp
ip dhcp relay
ip dhcp relay information option
ip dhcp relay information option vpn

interface ethernet 2/3
  ip dhcp relay address 10.132.7.120 use-vrf red
```

This example shows how to enable and use the DHCP smart relay agent. In this example, the device forwards the DHCP broadcast packets received on Ethernet interface 2/2 to the DHCP server (10.55.11.3), inserting

192.168.100.1 in the giaddr field. If the DHCP server has a pool configured for the 192.168.100.0/24 network, it responds. If the server does not respond, the device sends two more requests using 192.168.100.1 in the giaddr field. If the device still does not receive a response, it starts using 172.16.31.254 in the giaddr field instead.

```
feature dhcp
ip dhcp relay
ip dhcp smart-relay global

interface ethernet 2/2
  ip address 192.168.100.1/24
  ip address 172.16.31.254/24 secondary
  ip dhcp relay address 10.55.11.3
```

Configuration Examples for DHCP Client

The following example shows how the DHCP client feature can be used to assign an IPv4 address to a VLAN interface:

```
switch# configure terminal
switch(config)# interface vlan 7
switch(config-if)# no shutdown
switch(config-if)# ip address dhcp
switch(config-if)# show running-config interface vlan 7
interface Vlan7
no shutdown
ip address dhcp
```

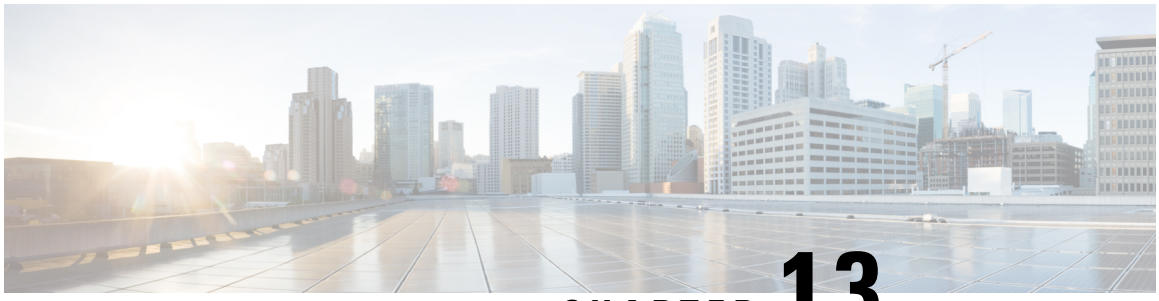
Additional References for DHCP

Related Documents

Related Topic	Document Title
vPCs	*Cisco Nexus 3400-S NX-OS Interfaces Configuration Guide*
VRFs and Layer 3 virtualization	*Cisco Nexus 3400-S NX-OS Unicast Routing Configuration Guide*

Standards

Standards	Title
RFC-2131	Dynamic Host Configuration Protocol (https://tools.ietf.org/html/rfc2131)
RFC-3046	DHCP Relay Agent Information Option (https://tools.ietf.org/html/rfc3046)
RFC-6607	Virtual Subnet Selection Options for DHCPv4 and DHCPv6 (https://tools.ietf.org/html/rfc6607)



CHAPTER 13

Configuring IPv6 First Hop Security

This chapter describes how to configure First Hop Security (FHS) features on Cisco NX-OS devices.

This chapter includes the following sections:

- Introduction to First-Hop Security, on page 237
- Guidelines and Limitations of First Hop Security, on page 238
- About vPC First Hop Security Configuration, on page 238
- RA Guard, on page 241
- DHCPv6 Guard, on page 242
- IPv6 Snooping, on page 243
- How to Configure IPv6 FHS, on page 244
- Configuration Examples, on page 252
- Additional References for IPv6 First-Hop Security, on page 253

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, and help with scale in large L2 domains. These features provide protection from a wide host of rogue or mis-configured users. You can use extended FHS features for different deployment scenarios, or attack vectors.

The following FHS features are supported:

- IPv6 RA Guard
- DHCPv6 Guard
- IPv6 Snooping

Note Use the **feature dhcp** command to enable the FHS features on a switch.

IPv6 Global Policies

IPv6 global policies provide storage and access policy database services. IPv6 snooping, DHCPv6 guard, and IPv6 RA guard are IPv6 global policies features. Each time IPv6 snooping, DHCPv6 guard, 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.

All port level FHS policies are programmed in the ifacl region, while the VLAN level policies are programmed in the FHS region. Use the hardware profile **tcam regionfhs tcam_size** command to configure the FHS. The range for the TCAM size is 0-4096.

IPv6 First-Hop Security Binding Table

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 IPv6 address, and prefix binding of the neighbors to prevent spoofing and redirect attacks.

Guidelines and Limitations of First Hop Security

The general guidelines and limitations of First Hop Security are as follows:

- Before enabling the FHS on the interface or VLAN, we recommend carving TCAM regions on Cisco Nexus 3400-SSeries switches. To enable FHS successfully:
  - On an interface, you must carve the **ifacl** TCAM region.
  - On a VLAN, you must carve the necessary redirect TCAM region.
  - On a FEX interface, you must carve the **fex-ipv6-ifacl** TCAM region.
  - On a
- Before enabling the FHS, we recommend carving the **ing-redirect** TCAM region on Cisco Nexus 3400-S Series switches.

About vPC First Hop Security Configuration

You can deploy IPv6 First Hop Security vPC in many ways. We recommend the following best practice deployment scenarios:

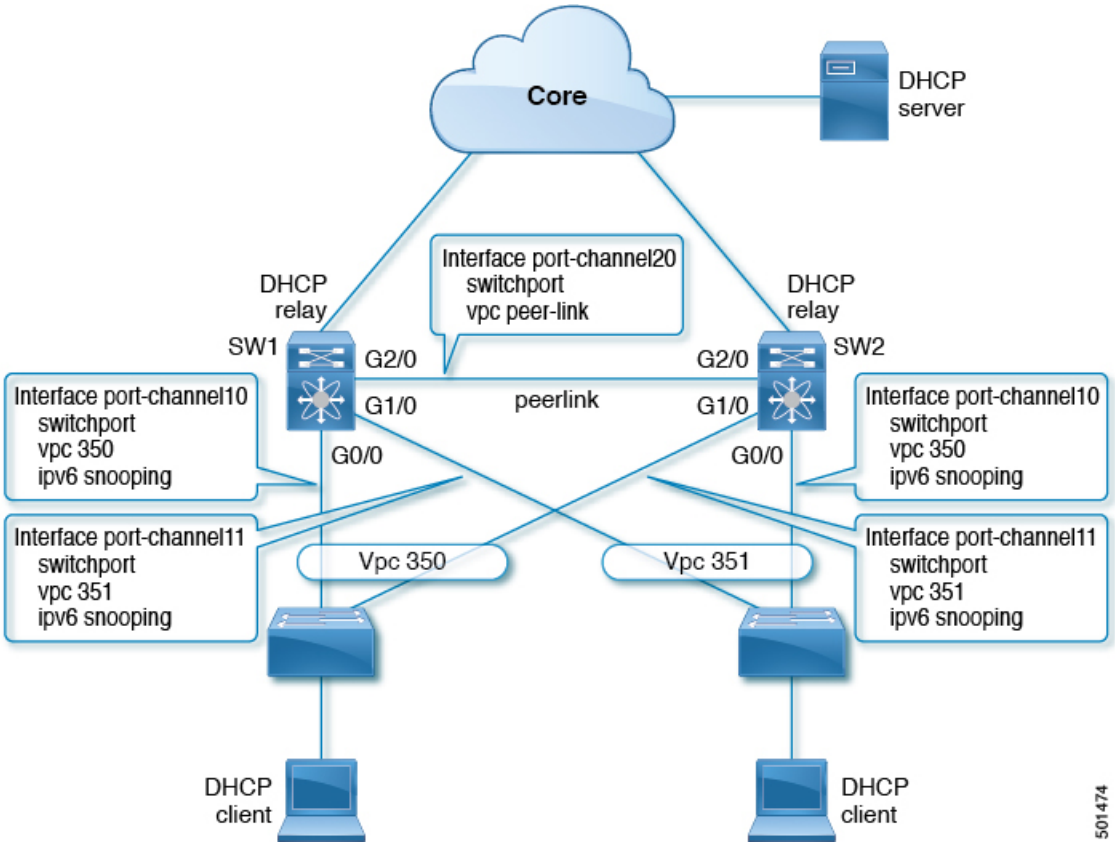
- DHCP relay on-stack
- DHCP relay on vPC leg
- DHCP client and relay on orphan ports

DHCP Relay On-stack

In this deployment scenario, you can directly connect clients behind the vPC link, or behind an intermediary switch with DHCP relay running on the Nexus switch. Connecting clients behind an intermediary switch with DHCP relay running on the Nexus switch, is ideal because you can configure the IPv6 Snooping feature on the vPC interface links directly, instead of at a VLAN level. Configuration at the interface level is efficient for the following reasons:

- Control traffic (DHCP/ND) will not be redirected to CPU for processing on both vPC peers if it goes over the peer link.
- Packets switched over the peer link aren't processed a second time.

Figure 9: FHS Configuration with DHCP relay on-stack



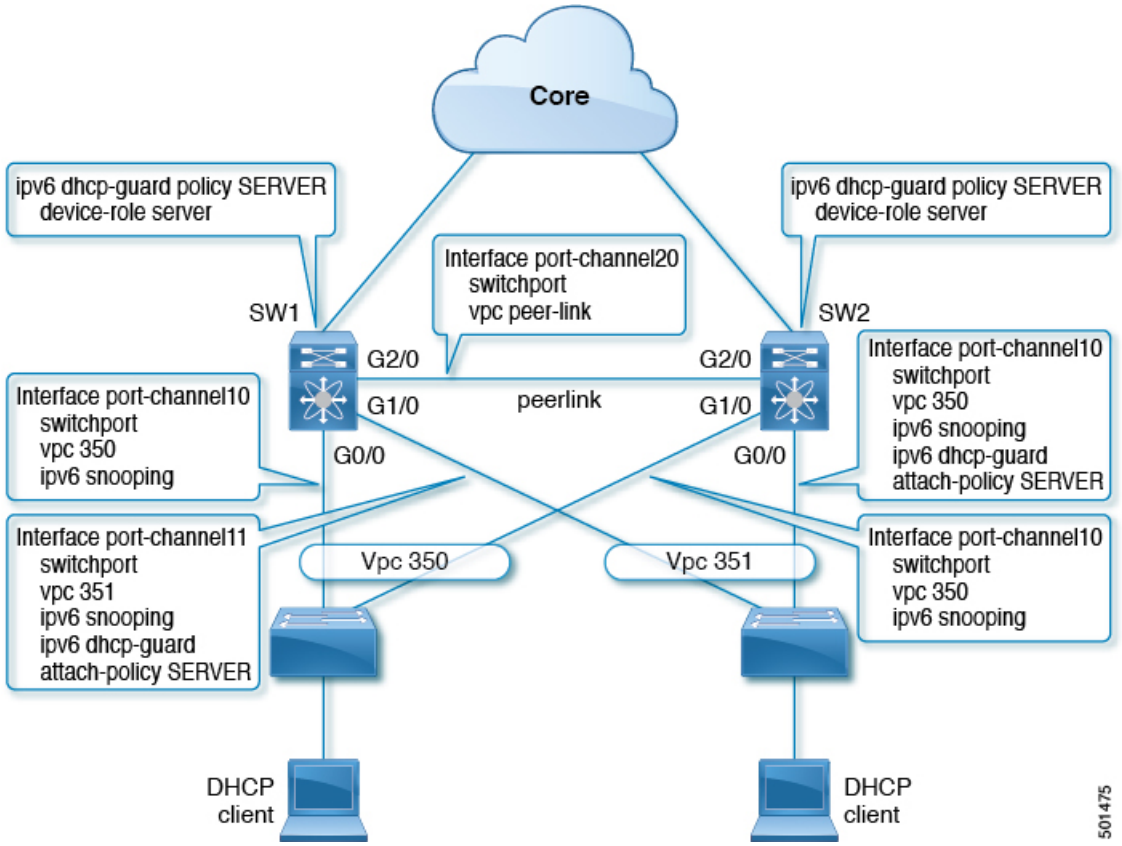
In the figure, snooping policy is enabled on both vPC links. In this scenario, the two vPC peers learn all the host IP/MAC bindings behind the vPC links and sync these up between themselves. The two vPC peers learn the bindings using both IPv6 ND and IPv6 DHCP control protocols.

DHCP Relay on VPC Leg

In this configuration, the relay agent does not run on the vPC peers. Instead, the DHCP relay agent (or a DHCP server) is runs behind a vPC link (it can be towards the access, or even somewhere in the core). In such a deployment scenario, the IPv6 Snooping feature doesn't implicitly trust the DHCP Server messages and drops DHCP Server messages by default. You can customize the IPv6 policy to implement:

- Security-level glean.
- IPv6 DHCP Guard policy with device-role server. In this configuration, IPv6 Snooping trusts DHCP server messages attached to the vPC link.

Figure 10: FHS Configuration with external DHCP relay



In the figure, the clients are located behind the vPC links with the default IPv6 snooping policy. You can attach both ipv6 snooping and ipv6 dhcp-guard attach-policy SERVER policies to the links where DHCP server traffic arrives. You will need both the server or relay facing and client facing IPv6 snooping policies to create the client binding entries via DHCP control traffic. This is because IPv6 Snooping needs to see both the client and server packets to create the binding. You must also configure the IPv6 DHCP Guard policy to allow DHCP server traffic by the IPv6 Snooping policy. Both peers require the same configuration because the vPC peers synch all newly learnt client entries learnt on the vPC port.

DHCP Client Relay on Orphan Ports

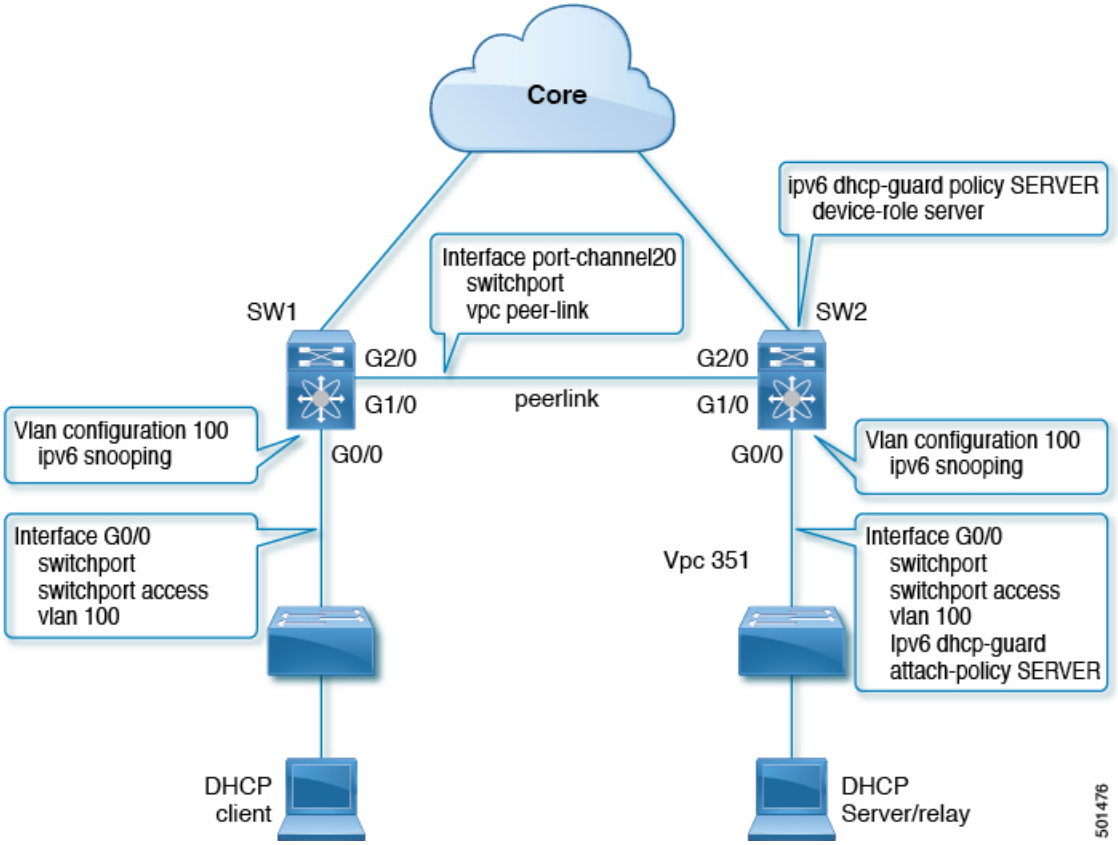
In this configuration, you can connect the client via an orphan port. The IPv6 Snooping feature only syncs client bindings on vPC ports, but not on orphan ports as these are not directly connected to both vPC peers. In such a configuration, the IPv6 Snooping feature runs independently on both switches. The figure illustrates the following:

- On the first switch, you must attach the IPv6 Snooping policy on the client facing interface. However, to accommodate DHCP server packets coming from the server on an orphan port behind the vPC peer, you must attach the policy at the VLAN level. In such a case, the policy applied at the VLAN inspects

both the client traffic interface and DHCP server traffic. You do not require an individual IPv6 snooping policy per interface. Any DHCP traffic arriving via the vPC peer is also implicitly trusted and if policing is required, the vPC peer automatically drops it.

- You must also configure IPv6 on the second switch at the VLAN level. You must also configure the IPv6 DHCP Guard policy with a "device-role server" on the server facing orphan port. This prevents the IPv6 Snooping feature from dropping the DHCP server packets. Both switches learn the client binding entries individually and will not sync them, because the client is not on a vPC link.

Figure 11: FHS configuration with client and DHCP relay on orphan port



RA Guard

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.

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 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.

DHCPv6 Guard

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. 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 DHCP server advertisements occurs for server preference checking.

If the device is configured as a DHCP server, all the messages need to be switched, regardless of the device role configuration.

Limitation of DHCPv6 Guard

The guidelines and limitations of DHCPv6 Guard are as follows:

- If a packet arriving from DHCP server is a Relay Forward or a Relay Reply, only the device role is checked. In addition, IPv6 DHCP Guard doesn't apply the policy for a packet sent out by the local relay agent running on the switch.

IPv6 Snooping

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, Neighbor Discovery Protocol (NDP) messages are directed to SISF. For DHCPv6, UDP messages sourced from dhcvp6_client and dhcvp_server ports 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.

Guidelines and Limitations for IPv6 Snooping

The guidelines and limitations of IPv6 Snooping are as follows:

- You must perform the same configurations on both the vPC peers. Automatic consistency checker for IPv6 snooping is not supported.
- The IPv6 Snooping feature is supported only in hardware when the ternary content addressable memory (TCAM) is programed.
- This feature can be configured on a switch port interface or VLAN only on the ingress port.
- For IPv6 Snooping to learn DHCP bindings, it must see both server and client replies. A IPv6 snooping policy must be attached to both the client facing the interface (or VLAN) as well as the DHCP server facing interface (or VLAN). In the case of DHCP Relay, an IPv6 Snooping policy must be attached at the VLAN level to see the server replies.

How to Configure IPv6 FHS

Configuring the IPv6 RA Guard Policy on the Device

Note When the **ipv6 nd raguard** command is configured on ports, router solicitation messages are not replicated to these ports. To replicate router solicitation messages, all ports that face routers must be set to the router role.

Procedure

	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. • device-role host—Interface or VLAN where you connect a regular node or host. This where you apply the IPV6 RA Guard policy. The device-role host allows incoming RS packets, and blocks incoming RA or RR packets. RS packets that are received on another interface, are not redirected to the device-role host. Only RA and RR packets (that are allowed) are redirected to the device-role host. • device-role switch—The device-role switch behaves similar to the device-role host. For example, you can use it as a label for a trunk port. • device-role monitor—This device monitors network traffic. It behaves similar to the device-role host, except that RS packets are also sent to this interface. This helps capture traffic.

	Command or Action	Purpose
		• device-role router—Interface that connects to the router. This interface allows incoming RS, RA, or RR packets.
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.

Configuring IPv6 RA Guard on an Interface

Procedure

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

	Command or Action	Purpose

Step 2	**interface** *type number*
	Example:
	`Device(config)# interface ethernet 1/1`
	Example:
	`Device(config)# vlan configuration 10`
		Specifies an interface type and number, and places the device in interface or VLAN 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.

Configuring DHCP—DHCPv6 Guard

Procedure

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

	Command or Action	Purpose
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). • device-role client—Interface where a normal DHCPv6 client is connected. It blocks any incoming server packets. • device-role server—Interface where a normal DHCPv6 server is connected. It allows all DHCPv6 packets originating on this interface.
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.
Step 9	**switchport** **Example:**	Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.

	Command or Action	Purpose
	`Device(config-if)# switchport`	
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.

Configuring IPv6 Snooping

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ipv6 snooping policy** *policy-name* **Example:** `Device(config)# ipv6 snooping policy policy1`	Configures an IPv6 snooping policy and enters IPv6 snooping configuration mode.
Step 3	**device-role** { **node** | **switch** } **Example:** `Device(config-snoop-policy)# device-node switch`	Specifies the role of the device attached to the target (interface or VLAN): • node—is the default. Bindings are created and entries are probed. • switch—Entries are not probed and when a trusted port is enabled, bindings are not created.
Step 4	**[no] limit address-count** **Example:** `Device(config-snoop-policy)# limit address-count 500`	Limits the number of binding entries, a no limit address-count means no limit.

Step 5	**[no] protocol** *dhcp* | *ndp*
	Example:
	`Device(config-snoop-policy)# protocol dhcp`
	`Device(config-snoop-policy)# protocol ndp`
		Turns on or switches off either DHCP or NDP gleaning.

Step 6	**trusted-port** **Example:** `Device(config-snoop-policy)# trusted-port`	Specifies that the policy be applied to a trusted port. If an entry is a trusted-port, none of it's traffic will be blocked or dropped.
Step 7	**security-level** *glean* | *guard* | *inspect* **Example:** `Device(config-snoop-policy)# security-level guard`	Specifies the type of security applied to the policy: glean, guard, or inspect. Here is what each security level means: • glean—learns bindings but does not drop packets.

	Command or Action	Purpose
		• inspect—learns bindings and drops packets in case it detects an issue, such as address theft. • guard—works like inspect, but in addition drops IPv6, ND, RA, and IPv6 DHCP Server packets in case of a threat.
Step 8	**tracking** **Example:** `Device(config-snoop-policy)# tracking enable`	Enables tracking.
Step 9	**exit** **Example:** `Device(config-snoop-policy)# exit`	Exits snooping configuration mode and returns to global configuration mode.
Step 10	**interface** *type-number* **Example:** `Device(config-if)# interface ethernet 1/25`	Specifies an interface and enters interface configuration mode.
Step 11	**[no] switchport** **Example:** `Device(config-if)# switchport`	Switches between Layer 2 and Layer 3 mode.
Step 12	**ipv6 snooping attach-policy** *policy-name* **Example:** `Device(config-if)# ipv6 snooping attach-policy policy1`	Attaches the IPv6 snooping policy to an interface.
Step 13	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 14	**vlan configuration** *vlan-id* **Example:** `Device(config)# vlan configuration 333`	Specifies a VLAN and enters VLAN configuration mode.
Step 15	**ipv6 snooping attach-policy** *policy-name* **Example:** `Device(config-vlan-config)# ipv6 snooping attach-policy policy1`	Attaches the IPv6 snooping policy to a VLAN.
Step 16	**exit** **Example:** `Device(config-vlan-config)# exit`	Exits VLAN configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 17	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 18	**show ipv6 snooping policy** *policy-name* **Example:** `Device(config)# show ipv6 snooping policy policy1`	Displays the policy configuration and the interfaces where the policy is applied.

Verifying and Troubleshooting IPv6 Snooping

Procedure

	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*] **Example:** `Device# show ipv6 snooping policies`	Displays information about the configured policies and the interfaces to which they are attached.
Step 5	**debug ipv6 snooping** **Example:** `Device# debug ipv6 snooping`	Enables debugging for snooping information in IPv6.

Configuration Examples

Example: IPv6 RA Guard Configuration

```
Device(config)# interface ethernet 1/1

Device(config-if)# ipv6 nd raguard attach-policy

Device# show running-config interface ethernet 1/1

Building configuration...
Current configuration : 129 bytes
!
interface ethernet1/1
 switchport
 switchport access vlan 222
 switchport mode access
 access-group mode prefer port
 ipv6 nd raguard
end
```

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
```

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
```

Example: Configuring IPv6 Snooping

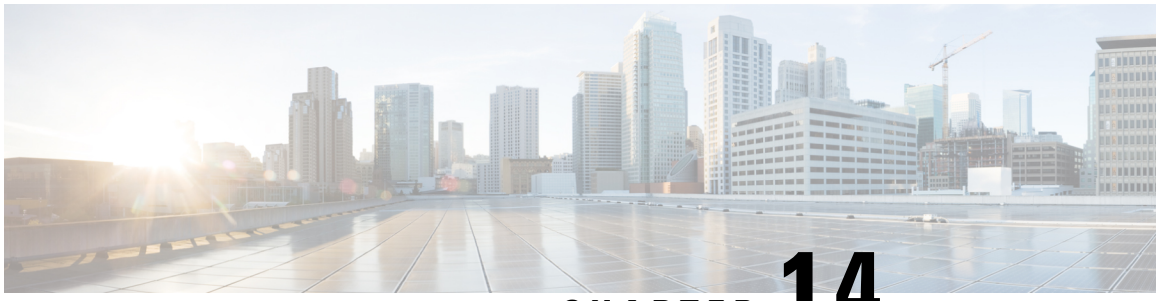
```
switch (config)# ipv6 snooping policy policy1
switch(config-ipv6-snooping)# ipv6 snooping attach-policy policy1
switch(config-ipv6-snooping)# exit
.
.
.
Device# show ipv6 snooping policies policy1
Policy policy1 configuration:
  trusted-port
  device-role node
Policy applied on the following interfaces:
   Et0/0       vlan all
   Et1/0       vlan all
Policy applied on the following vlans:
   vlan 1-100,200,300-400
```

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
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 3400-S NX-OS Security Command Reference*



CHAPTER 14

Configuring Password Encryption

This chapter describes how to configure password encryption on Cisco NX-OS devices.

This chapter includes the following sections:

- About AES Password Encryption and Primary Encryption Keys, on page 255
- Guidelines and Limitations for Password Encryption, on page 255
- Default Settings for Password Encryption, on page 256
- Configuring Password Encryption, on page 256
- Verifying the Password Encryption Configuration, on page 258
- Configuration Examples for Password Encryption, on page 258

About AES Password Encryption and Primary Encryption Keys

You can enable strong, reversible 128-bit Advanced Encryption Standard (AES) password encryption, also known as type-6 encryption. To start using type-6 encryption, you must enable the AES password encryption feature and configure a primary encryption key, which is used to encrypt and decrypt passwords.

After you enable AES password encryption and configure a primary key, all existing and newly created clear-text passwords for supported applications (currently RADIUS and TACACS+) are stored in type-6 encrypted format, unless you disable type-6 password encryption. You can also configure Cisco NX-OS to convert all existing weakly encrypted passwords to type-6 encrypted passwords.

Guidelines and Limitations for Password Encryption

Password encryption has the following configuration guidelines and limitations:

- Only users with administrator privilege (network-admin) can configure the AES password encryption feature, associated encryption and decryption commands, and primary keys.
- RADIUS and TACACS+ are the only applications that can use the AES password encryption feature.
- Configurations containing type-6 encrypted passwords are not rollback compliant.
- You can enable the AES password encryption feature without a primary key, but encryption starts only when a primary key is present in the system.
- Deleting the primary key stops type-6 encryption and causes all existing type-6 encrypted passwords to become unusable, unless the same primary key is reconfigured.

- To move the device configuration to another device, either decrypt the configuration before porting it to the other device or configure the same primary key on the device to which the configuration will be applied.

Default Settings for Password Encryption

This table lists the default settings for password encryption parameters.

Table 16: Default Password Encryption Parameter Settings

Parameters	Default
AES password encryption feature	Disabled
Primary key	Not configured

Configuring Password Encryption

This section describes the tasks for configuring password encryption on Cisco NX-OS devices.

Configuring a Primary Key and Enabling the AES Password Encryption Feature

You can configure a pimary key for type-6 encryption and enable the Advanced Encryption Standard (AES) password encryption feature.

Procedure

	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 primary key to be used with the AES password encryption feature. The primary key can contain between 16 and 32 alphanumeric characters. You can use the **no** form of this command to delete the primary key at any time.
		If you enable the AES password encryption feature before configuring a primary key, a message appears stating that password encryption will not take place unless a primary key is configured. If a primary key is already configured, you are prompted to enter the current primary key before entering a new primary key.

Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`switch# configure terminal` `switch(config)#`	
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 primary 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 primary key in the running configuration and the startup configuration.

Converting Existing Passwords to Type-6 Encrypted Passwords

You can convert existing plain or weakly encrypted passwords to type-6 encrypted passwords.

Before you begin

Ensure that you have enabled the AES password encryption feature and configured a primary key.

Procedure

	Command or Action	Purpose
Step 1	**encryption re-encrypt obfuscated** **Example:** `switch# encryption re-encrypt obfuscated`	Converts existing plain or weakly encrypted passwords to type-6 encrypted passwords.

Converting Type-6 Encrypted Passwords Back to Their Original States

You can convert type-6 encrypted passwords back to their original states.

Before you begin

Ensure that you have configured a primary key.

Procedure

	Command or Action	Purpose

Step 1	**encryption decrypt type6**
	Example:
	`switch# encryption decrypt type6`
	`Please enter current Master Key:`
		Converts type-6 encrypted passwords back to their original states.

Deleting Type-6 Encrypted Passwords

You can delete all type-6 encrypted passwords from the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**encryption delete type6** **Example:** `switch# encryption delete type6`	Deletes all type-6 encrypted passwords.

Verifying the Password Encryption Configuration

To display password encryption configuration information, perform the following task:

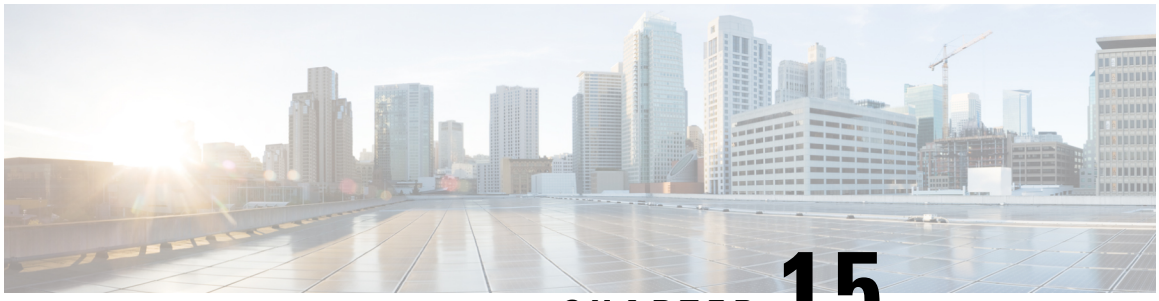
Command	Purpose
show encryption service stat	Displays the configuration status of the AES password encryption feature and the primary key.

Configuration Examples for Password Encryption

The following example shows how to create a primary key, enable the AES password encryption feature, and configure a type-6 encrypted password for a TACACS+ application:

```
key config-key ascii
  New Master Key:
  Retype Master Key:
configure terminal
feature password encryption aes
show encryption service stat
  Encryption service is enabled.
  Master Encryption Key is configured.
  Type-6 encryption is being used.
feature tacacs+
tacacs-server key Cisco123
show running-config tacacs+
  feature tacacs+
  logging level tacacs 5
```

```
  tacacs-server key 6
"JDYkqyIFWeBvzpljSfWmRZrmRSRE8syxKlOSjP9RCCkFinZbJI3GD5c6rckJR/Qju2PKLmOewbheAA=="
```



CHAPTER 15

Configuring Keychain Management

This chapter describes how to configure keychain management on a Cisco NX-OS device.

This chapter includes the following sections:

- About Keychain Management, on page 261
- Prerequisites for Keychain Management, on page 262
- Guidelines and Limitations for Keychain Management, on page 262
- Default Settings for Keychain Management, on page 262
- Configuring Keychain Management, on page 263
- Determining Active Key Lifetimes, on page 269
- Verifying the Keychain Management Configuration, on page 269
- Determining Active Key Lifetimes, on page 270
- Verifying the Keychain Management Configuration, on page 270
- Additional References for Keychain Management, on page 270

About 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 *Cisco Nexus 3400-S NX-OS Unicast Routing Configuration Guide*.

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.

Prerequisites for Keychain Management

Keychain management has no prerequisites.

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.

Default Settings for Keychain Management

This table lists the default settings for Cisco NX-OS keychain management parameters.

Table 17: 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.

Configuring Keychain Management

Creating a Keychain

You can create a keychain on the device. A new keychain contains no keys.

Procedure

	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 bgp-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 bgp-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.

Removing a Keychain

You can remove a keychain on the device.

Note Removing a keychain removes any keys within the keychain.

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.

Procedure

	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 bgp-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 bgp-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.

Configuring a Primary Key and Enabling the AES Password Encryption Feature

You can configure a pimary key for type-6 encryption and enable the Advanced Encryption Standard (AES) password encryption feature.

Procedure

	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 primary key to be used with the AES password encryption feature. The primary key can contain between 16 and 32 alphanumeric characters. You can use the **no** form of this command to delete the primary key at any time.
		If you enable the AES password encryption feature before configuring a primary key, a message appears stating that password encryption will not take place unless a primary key is configured. If a primary key is already configured, you are prompted to enter the current primary key before entering a new primary key.

	Command or Action	Purpose

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 primary 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 primary key in the running configuration and the startup configuration.

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 Cisco NX-OS 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.

Procedure

	Command or Action	Purpose

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

Step 2	**key chain** *name* **Example:**	Enters keychain configuration mode for the keychain that you specified.

	Command or Action	Purpose
	`switch(config)# key chain bgp-keys` `switch(config-keychain)#`	

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 Cisco 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 Cisco NX-OS device.
Step 5	(Optional) **show key chain** *name* [**mode decrypt**] **Example:** `switch(config-keychain-key)# show key chain bgp-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.

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.

Note We recommend that you configure the keys in a keychain to have overlapping lifetimes. This practice prevents loss of key-secured communication due to moments where no key is active.

Procedure

	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 bgp-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 2013 23:59:59 Sep 12 2013`	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 2013 23:59:59 Aug 12 2013`	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:

	Command or Action	Purpose
		• **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.
Step 6	(Optional) **show key chain** *name* [**mode decrypt**] **Example:** `switch(config-keychain-key)# show key chain bgp-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.

Configuring a Key for OSPFv2 Cryptographic Authentication

You can configure message digest 5 (MD5) or hash-based message authentication code secure hash algorithm (HMAC-SHA) authentication for OSPFv2.

Procedure

	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 bgp-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.

	Command or Action	Purpose
		Note For OSPFv2, the key identifier in the key key-id command supports values from 0 to 255 only.
Step 4	[**no**] **cryptographic-algorithm** {**HMAC-SHA-1** | **HMAC-SHA-256** | **HMAC-SHA-384** | **HMAC-SHA-512** | **MD5**} **Example:** `switch(config-keychain-key)# cryptographic-algorithm HMAC-SHA-1`	Configures the OSPFv2 cryptographic algorithm to be used for the specified key. You can configure only one cryptographic algorithm per key.
Step 5	(Optional) **show key chain** *name* **Example:** `switch(config-keychain-key)# show key chain bgp-keys`	Shows the keychain configuration.
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.

Determining Active Key Lifetimes

To determine which keys within a key chain have active accept or send lifetimes, use the command in this table.

Command	Purpose
show key chain	Displays the key chains configured on the device.

Verifying the Keychain Management Configuration

To display keychain management configuration information, perform the following task:

Command	Purpose
show key chain *name*	Displays the keychains configured on the device.

Determining Active Key Lifetimes

To determine which keys within a key chain have active accept or send lifetimes, use the command in this table.

Command	Purpose
show key chain	Displays the key chains configured on the device.

Verifying the Keychain Management Configuration

To display keychain management configuration information, perform the following task:

Command	Purpose
show key chain *name*	Displays the keychains configured on the device.

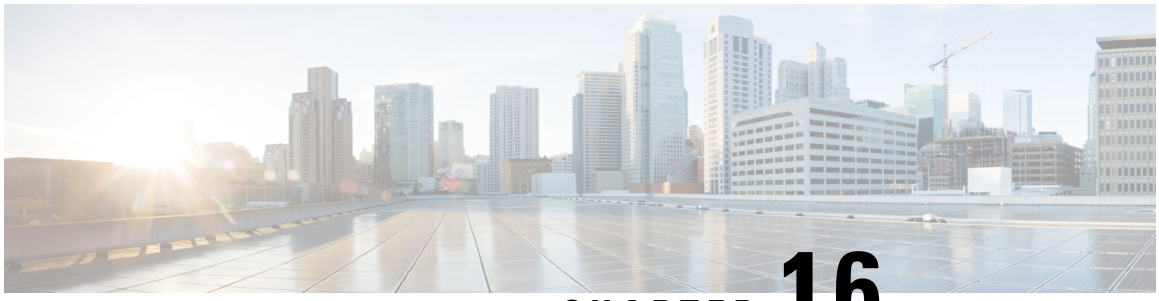
Additional References for Keychain Management

Related Documents

Related Topic	Document Title
Border Gateway Protocol	*Cisco Nexus 3400-S NX-OS Unicast Routing Configuration Guide*
OSPFv2	*Cisco Nexus 3400-S NX-OS 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.	—



CHAPTER 16

Configuring Unicast RPF

This chapter describes how to configure unicast reverse path forwarding (uRPF) on Cisco NX-OS devices.

This chapter includes the following sections:

- About Unicast RPF, on page 271
- Guidelines and Limitations for Unicast RPF, on page 272
- Default Settings for Unicast RPF, on page 273
- Configuring Unicast RPF , on page 273
- Configuration Examples for Unicast RPF, on page 275
- Verifying the Unicast RPF Configuration, on page 276
- Additional References for Unicast RPF, on page 276

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 switch 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).

Note Unicast RPF is an ingress function and is applied only on the ingress interface of a switch at the upstream end of a connection.

Unicast RPF verifies that any packet received at a switch 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.

Note With unicast RPF, all equal-cost "best" return paths are considered valid, which means that unicast RPF works where multiple return paths exist, if each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB. Unicast RPF also functions where Enhanced Interior Gateway Routing Protocol (EIGRP) variants are being used and unequal candidate paths back to the source IP address exist.

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 Cisco NX-OS software performs the following actions:

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.

Guidelines and Limitations for Unicast RPF

Unicast RPF (uRPF) has the following configuration guidelines and limitations:

- You must apply uRPF at the interface downstream from the larger portion of the network, preferably at the edges of your network.
- The further downstream that you apply uRPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying uRPF 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 uRPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying uRPF across many sites does add to the administration cost of operating the network.

- The more entities that deploy uRPF across Internet, intranet, and extranet resources means the better the chances of mitigating large-scale network disruptions throughout the Internet community and of tracing the source of an attack.
- uRPF will not inspect IP packets that are encapsulated in tunnels, such as generic routing encapsulation (GRE) tunnels. You must configure uRPF at a home gateway so that uRPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.
- You can use uRPF 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 uRPF 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 uRPF only where there is natural or configured symmetry.
- uRPF 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.

Default Settings for Unicast RPF

This table lists the default settings for unicast RPF parameters.

Table 18: Default Unicast RPF Parameter Settings

Parameters	Default
Unicast RPF	Disabled

Configuring Unicast RPF

You can configure one of the following Unicast RPF modes on an ingress interface for Cisco Nexus 3400-S Series switches.

Strict Unicast RPF mode
A strict mode check is successful when Unicast RPF 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.

Procedure

	Command or Action	Purpose

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

Step 2	[**no**] **system urpf disable** **Example:** `switch(config)# no system urpf disable`	Enables Unicast RPF on the switch. **Note** You must reload the Cisco NX-OS box to apply the Unicast RPF configuration.

Step 3	**interface ethernet** *slot/port*
	Example:
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		Specifies an Ethernet interface and enters interface configuration mode.

Step 4	{**ip** | **ipv6**} **address** *ip-address/length* **Example:** `switch(config-if)# ip address 172.23.231.240/23`	Specifies an IPv4 or IPv6 address for the interface.
Step 5	{**ip** | **ipv6**} **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 both IPv4 and IPv6. **Note** When you enable Unicast RPF for IPv4 or IPv6 (using the **ip** or **ipv6** keyword), Unicast RPF is enabled for both IPv4 and IPv6. You can configure only one version of the available IPv4 and IPv6 Unicast RPF command on an interface. When you configure one version, all the mode changes must be done by this version and all other versions will be blocked by that interface. • 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. **Note** The **allow-default** keyword is not applicable in the ALPM routing mode.

	Command or Action	Purpose
		Note The source address lookup (in case of a loose Unicast RPF check) does not match the default route if you do not specify the **allow-default** keyword. • The **rx** keyword specifies strict Unicast RPF.

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

Step 7	(Optional) **show ip interface ethernet** *slot/port*
	Example:
	`switch(config)# show ip interface ethernet 1/54 | grep -i "unicast reverse path forwarding"`
	`IP unicast reverse path forwarding: none`
		Displays the IP information for an interface and verifies if Unicast RPF is enabled.

Step 8	(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 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuration Examples for Unicast RPF

The following example shows how to configure loose unicast RPF for IPv4 packets on a Cisco Nexus 3400-S Series switch:

```
interface Ethernet2/3
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via any
```

The following example shows how to configure loose unicast RPF for IPv6 packets on a Cisco Nexus 3400-S Series switch:

```
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 loose unicast RPF for IPv4 packets on a Cisco Nexus 3400-S Series switch:

```
no system urpf disable
interface Ethernet2/3
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via any
```

The following example shows how to configure loose unicast RPF for IPv6 packets on a Cisco Nexus 3400-S Series switch:

```
no system urpf disable
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 RPF for IPv4 packets on a Cisco Nexus 3400-S Series switch:

```
no system urpf disable
interface Ethernet2/2
  ip address 172.23.231.240/23
  ip verify unicast source reachable-via rx
```

The following example shows how to configure strict unicast RPF for IPv6 packets on a Cisco Nexus 3400-S Series switch:

```
no system urpf disable
interface Ethernet2/4
  ipv6 address 2001:0DB8:c18:1::3/64
  ipv6 verify unicast source reachable-via rx
```

Verifying the Unicast RPF Configuration

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

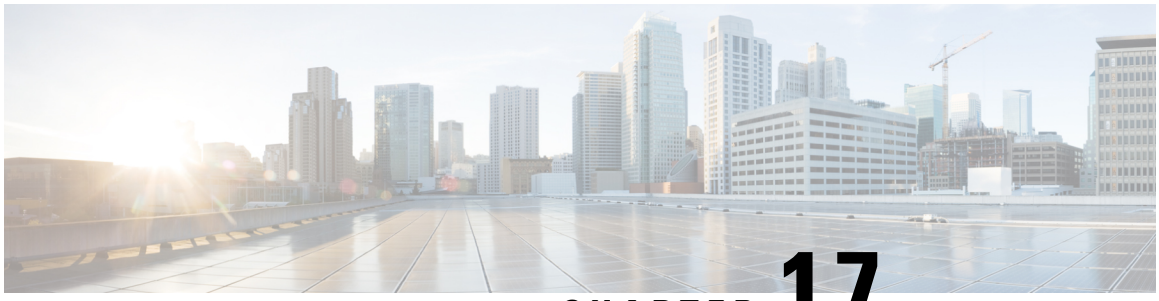
Command	Purpose
show running-config interface ethernet *slot*/*port*	Displays the interface configuration in the running configuration.
show running-config ip [**all**]	Displays the IPv4 configuration in the running configuration.
show running-config ipv6 [**all**]	Displays the IPv6 configuration in 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.

Additional References for Unicast RPF

This section includes additional information related to implementing unicast RPF.

Related Documents

Related Topic	Document Title
MPLS VPN	Cisco Nexus 3400-S NX-OS Label Switching Configuration Guide.



CHAPTER 17

Configuring Control Plane Policing

This chapter contains the following sections:

- About CoPP, on page 279
- Control Plane Protection, on page 280
- Guidelines and Limitations for CoPP, on page 280
- Default Settings for CoPP, on page 282
- Configuring CoPP, on page 282
- Verifying the CoPP Configuration, on page 287
- Displaying the CoPP Configuration Status, on page 289
- Monitoring CoPP, on page 289
- Monitoring CoPP with SNMP, on page 290
- Clearing the CoPP Statistics, on page 290
- Configuration Examples for CoPP, on page 290
- Additional References for CoPP, on page 291

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 entering the switch from a non-management port. A common attack vector for network devices is the denial-of-service (DoS) attack, where excessive traffic is directed at the device interfaces.

The Cisco NX-OS 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 Cisco NX-OS 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 Cisco NX-OS 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 Cisco NX-OS device. For example, a DoS attack on the supervisor module 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 include:

- 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.

Control Plane Protection

To protect the control plane, the Cisco NX-OS device segregates different packets destined for the control plane into different classes. Once these classes are identified, the Cisco NX-OS device polices the packets, which ensures that the supervisor module is not overwhelmed.

Guidelines and Limitations for CoPP

CoPP has the following configuration guidelines and limitations:

- Before you upgrade to Release 9.3(7), set the scale factor to 1 using the **scale-factor 1 module** *multiple-module-range* command.
- For CIR lower than 7812 pps, the policer works in steps of 122 pps. For CIR greater than 7812 pps, you can expect a 1.6% deviation in the configured CIR.
- 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.
- 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 Cisco NX-OS device and require a console connection.
- The Cisco NX-OS 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 Cisco NX-OS 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.
- If multiple flows map to the same class, individual flow statistics will not be available.
- Skip CoPP policy option has been removed from the Cisco NX-OS initial setup utility because using it can impact the control plane of the network.
- Custom CoPP with user defined class-map is not supported.
- The copp-system-class-fcoe class is not supported for Cisco Nexus 3400-S Series switches.
- The following guidelines and limitations apply to static CoPP ACLs:
    - Only Cisco Nexus 3400-S Series switches use static CoPP ACLs.

- Access control entries (ACEs) cannot be modified or removed for static CoPP ACLs.
- If a CoPP ACL has a static ACL substring, it will be mapped to that type of traffic. For example, if the ACL includes the acl-mac-stp substring, STP traffic will be classified to the class map for that ACL.
- You must have static CoPP ACLs in the CoPP policy. Otherwise, the CoPP policy will be rejected.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings for CoPP

This table lists the default settings for CoPP parameters.

Table 19: 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

Configuring CoPP

This section describes how to configure CoPP.

Configuring a Control Plane Policy Map

You must configure a policy map for CoPP, which includes policing parameters. If you do not configure a policer for a class, the default is configured. Configuration changes are permitted only to those control plane policy maps that are a copy of one of the CoPP best practice policy profiles. For more information, see Copying the CoPP Best Practice Policy, on page 286.

Before you begin

Ensure that you have configured a control plane class map.

Procedure

	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	Enter one of the following commands: • **police** [**cir**] {*cir-rate* [*rate-type*]} • **police** [**cir**] {*cir-rate* [*rate-type*]} [**bc**] *burst-size* [*burst-size-type*] • **police** [**cir**] {*cir-rate* [*rate-type*]]} **conform transmit** [**violate drop**] **Example:** `switch(config-pmap-c)# police cir 52000 bc 1000 packets`	Specifies the committed information rate (CIR). The rate range is 25 to 60000000 (60 Million) pps.. The committed burst (BC) range is 1:1073741 packets. The **conform transmit** action transmits the packet. **Note** You can specify the BC and conform action for the same CIR.
Step 5	(Optional) **set cos** *cos-value* **Example:** `switch(config-pmap-c)# set cos 1`	Specifies the 802.1Q class of service (CoS) value. The range is from 0 to 7. The default value is 0.

Step 6	**exit**
	Example:
	`switch(config-pmap-c)# exit`
	`switch(config-pmap)#`
		Exits policy map class configuration mode.

Step 7	**exit**
	Example:
	`switch(config-pmap)# exit`
	`switch(config)#`
		Exits policy map configuration mode.

Step 8	(Optional) **show policy-map type control-plane** [**expand**] [**name** *class-map-name*]	Displays the control plane policy map configuration.

	Command or Action	Purpose
	Example: `switch(config)# show policy-map type control-plane`	
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

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 have configured a control plane policy map.

Procedure

	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	[**no**] **service-policy input** *policy-map-name* **Example:** `switch(config-cp)# service-policy input PolicyMapA`	Specifies a policy map for the input traffic. Repeat this step if you have more than one policy map. You cannot disable CoPP. If you enter the **no** form of this command, packets are rate limited at 125 packets per seconds.

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.

	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.

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.

Procedure

	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	**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 policy-map interface control-plane** **Example:** `switch(config-cp)# show policy-map interface control-plane`	Displays the applied scale factor values when a CoPP policy is applied.

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

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.

Procedure

	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. You cannot disable CoPP. If you enter the **no** form of this command, packets are rate limited at 125 packets per seconds.
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.

Copying the CoPP Best Practice Policy

The CoPP best practice policy is read-only. If you want to modify its configuration, you must copy it.

Procedure

	Command or Action	Purpose
Step 1	**copp copy profile** {**strict** | **moderate** | **lenient** | **dense**} {**prefix** | **suffix**} *string* **Example:** `switch# copp copy profile strict prefix abc`	Creates a copy of the CoPP best practice policy. CoPP renames all class maps and policy maps with the specified prefix or suffix.
Step 2	(Optional) **show copp status** **Example:** `switch# show copp status`	Displays the CoPP status, including the last configuration operation and its status. This command also enables you to verify that the

	Command or Action	Purpose
		copied policy is not attached to the control plane.
Step 3	(Optional) **show running-config copp** **Example:** `switch# show running-config copp`	Displays the CoPP configuration in the running configuration, including the copied policy configuration.

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	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 and BC values internally on each module, but the display shows the configured CIR and BC 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.

Command	Purpose
show copp diff profile {**strict** | **moderate** | **lenient** | **dense**} [**prior-ver**] **profile** {**strict** | **moderate** | **lenient** | **dense**} **show copp diff profile**	Displays the difference between two CoPP best practice policies. When you do not include the prior-ver option, this command displays the difference between two currently applied default CoPP best practice policies (such as the currently applied strict and currently applied moderate policies). When you include the prior-ver option, this command displays the difference between a currently applied default CoPP best practice policy and a previously applied default CoPP best practice policy (such as the currently applied strict and the previously applied lenient policies).
show copp profile {**strict** | **moderate** | **lenient** | **dense**}	Displays the details of the CoPP best practice policy, along with the classes and policer values.
show running-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the running configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the running configuration.
show running-config copp [**all**]	Displays the CoPP configuration in the running configuration.
show startup-config aclmgr [**all**]	Displays the user-configured access control lists (ACLs) in the startup configuration. The **all** option displays both the default (CoPP-configured) and user-configured ACLs in the startup configuration.

Displaying the CoPP Configuration Status

Procedure

	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
```

Monitoring CoPP

Procedure

	Command or Action	Purpose
Step 1	switch# **show policy-map interface control-plane**	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).

Example

This example shows how to monitor CoPP:

```
switch# show policy-map interface control-plane
Control Plane

  Service-policy  input: copp-system-p-policy-strict

    class-map copp-system-p-class-critical (match-any)
      set cos 7
      police cir 19000 pps , bc 128 packets
      module 4 :
        transmitted 373977 packets;
        dropped 0 packets;
```

Monitoring CoPP with SNMP

Beginning with Cisco Nexus Release 9.2(3), CoPP supports the Cisco class-based QoS MIB (cbQoSMIB). All 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

Clearing the CoPP Statistics

Procedure

	Command or Action	Purpose
Step 1	(Optional) switch# **show policy-map interface control-plane**	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
```

Configuration Examples for CoPP

This section includes example CoPP configurations.

Additional References for CoPP

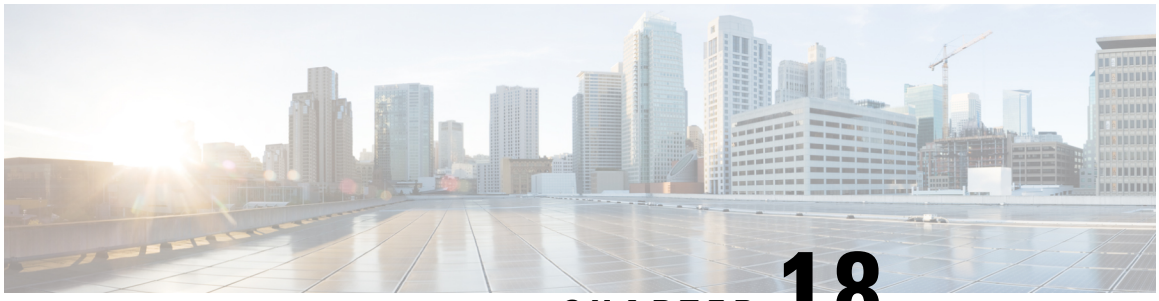
This section provides additional information related to implementing CoPP.

Related Documents

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*

Standards

Standards	Title
RFC 2698	A Two Rate Three Color Marker



CHAPTER 18

Configuring Rate Limits

This chapter describes how to configure rate limits for supervisor-bound traffic on Cisco NX-OS devices.

This chapter includes the following sections:

- About Rate Limits, on page 293
- Guidelines and Limitations for Rate Limits, on page 293
- Default Settings for Rate Limits, on page 294
- Configuring Rate Limits, on page 294
- Monitoring Rate Limits, on page 296
- Clearing the Rate Limit Statistics, on page 296
- Verifying the Rate Limit Configuration, on page 297
- Configuration Examples for Rate Limits, on page 297
- Additional References for Rate Limits, on page 297

About Rate Limits

Rate limits can prevent redirected packets for exceptions from overwhelming the supervisor module on a Cisco NX-OS device. You can configure rate limits in packets per second for the following types of redirected packets:

- Bidirectional forwarding detection (BFD) packets
- Sflow

Guidelines and Limitations for Rate Limits

Rate limits 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 Hardware rate-limiters protect the supervisor CPU from excessive inbound traffic. The traffic rate allowed by the hardware rate-limiters is configured globally and applied to each individual I/O module. The resulting allowed rate depends on the number of I/O modules in the system. CoPP provides more granular supervisor CPU protection by utilizing the modular quality-of-service CLI (MQC).

- You can configure a hardware rate-limiter to show statistics for outbound traffic on SPAN egress ports.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Default Settings for Rate Limits

This table lists the default settings for rate limits parameters.

Table 20: Default Rate Limits Parameters Settings

Parameters	Default
BFD packets rate limit	10,000 packets per second
Sflow	40,000 packets per second

Configuring Rate Limits

You can set rate limits on supervisor-bound traffic.

Procedure

	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* | **disable**} [**module** *module* [**port** *start end*]] **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 10000.

	Command or Action	Purpose
Step 3	**hardware rate-limiter bfd** *packets* [**module** *module* [**port** *start end*]] **Example:** `switch(config)# hardware rate-limiter bfd 500`	Configures rate limits in packets per second for bidirectional forwarding detection (BFD) packets. The range is from 0 to 10000.
Step 4	**hardware rate-limiter exception** *packets* [**module** *module* [**port** *start end*]] **Example:** `switch(config)# hardware rate-limiter exception 500`	Configures rate limits in packets per second for any exception traffic in the system that is not classified by the Control Plane Policing (CoPP) policy. The range is from 0 to 10000.
Step 5	**hardware rate-limiter layer-3 glean** *packets* [**module** *module* [**port** *start end*]] **Example:** `switch(config)# hardware rate-limiter layer-3 glean 500`	Configures rate limits in packets per second for Layer 3 glean packets. The range is from 0 to 10000. A node receiving traffic for a particular destination might be unable to forward traffic because it is unaware of the rewrite information or the physical layer interface behind which the destination resides. During this time, it is possible to install a glean entry in the data path for that destination. Because this might not be a pointer to the global punt adjacency, a reserved module or port value is used to punt such packets to the supervisor. This glean rate can be controlled using the given rate limiter. **Note** The CoPP policy controls the rate of glean packets that are forwarded due to global punt adjacency, and this rate limiter controls the destination-specific glean packets.
Step 6	**hardware rate-limiter layer-3 multicast local-groups** *packets* **Example:** `switch(config)# hardware rate-limiter layer-3 multicast local-groups 300`	Configures rate limits in packets per second for Layer 3 multicast data packets that are punted for initiating a shortest-path tree (SPT) join. The range is from 0 to 10000.
Step 7	**hardware rate-limiter span-egress** *rate* [**module** *module*] **Example:** `switch(config)# hardware rate-limiter span-egress 123`	Configures rate limits in kilobits per second for SPAN for egress traffic. The range is from 0 to 100000000. **Note** You should not configure both sFlow and the SPAN egress rate-limiter.

	Command or Action	Purpose
Step 8	(Optional) **show hardware rate-limiter** [**access-list-log | bfd |exception | fex | layer-3 glean | layer-3 multicast local-groups | span-egress |module** *module*] **Example:** `switch# show hardware rate-limiter`	Displays the rate limit configuration. The module range is from 1 to 30.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Monitoring Rate Limits

You can monitor rate limits.

Procedure

	Command or Action	Purpose
Step 1	**show hardware rate-limiter** [**access-list-log** | **bfd** | **exception** | **fex** | **layer-3 glean** | **layer-3 multicast local-groups** | **span-egress** | **module** *module*] **Example:** `switch# show hardware rate-limiter access-list-log`	Displays the rate limit statistics.

Clearing the Rate Limit Statistics

You can clear the rate limit statistics.

Procedure

	Command or Action	Purpose
Step 1	**clear hardware rate-limiter** {**all** | **access-list-log** | **bfd** | **exception** | **fex** | **layer-3 glean** | **layer-3 multicast local-groups** [**module** *module*] } **Example:** `switch# clear hardware rate-limiter access-list-log`	Clears the rate limit statistics.

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** | **bfd** | **exception** | **fex** | **layer-3 glean** | **layer-3 multicast local-groups** | **module** *module*]	Displays the rate limit configuration.

Configuration Examples for Rate Limits

The following example shows how to configure rate limits for packets copied to the supervisor module for access list logging:

```
switch(config)# hardware rate-limiter access-list-log
switch(config)# show hardware rate-limiter access-list-log
Units for Config: packets per second
Allowed, Dropped & Total: aggregated since last clear counters


Module: 4
  R-L Class           Config           Allowed         Dropped             Total
 +------------------+--------+---------------+---------------+-----------------
+
  access-list-log         100               0               0                 0

  Port group with configuration same as default configuration
      Eth4/1-36

Module: 22
  R-L Class           Config           Allowed         Dropped             Total
 +------------------+--------+---------------+---------------+-----------------
+
  access-list-log         100               0               0                 0

  Port group with configuration same as default configuration
      Eth22/1-0
```

Additional References for Rate Limits

This section includes additional information related to implementing rate limits.

Related Documents

Related Topic	Document Title
Cisco NX-OS licensing	*Cisco NX-OS Licensing Guide*

INDEX

A

aaa accounting default group 22
aaa accounting default local 22
aaa authentication login {mschap | mschapv2} enable 20
aaa authentication login ascii-authentication 70
aaa authentication login chap enable 19
aaa authentication login console 12–13
aaa authentication login console group 12–13
aaa authentication login console local 12–13
aaa authentication login console none 12–13
aaa authentication login error-enable 16
aaa authorization {commands | config-commands} {console | default} {group} 73
aaa authorization {group | local} 93
aaa authorization {ssh-certificate | ssh-publickey} 93
aaa authorization default 93
aaa authorization ssh-certificate default 71
aaa group server ldap 88
aaa group server radius 37
aaa group server tacacs+ 63
aaa user default-role 16
absolute end 180
absolute start 180
accept-lifetime 267
action {drop | forward | redirect} 194
authentication (bind-first | compare} 88

B

BGP 272
  using with Unicast RPF 272

C

chgrp 99
chown 99
class 283
class class-default 283
class insert-before 283
clear accounting log 23
clear copp statistics 290
clear hardware rate-limiter {all | access-list-log | bfd | exception | fex | layer-3 glean | layer-3 multicast local-groups | span-egress} 296
clear hardware rate-limiter module 296
clear ip access-list counters 172
clear ip dhcp global statistics 234
clear ip dhcp relay statistics interface 234
clear ip dhcp snooping binding interface ethernet 233
clear ip dhcp snooping binding interface port-channel 233
clear ip dhcp snooping binding vlan 234
clear ip dhcp snooping statistics 234
clear ip dhcp snooping statistics vlan 234
clear ipv6 access-list counters 172
clear ipv6 dhcp relay statistics interface 234
clear ldap-server statistics 95
clear line 115, 117
clear mac access-list counters 189
clear radius-server statistics 52
clear ssh hosts 113
control-plane 284–285
copp copy profile {strict | moderate | lenient| dense 286
copp copy profile prefix | suffix} 286
copp profile 286
copp profile dense 286
copp profile lenient 286
copp profile moderate 286
copp profile strict 286
copy scp 106, 119
copy scp: 105
copy sftp 106, 119
crypto ca authenticate 109
crypto ca crl request 109
crypto ca trustpoint 109
cryptographic-algorithm {HMAC-SHA-1 | HMAC-SHA-256 | HMAC-SHA-384 | HMAC-SHA-512 | MD5} 269

D

deadtime 37
deny 158–161
description 131
DHCP client relay on orphan ports 240
  description 240
DHCP relay on VPC Leg 239
  description 239
DHCP relay on-stack 239
  description 239

E

enable Cert-DN-match 88
enable user-server-group 88
encryption decrypt type6 258
encryption delete type6 258
encryption re-encrypt obfuscated 257

F

feature 132
feature dhcp 209
feature ldap 85
feature password encryption aes 257, 265
feature scp-server 107
feature sftp-server 108
feature ssh 101, 112
feature tacacs+ 59
feature telnet 115
fragments {permit-all | deny-all} 158–159

H

hardware access-list tcam region 164
hardware access-list tcam region racl qualify udf 166
hardware rate-limiter access-list-log 294
hardware rate-limiter bfd 295
hardware rate-limiter exception 295
hardware rate-limiter layer-3 glean 295
hardware rate-limiter layer-3 multicast local-groups 295
hardware rate-limiter span-egress 295
host 174–175

I

interface policy dent 133
ip access-class 161
ip access-group 168
ip access-list 158–159, 161, 167
ip dhcp packet strict-validation 201, 214
ip dhcp relay 219
ip dhcp relay address 222–223
ip dhcp relay address use-vrf 222
ip dhcp relay information option 220
ip dhcp relay information option trust 216
ip dhcp relay information trust-all 218
ip dhcp relay information trusted 217
ip dhcp relay source-interface 223, 230
ip dhcp relay sub-option circuit-id customized 220
ip dhcp relay sub-option circuit-id format-type string 220
ip dhcp smart-relay 225
ip dhcp smart-relay global 224
ip dhcp snooping information option 213
ip dhcp snooping trust 215
ip dhcp snooping verify mac-address 212
ip dhcp snooping vlan 211
ip port access group 169
ip radius source-interface 38
ip tacacs source-interface 64
ip verify unicast source reachable-via 274
ipv6 access-class 161
ipv6 access-list 158–159, 161
ipv6 address use-link-local-only 232
ipv6 dhcp relay 226
ipv6 dhcp relay address 229
ipv6 dhcp relay option type cisco 221, 227
ipv6 dhcp relay option vpn 221, 227
ipv6 port traffic-filter 169
ipv6 traffic-filter 168
ipv6 verify unicast source reachable-via 274

K

key 266–267, 269
key chain 263, 265, 267–268
key config-key ascii 256, 264
key-string 266

L

ldap search-map 91
ldap-server deadtime 92
ldap-server host 86, 90, 92
ldap-server host idle-time 92
ldap-server host password 87, 92
ldap-server host port 87, 90
ldap-server host rootDN 87
ldap-server host test rootDN 92
ldap-server host timeout 87, 90
ldap-server host username 92
ldap-server timeout 89, 93
line vty 161
login on-failure log 17
login on-success log 17

M

mac access-list 184–185
mac port access-group 188
match {ip | ipv6} address 193
match mac address 193

N

no {periodic | absolute} 180
no aaa authentication login {console | default | fallback error local 15
no aaa authentication login ascii-authentication 19–20
no feature ssh 100, 112–114
no feature tacacs+ 77
no host 174–175

no ip access-list 163
no ipv6 access-list 163
no key chain 264
no mac access-list 187
no object-group {ip address | ipv6 address | ip port} 177
no ssh key dsa 114
no ssh key rsa 114
no time-range 181
no vlan access-map 195

O

object-group ip address 174
object-group ip port 176
object-group ipv6 address 175

P

password strength-check 127
periodic 179–180
permit 158–161
permit | deny 184
permit interface 134
permit ip 167
permit udf 167
permit vlan 135
permit vrf 136
police 283
police cir 283
policy-map type control-plane 283

R

radius commit 34, 39–40, 43, 47
radius-server deadtime 44, 46–47
radius-server directed-request 39
radius-server host 34, 36–37, 41–42, 46
radius-server host accounting 43
radius-server host acct-port 42
radius-server host auth-port 43
radius-server host authentication 43
radius-server host idle-time 46
radius-server host password 46
radius-server host retransmit 41
radius-server host test 46
radius-server host timeout 41
radius-server host username 46
radius-server key 35
radius-server retransmit 40
radius-server test {idle-time} 44
radius-server test {password} 44
radius-server test {username} 44
radius-server timeout 40
reload 164, 167, 170
resequence {ip | ipv6} access-list 162
resequence mac access-list 186
resequence time-range 181
role commit 131, 133–136
role feature-group name 132
role name 130, 133, 135–136
role name priv 75
rule {deny | permit) command 130
rule {deny | permit} {read | read-write} 130
rule {deny | permit} {read | read-write} feature 130
rule {deny | permit} {read | read-write} feature-group 131
rule {deny | permit} {read | read-write} oid 131
rule {deny | permit} command 75

S

scale-factor 285
send-lifetime 267
server 37, 63, 88
service-policy input 284
set cos 283
show {ip | ipv6 | access-lists} 177
show aa accounting 23
show aaa accounting 22
show aaa authentication 12, 14–15, 17, 23
show aaa authentication login {ascii-authentication | chap | error-enable | mschap | mschapv2} 23
show aaa authentication login {mschap | mschapv2} 21
show aaa authentication login chap 19
show aaa authorization 72, 74, 94
show aaa authorization all 72
show aaa groups 23
show aaa user default-role 16
show accounting log 23
show class-map type control-plane 287
show cli syntax roles network-admin 139
show cli syntax roles network-operator 139
show copp profile 288
show copp status 286, 289
show crypto ca certificates 110, 117
show crypto ca crl 110, 117
show encryption service stat 257, 265
show hardware access-list tcam region 164, 171
show hardware rate-limiter 296–297
show hardware rate-limiter access-list-log 296–297
show hardware rate-limiter bfd 296–297
show hardware rate-limiter exception 296–297
show hardware rate-limiter fex 296–297
show hardware rate-limiter layer-3 glean 296–297
show hardware rate-limiter layer-3 multicast local-groups 296–297
show hardware rate-limiter module 296–297
show hardware rate-limiter span-egress 296–297
show ip access-lists 158, 160–162, 171–172
show ip access-lists summary 163
show ip dhcp relay 216, 219–220, 224, 226, 230, 232
show ip dhcp relay address 233
show ip dhcp relay information trusted-sources 216–218

show ip dhcp relay statistics 234
show ip dhcp snooping binding 233
show ipv6 access-lists 158, 160–161, 171–172
show ipv6 access-lists summary 163
show ipv6 dhcp relay 221, 226–227, 233
show ipv6 dhcp relay interface 221, 227
show ipv6 dhcp relay statistics 234
show key chain 263–264, 266, 268–270
show key chain mode decrypt 266, 268
show ldap-search-map 91, 95
show ldap-server 86–87, 89–90, 92–93, 95
show ldap-server groups 89, 95
show ldap-server statistics 94–95
show login on-failure log 18
show login on-successful log 18
show mac access-lists 184, 186–187, 189
show object-group 174–177
show password strength-check 127
show policy-map interface control-plane 285, 287, 289–290
show policy-map type control-plane 283, 287
show policy-map type control-plane expand 283
show policy-map type control-plane name 283
show privilege 78
show radius {pending | pending-diff} 34, 39–41, 43, 47
show radius {status | pending | pending-diff} 50
show radius-server 34–36, 38, 40, 42–43, 45–47, 51
show radius-server directed-request 39
show radius-server groups 37
show radius-server statistics 51
show role 128, 131, 134–136, 139
show role {pending | pending-diff} 131, 133–136
show role feature 139
show role feature-group 133, 139
show running-config aaa 23
show running-config aclmgr 168–169, 171, 177, 188–189, 194–196, 288
show running-config aclmgr all 171, 189
show running-config copp 284, 286–288
show running-config copp all 284
show running-config dhcp 209, 211–212, 214–221, 223–230
show running-config interface ethernet 232, 276
show running-config interface mgmt 0 232
show running-config interface vlan 232
show running-config ip 276
show running-config ipv6 276
show running-config ldap 95
show running-config radius 51
show running-config security 108, 117, 139
show running-config security all 104, 117, 139
show running-config tacacs 78
show running-config tacacs all 78
show ssh key 101, 114
show ssh server 113, 117
show startup-config aaa 23
show startup-config acllog 171
show startup-config aclmgr 171, 189, 196, 288
show startup-config aclmgr all 171, 189, 196
show startup-config dhcp 233
show startup-config dhcp all 233
show startup-config interface ethernet 276
show startup-config ip 276
show startup-config ldap 95
show startup-config radius 51
show startup-config security 139
show startup-config tacacs 78
show tacacs-server 60–62, 64, 66–67, 69–71, 78
show tacacs-server directed-request 65, 78
show tacacs-server groups 63, 78
show tacacs-server sorted 78
show tacacs-server statistics 77–78
show tacacs+ {pending | pending-diff} 60, 65–67, 69–70, 73
show tacacs+ {status | pending | pending-diff} 78
show telnet server 115, 117
show time-range 180–182
show user-account 102–103, 110, 117, 129, 138–139
show username keypair 117
show users 110, 115–117
show vlan access-map 196
show vlan filter 196
ssh 105
ssh key 100
ssh key force 100
ssh key rsa 100
ssh login-attempts 104
ssh vrf 105
ssh6 105
ssh6 vrf 105
statistics per-entry 158, 160, 184, 186, 194

T

tacacs-server dead-time 68
tacacs-server deadtime 69
tacacs-server directed-request 65
tacacs-server host 60, 62–63, 66–67
tacacs-server host port 67
tacacs-server host timeout 66
tacacs-server key 61
tacacs-server test 68
tacacs-server test idle-time 68
tacacs-server test username 68
tacacs+ commit 60, 65–67, 70–71, 73
telnet 116
telnet vrf 116
telnet6 116
telnet6 vrf 116
terminal no verify-only 75
terminal no verify-only username 75
terminal verify-only 75
terminal verify-only username 75
test aaa authorization command-type {commands | config-commands} user command 74
test aaa group 48, 76

test aaa server radius 48
test aaa server radius vrf 48
test aaa server tacacs+ 76
time-range 179

U

udf 166
Unicast RPF 271–273, 275–276
  BGP attributes 272
  default settings 273
  description 271
  example configurations 275
  FIB 271
  implementation 272
  verifying configuration 276
use-vrf 37, 88
username 102
username password 108, 128
username sshkey 103
username sshkey file bootflash 102

V

vlan access-map 193
vlan filter 195
vlan policy deny 135
vPC First Hop Security Configuration 238
  description 238
vrf policy deny 136