Cisco APIC Security Configuration Guide, Release 4.0(1)

First Published: 2018-10-24

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

© 2018–2023 Cisco Systems, Inc. All rights reserved.

Trademarks

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: https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html. Cisco product warranty information is available at https://www.cisco.com/c/en/us/products/warranty-listing.html. US Federal Communications Commission Notices are found here https://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: https://www.cisco.com/c/en/us/about/legal/trademarks.html. 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)

Trademarks

CONTENTS

PREFACE

Trademarks iii

CHAPTER 1

New and Changed Information 1

New and Changed Information 1

CHAPTER 2

Overview 3

Overview 3

CHAPTER 3

Access, Authentication, and Accounting 5

Overview 5

User Access, Authorization, and Accounting 5

Multiple Tenant Support 5

User Access: Roles, Privileges, and Security Domains 5

Access Rights Workflow Dependencies 7

AAA RBAC Roles and Privileges 7

Custom Roles 17

Selectively Expose Physical Resources across Security Domains 17

Enable Sharing of Services across Security Domains 18

APIC Local Users 18

Externally Managed Authentication Server Users 21

Cisco AV Pair Format 23

Change Remote User Role 24

About Signature-Based Transactions 26

Guidelines and Limitations 26

Accounting 26

Routed Connectivity to External Networks as a Shared Service Billing and Statistics 27

Configuration 27
Configuring a Local User 27
Configuring a Local User Using the GUI 28
Configuring SSH Public Key Authentication Using the GUI 29
Configuring a Local User Using the NX-OS Style CLI 29
Configuring a Local User Using the REST API 30
Generating an X.509 Certificate and a Private Key 30
Creating a Local User and Adding a User Certificate Using the GUI 32
Creating a Local User and Adding a User Certificate Using the REST API 32
Creating a Local User Using Python SDK 34
Using a Private Key to Calculate a Signature 36
Configuring a Local User for OTP-Based Two-Factor Authentication Using the GUI 38
Completing the Configuration of OTP-Based Two-Factor Authentication by a User Using the GUI 39

CHAPTER 4 **TACACs+, RADIUS, LDAP, RSA, and SAML 41**
Overview 41
RADIUS 41
TACACS+ Authentication 42
User IDs in the APIC Bash Shell 42
Login Domains 43
LDAP/Active Directory Authentication 43
RSA Secure ID Authentication 44
Configuring APIC for RSA Access Using the GUI 44
Configuring a Remote User 45
AV Pair on the External Authentication Server 45
Best Practice for Assigning AV Pairs 46
Configuring an AV Pair on the External Authentication Server 46
Configuring APIC for TACACS+ Access 47
Configuring APIC for RADIUS Access 48
Configuring a Cisco Secure Access Control Server for RADIUS and TACACS+ Access to the APIC 49
Configuring LDAP 51
Configuring Windows Server 2008 LDAP for APIC Access with Cisco AVPair 51

Configuring APIC for LDAP Access 52
Configuring LDAP Group Map Rules on the Cisco APIC 53
Configuring an LDAP Group Map on the Cisco APIC 54
Configuring a Remote User Using the NX-OS Style CLI 54
Changing the Default Behavior for Remote Users with Missing or Bad Cisco AV Pairs 55
Changing Default Behavior for Remote Users with Missing or Bad Cisco AV Pairs Using the NX-OS Style CLI 55
About SAML 56
Basic Elements of SAML 57
Supported IdPs and SAML Components 58
Configuring APIC for SAML Access 60
Setting Up a Relying Party Trust in AD FS 61

CHAPTER 5 802.1X 63
802.1X Overview 63
Host Support 63
Authentication Modes 64
Guidelines and Limitations 64
Configuration Overview 65
Configuring 802.1X Port Authentication Using the APIC GUI 65
Configuring 802.1X Node Authentication Using the APIC GUI 66
Configuring 802.1X Port Authentication Using the NX-OS Style CLI 67
Configuring 802.1X Node Authentication Using NX-OS Style CLI 68
Configuring 802.1X Port Authentication Using the REST API 69
Configuring 802.1X Node Authentication Using the REST API 69

CHAPTER 6 Port Security 71
About Port Security and ACI 71
Port Security Guidelines and Restrictions 71
Port Security at Port Level 72
Configuring Port Security Using the APIC GUI 72
Configuring Port Security Using REST API 73
Configuring Port Security Using the CLI 73
Port Security and Learning Behavior 75

Protect Mode 75

CHAPTER 7 First Hop Security 77

About First Hop Security 77

ACI FHS Deployment 78

Guidelines and Limitations 78

Configuring FHS Using the APIC GUI 79

Configuring FHS Using the NX-OS CLI 79

FHS Switch iBASH Commands 86

Configuring FHS in APIC Using REST API 91

CHAPTER 8 Protocol Authentication 93

COOP 93

Overview 93

Using COOP with Cisco APIC 93

Guidelines and Limitations 94

Configuring COOP Authentication Using the APIC GUI 94

Configuring COOP Authentication Using the Cisco NX-OS-Style CLI 94

Configuring COOP Authentication Using the REST API 95

EIGRP 95

Overview 95

Guidelines and Limitations 95

Configuring EIGRP Authentication Using the APIC GUI 96

Configuring EIGRP Authentication Using the NX-OS CLI 96

CHAPTER 9 Control Plane Traffic 99

About Control Plane Policing 99

Guidelines and Limitations for CoPP 101

Configuring CoPP Using the APIC GUI 102

Configuring CoPP Using the Cisco NX-OS CLI 103

Configuring CoPP Using the REST API 104

Viewing CoPP Statistics Using the GUI 105

Configuring Per Interface Per Protocol CoPP Policy Using the APIC GUI 105

Configuring Per Interface Per Protocol CoPP Policy Using the NX-OS Style CLI 106

Configuring CoPP Per Interface Per Protocol Using REST API 106
About CoPP Prefilters 107
Supported Platforms 108
Limitations 108
Configuring a CoPP Prefilter, Policy Group, and Profile Using the GUI 108
Configuring a CoPP Prefilter Using the Cisco APIC GUI 108
Configuring a Leaf Policy Group Using the GUI 109
Configuring a Leaf Profile Using the GUI 110
Configuring a CoPP Prefilter Using the CLI 110
Configuring the CoPP Prefilter for a Leaf Switch Using the CLI 110
Configuring the CoPP Prefilter for a Spine Switch Using the CLI 111
Configuring a CoPP Prefilter Using the REST API 112
Configuring a CoPP Prefilter Policy for a Leaf Switch Using the REST API 112
Configuring a CoPP Prefilter Policy for a Spine Using the REST API 113

CHAPTER 10 **Fabric Security** 115
About Federal Information Processing Standards (FIPS) 115
Guidelines and Limitations for FIPS 115
Configuring FIPS for Cisco APIC Using the GUI 116
Configuring FIPS for Cisco APIC Using NX-OS Style CLI 116
Configuring FIPS for Cisco APIC Using REST API 117

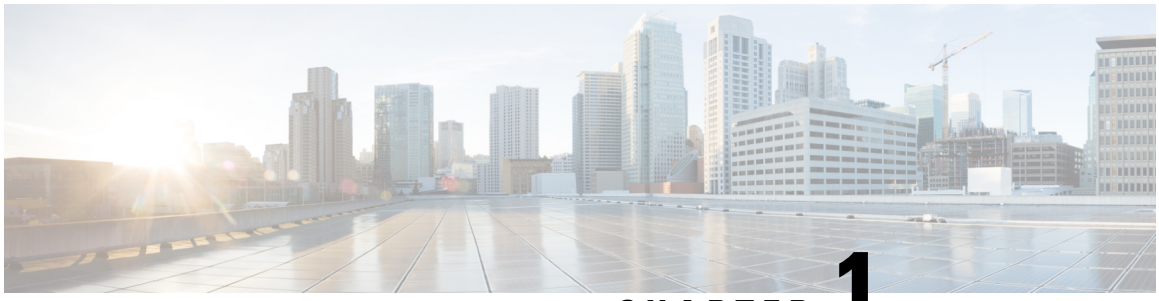
CHAPTER 11 **Security Policies** 119
ACI Fabric Network Access Security Policy Model (Contracts) 119
Access Control List Limitations 120
Contracts Contain Security Policy Specifications 120
Security Policy Enforcement 123
Multicast and EPG Security 123
Taboos 124
Enabling and Viewing ACL Contract and Deny Logs 124
About ACL Contract Permit and Deny Logs 124
Enabling ACL Contract Permit and Deny Logging Using the GUI 125
Enabling ACL Contract Permit Logging Using the NX-OS CLI 126
Enabling ACL Contract Permit Logging Using the REST API 127

Enabling Taboo Contract Deny Logging Using the GUI 127
Enabling Taboo Contract Deny Logging Using the NX-OS CLI 128
Enabling Taboo Contract Deny Logging Using the REST API 129
Viewing ACL Permit and Deny Logs Using the GUI 129
Viewing ACL Permit and Deny Logs Using the REST API 130
Viewing ACL Permit and Deny Logs Using the NX-OS CLI 131

CHAPTER 12 **Data Plane Policing** 135
Overview of Data Plane Policing 135
Guidelines and Limitations 136
Configuring Data Plane Policing for Layer 2 Interface Using the GUI 137
Configuring Data Plane Policing for Layer 3 Interface Using the APIC GUI 139
Configuring Data Plane Policing Using the REST API 140
Configuring Data Plane Policing Using NX-OS Style CLI 142
Data Plane Policing at the Endpoint Group Level 147
Configuring Data Plane Policing at the Endpoint Group Level Using CLI 148
Configuring Data Plane Policing at the Endpoint Group Level Using the APIC GUI 149
Configuring Data Plane Policing at the Endpoint Group Level Using Rest API 149
Accessing Statistics for the Data Plane Policer at the Endpoint Group Level in the GUI 150

CHAPTER 13 **HTTPS Access** 151
Overview 151
Configuring Custom Certificate Guidelines 151
Modifying the SSL Cipher Configuration 152
Mapping the Cisco APIC SSL Configuration Options to the Cipher List Formatting 152
Testing the Cipher List Format Before Modifying the Cisco APIC SSL Configuration 153
Configuring a Custom Certificate for Cisco ACI HTTPS Access Using the GUI 153
Configuring the Default SSL Protocols and Diffie-Hellman Key Exchange Using the GUI 156
Enabling Certificate Based Authentication Using the NX-OS CLI 156

CHAPTER 14 **Additional ACI Security Features** 157
Additional Security Features 157
Turning Off Generated Session Log Files in APIC 157



CHAPTER 1

New and Changed Information

This chapter contains the following sections:

- New and Changed Information, on page 1

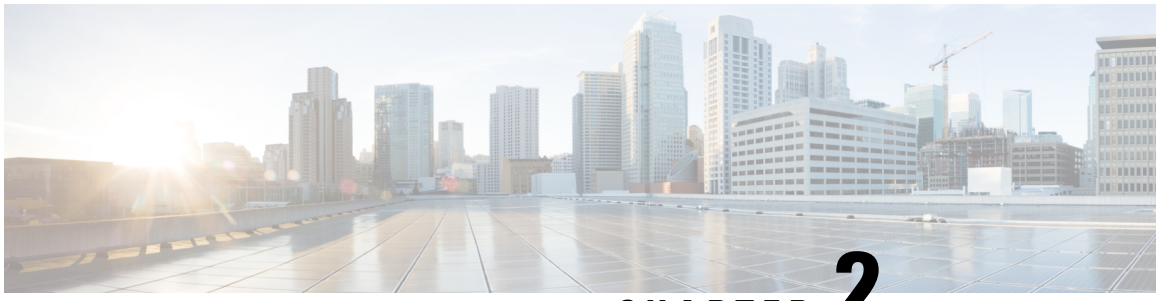
New and Changed Information

The following table provides an overview of the significant changes to the organization and features in this guide up to this current release. The table does not provide an exhaustive list of all changes made to the guide or of the new features up to this release.

Table 1: New Features and Changed Behavior in Cisco APIC for Cisco APIC Release 4.0(1)

Feature or Change	Description	Where Documented
Certificate Based Authentication	Support for Certificate based authentication.	Chapters: *Access, Authentication, and Accounting* and *HTTP Access*

New and Changed Information



CHAPTER 2

Overview

This chapter contains the following topic:

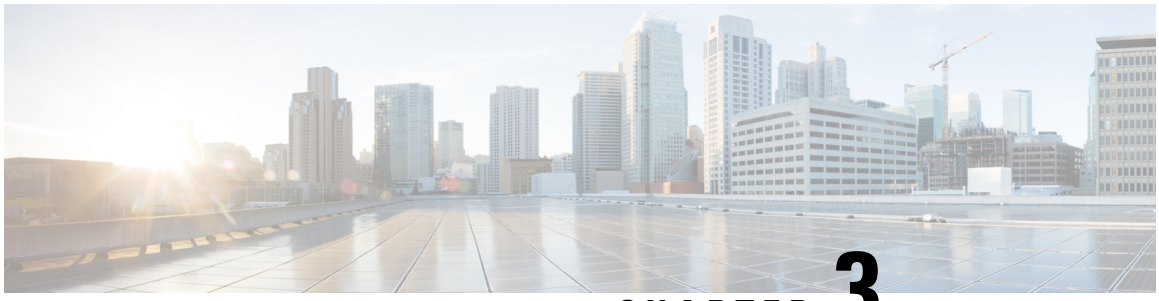
- Overview, on page 3

Overview

The Cisco Application Centric Infrastructure (ACI) 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.

For information on core fabric services, see the *Cisco APIC Basic Configuration Guide*, which you can find on the Cisco Application Policy Infrastructure Controller (APIC) documentation page.

Overview



CHAPTER 3

Access, Authentication, and Accounting

- Overview, on page 5
- Configuration, on page 27

Overview

User Access, Authorization, and Accounting

Application Policy Infrastructure Controller (APIC) policies manage the authentication, authorization, and accounting (AAA) functions of the Cisco Application Centric Infrastructure (ACI) fabric. The combination of user privileges, roles, and domains with access rights inheritance enables administrators to configure AAA functions at the managed object level in a granular fashion. These configurations can be implemented using the REST API, the CLI, or the GUI.

Note There is a known limitation where you cannot have more than 32 characters for the login domain name. In addition, the combined number of characters for the login domain name and the user name cannot exceed 64 characters.

Multiple Tenant Support

A core Application Policy Infrastructure Controller (APIC) internal data access control system provides multitenant isolation and prevents information privacy from being compromised across tenants. Read/write restrictions prevent any tenant from seeing any other tenant's configuration, statistics, faults, or event data. Unless the administrator assigns permissions to do so, tenants are restricted from reading fabric configuration, policies, statistics, faults, or events.

User Access: Roles, Privileges, and Security Domains

The APIC provides access according to a user's role through role-based access control (RBAC). An Cisco Application Centric Infrastructure (ACI) fabric user is associated with the following:

- A set of roles
- For each role, a privilege type: no access, read-only, or read-write

- One or more security domain tags that identify the portions of the management information tree (MIT) that a user can access

The ACI fabric manages access privileges at the managed object (MO) level. A privilege is an MO that enables or restricts access to a particular function within the system. For example, fabric-equipment is a privilege bit. This bit is set by the Application Policy Infrastructure Controller (APIC) on all objects that correspond to equipment in the physical fabric.

A role is a collection of privilege bits. For example, because an “admin” role is configured with privilege bits for “fabric-equipment” and “tenant-security,” the “admin” role has access to all objects that correspond to equipment of the fabric and tenant security.

A security domain is a tag associated with a certain subtree in the ACI MIT object hierarchy. For example, the default tenant “common” has a domain tag `common`. Similarly, the special domain tag `all` includes the entire MIT object tree. An administrator can assign custom domain tags to the MIT object hierarchy. For example, an administrator could assign the “solar” domain tag to the tenant named solar. Within the MIT, only certain objects can be tagged as security domains. For example, a tenant can be tagged as a security domain but objects within a tenant cannot.

Note Security Domain password strength parameters can be configured by creating **Custom Conditions** or by selecting **Any Three Conditions** that are provided.

Creating a user and assigning a role to that user does not enable access rights. It is necessary to also assign the user to one or more security domains. By default, the ACI fabric includes two special pre-created domains:

- `All`—allows access to the entire MIT
- `Infra`— allows access to fabric infrastructure objects/subtrees, such as fabric access policies

Note For read operations to the managed objects that a user's credentials do not allow, a "DN/Class Not Found" error is returned, not "DN/Class Unauthorized to read." For write operations to a managed object that a user's credentials do not allow, an HTTP 401 Unauthorized error is returned. In the GUI, actions that a user's credentials do not allow, either they are not presented, or they are grayed out.

A set of predefined managed object classes can be associated with domains. These classes should not have overlapping containment. Examples of classes that support domain association are as follows:

- Layer 2 and Layer 3 network managed objects
- Network profiles (such as physical, Layer 2, Layer 3, management)
- QoS policies

When an object that can be associated with a domain is created, the user must assign domain(s) to the object within the limits of the user's access rights. Domain assignment can be modified at any time.

If a virtual machine management (VMM) domain is tagged as a security domain, the users contained in the security domain can access the correspondingly tagged VMM domain. For example, if a tenant named solar is tagged with the security domain called sun and a VMM domain is also tagged with the security domain called sun, then users in the solar tenant can access the VMM domain according to their access rights.

Access Rights Workflow Dependencies

The Cisco Application Centric Infrastructure (ACI) RBAC rules enable or restrict access to some or all of the fabric. For example, in order to configure a leaf switch for bare metal server access, the logged in administrator must have rights to the `infra` domain. By default, a tenant administrator does not have rights to the `infra` domain. In this case, a tenant administrator who plans to use a bare metal server connected to a leaf switch could not complete all the necessary steps to do so. The tenant administrator would have to coordinate with a fabric administrator who has rights to the `infra` domain. The fabric administrator would set up the switch configuration policies that the tenant administrator would use to deploy an application policy that uses the bare metal server attached to an ACI leaf switch.

AAA RBAC Roles and Privileges

The Application Policy Infrastructure Controller (APIC) provides the following AAA roles and privileges:

Note For each of the defined roles in Cisco APIC, the *APIC Roles and Privileges Matrix* shows which managed object classes can be written and which can be read. The matrix can be found at this URL: https://www.cisco.com/c/dam/en/us/td/docs/Website/datacenter/apicroles/roles.html

Role	Privilege	Description
aaa	aaa	Used for configuring authentication, authorization, accounting, and import/export policies.
admin	admin	Provides full access to all of the features of the fabric. The admin privilege can be considered to be a union of all other privileges.

Role: access-admin	
Privilege	**Description**
access-connectivity-l1	Used for Layer 1 configuration under infra. Example: selectors and port Layer 1 policy configurations.
access-connectivity-l2	Used for Layer 2 configuration under infra. Example: encap configurations on selectors, and attachable entity.
access-connectivity-l3	Used for Layer 3 configuration under infra and static route configurations under a tenant's L3Out.
access-connectivity-mgmt	Used for management infra policies.
access-connectivity-util	Used for tenant ERSPAN policies.
access-equipment	Used for access port configuration.
access-protocol-l1	Used for Layer 1 protocol configurations under infra.
access-protocol-l2	Used for Layer 2 protocol configurations under infra.

Role: access-admin	
Privilege	**Description**
access-protocol-l3	Used for Layer 3 protocol configurations under infra.
access-protocol-mgmt	Used for fabric-wide policies for NTP, SNMP, DNS, and image management.
access-protocol-ops	Used for operations-related access policies such as cluster policy and firmware policies.
access-qos	Used for changing CoPP and QoS-related policies.

Role:fabric-admin	
Privilege	**Description**
fabric-connectivity-l1	Used for Layer 1 configuration under the fabric. Example: selectors and port Layer 1 policy and vPC protection.
fabric-connectivity-l2	Used in firmware and deployment policies for raising warnings for estimating policy deployment impact.
fabric-connectivity-l3	Used for Layer 3 configuration under the fabric. Example: Fabric IPv4, IPv6, and MAC protection groups.
fabric-connectivity-mgmt	Used for atomic counter and diagnostic policies on leaf switches and spine switches.
fabric-connectivity-util	Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches.
fabric-equipment	Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches.
fabric-protocol-l1	Used for Layer 1 protocol configurations under the fabric.
fabric-protocol-l2	Used for Layer 2 protocol configurations under the fabric.
fabric-protocol-l3	Used for Layer 3 protocol configurations under the fabric.
fabric-protocol-mgmt	Used for fabric-wide policies for NTP, SNMP, DNS, and image management.
fabric-protocol-ops	Used for ERSPAN and health score policies.
fabric-protocol-util	Used for firmware management traceroute and endpoint tracking policies.
tenant-connectivity-util	Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches.
tenant-connectivity-l2	Used for Layer 2 connectivity changes, including bridge domains and subnets.

Role:fabric-admin	
Privilege	**Description**
tenant-connectivity-l3	Used for Layer 3 connectivity changes, including VRFs.
tenant-protocol-ops	Used for tenant traceroute policies.

Role	Privilege	Description
nw-svc-admin	nw-svc-device	Used for managing Layer 4 to Layer 7 service devices.
	nw-svc-devshare	Used for managing shared Layer 4 to Layer 7 service devices.
	nw-svc-policy	Used for managing Layer 4 to Layer 7 network service orchestration.
nw-svc-params	nw-svc-params	Used for managing Layer 4 to Layer 7 service policies.

Role: ops	
Privilege	**Description**
ops	Used for viewing the policies configured including troubleshooting policies. **Note** The **ops** role cannot be used for creating new monitoring and troubleshooting policies. Those policies need to be created by using the **admin** privilege, just like any other configurations in the Cisco APIC.

Role: read-all	
Privilege	**Description**
access-connectivity-l1	Used for Layer 1 configuration under infra. Example: selectors and port Layer 1 policy configurations.
access-connectivity-l2	Used for Layer 2 configuration under infra. Example: Encap configurations on selectors, and attachable entity.
access-connectivity-l3	Used for Layer 3 configuration under infra and static route configurations under a tenant's L3Out.
access-connectivity-mgmt	Used for management infra policies.
access-connectivity-util	Used for tenant ERSPAN policies.
access-equipment	Used for access port configuration.
access-protocol-l1	Used for Layer 1 protocol configurations under infra.

Role: read-all	
Privilege	**Description**
access-protocol-l2	Used for Layer 2 protocol configurations under infra.
access-protocol-l3	Used for Layer 3 protocol configurations under infra.
access-protocol-mgmt	Used for fabric-wide policies for NTP, SNMP, DNS, and image management.
access-protocol-ops	Used for operations-related access policies such as cluster policy and firmware policies.
access-qos	Used for changing CoPP and QoS-related policies.
fabric-connectivity-l1	Used for Layer 1 configuration under the fabric. Example: selectors and port Layer 1 policy and vPC protection.
fabric-connectivity-l2	Used in firmware and deployment policies for raising warnings for estimating policy deployment impact.
fabric-connectivity-l3	Used for Layer 3 configuration under the fabric. Example: Fabric IPv4, IPv6, and MAC protection groups.
fabric-protocol-l1	Used for Layer 1 protocol configurations under the fabric.
fabric-protocol-l2	Used for Layer 2 protocol configurations under the fabric.
fabric-protocol-l3	Used for Layer 3 protocol configurations under the fabric.
nw-svc-device	Used for managing Layer 4 to Layer 7 service devices.
nw-svc-devshare	Used for managing shared Layer 4 to Layer 7 service devices.
nw-svc-params	Used for managing Layer 4 to Layer 7 service policies.
nw-svc-policy	Used for managing Layer 4 to Layer 7 network service orchestration.
ops	Used for viewing the policies configured including troubleshooting policies. **Note** The **ops** role cannot be used for creating new monitoring and troubleshooting policies. Those policies need to be created by using the **admin** privilege, just like any other configurations in the Cisco APIC.
tenant-connectivity-util	Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches.
tenant-connectivity-l2	Used for Layer 2 connectivity changes, including bridge domains and subnets.
tenant-connectivity-l3	Used for Layer 3 connectivity changes, including VRFs.

Role: read-all	
Privilege	**Description**
tenant-connectivity-mgmt	Used for tenant in-band and out-of-band management connectivity configurations and for debugging/monitoring policies such as atomic counters and health score.
tenant-epg	Used for managing tenant configurations such as deleting/creating endpoint groups.
tenant-ext-connectivity-l1	Used for write access firmware policies.
tenant-ext-connectivity-l2	Used for managing tenant L2Out configurations.
tenant-ext-connectivity-l3	Used for managing tenant L3Out configurations.
tenant-ext-connectivity-mgmt	Used as write access for firmware policies.
tenant-ext-connectivity-util	Used for debugging/monitoring/observer policies such as traceroute, ping, oam, and eptrk.
tenant-ext-protocol-l1	Used for managing tenant external Layer 1 protocols. Generally only used for write access for firmware policies.
tenant-ext-protocol-l2	Used for managing tenant external Layer 2 protocols. Generally only used for write access for firmware policies.
tenant-ext-protocol-l3	Used for managing tenant external Layer 3 protocols such as BGP, OSPF, PIM, and IGMP.
tenant-ext-protocol-mgmt	Used as write access for firmware policies.
tenant-ext-protocol-util	Used for debugging/monitoring/observer policies such as traceroute, ping, oam, and eptrk.
tenant-network-profile	Used for managing tenant configurations, such as deleting and creating network profiles, and deleting and creating endpoint groups.
tenant-protocol-l1	Used for managing configurations for Layer 1 protocols under a tenant.
tenant-protocol-l2	Used for managing configurations for Layer 2 protocols under a tenant.
tenant-protocol-l3	Used for managing configurations for Layer 3 protocols under a tenant.
tenant-protocol-mgmt	Only used as write access for firmware policies.
tenant-protocol-ops	Used for tenant traceroute policies.
tenant-QoS	Used for QoS-related configurations for a tenant.
tenant-security	Used for contract-related configurations for a tenant.
vmm-connectivity	Used to read all the objects in Cisco APIC's VMM inventory required for virtual machine connectivity.

Role: read-all	
Privilege	**Description**
vmm-ep	Used to read virtual machine and hypervisor endpoints in the Cisco APIC's VMM inventory.
vmm-policy	Used for managing policies for virtual machine networking.
vmm-protocol-ops	Not used by VMM policies.
vmm-security	Used for managing authentication policies for VMM, such as the username and password for VMware vCenter.

Role: tenant-admin	
Privilege	**Description**
aaa	Used for configuring authentication, authorization, accouting and import/export policies.
access-connectivity-l1	Used for Layer 1 configuration under infra. Example: selectors and port Layer 1 policy configurations.
access-connectivity-l2	Used for Layer 2 configuration under infra. Example: Encap configurations on selectors, and attachable entity.
access-connectivity-l3	Used for Layer 3 configuration under infra and static route configurations under a tenant's L3Out.
access-connectivity-mgmt	Used for management infra policies.
access-connectivity-util	Used for tenant ERSPAN policies.
access-equipment	Used for access port configuration.
access-protocol-l1	Used for Layer 1 protocol configurations under infra.
access-protocol-l2	Used for Layer 2 protocol configurations under infra.
access-protocol-l3	Used for Layer 3 protocol configurations under infra.
access-protocol-mgmt	Used for fabric-wide policies for NTP, SNMP, DNS, and image management.
access-protocol-ops	Used for operations-related access policies such as cluster policy and firmware policies.
access-qos	Used for changing CoPP and QoS-related policies.
fabric-connectivity-l1	Used for Layer 1 configuration under the fabric. Example: selectors and port Layer 1 policy and vPC protection.
fabric-connectivity-l2	Used in firmware and deployment policies for raising warnings for estimating policy deployment impact.

Role: tenant-admin	
Privilege	**Description**
fabric-connectivity-l3	Used for Layer 3 configuration under the fabric. Example: Fabric IPv4, IPv6, and MAC protection groups.
fabric-connectivity-mgmt	Used for atomic counter and diagnostic policies on leaf switches and spine switches.
fabric-connectivity-util	Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches.
fabric-equipment	Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches.
fabric-protocol-l1	Used for Layer 1 protocol configurations under the fabric.
fabric-protocol-l2	Used for Layer 2 protocol configurations under the fabric.
fabric-protocol-l3	Used for Layer 3 protocol configurations under the fabric.
fabric-protocol-mgmt	Used for fabric-wide policies for NTP, SNMP, DNS, and image management.
fabric-protocol-ops	Used for ERSPAN and health score policies.
fabric-protocol-util	Used for firmware management traceroute and endpoint tracking policies.
nw-svc-device	Used for managing Layer 4 to Layer 7 service devices.
nw-svc-devshare	Used for managing shared Layer 4 to Layer 7 service devices.
nw-svc-params	Used for managing Layer 4 to Layer 7 service policies.
nw-svc-policy	Used for managing Layer 4 to Layer 7 network service orchestration.
ops	Used for viewing the policies configured including troubleshooting policies. **Note** The **ops** role cannot be used for creating new monitoring and troubleshooting policies. Those policies need to be created by using the **admin** privilege, just like any other configurations in the Cisco APIC.
tenant-connectivity-util	Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches.
tenant-connectivity-l2	Used for Layer 2 connectivity changes, including bridge domains and subnets.
tenant-connectivity-l3	Used for Layer 3 connectivity changes, including VRFs.

Role: tenant-admin	
Privilege	**Description**
tenant-connectivity-mgmt	Used for tenant in-band and out-of-band management connectivity configurations and for debugging/monitoring policies such as atomic counters and health score.
tenant-epg	Used for managing tenant configurations such as deleting/creating endpoint groups.
tenant-ext-connectivity-l1	Used for write access firmware policies.
tenant-ext-connectivity-l2	Used for managing tenant L2Out configurations.
tenant-ext-connectivity-l3	Used for managing tenant L3Out configurations.
tenant-ext-connectivity-mgmt	Used as write access for firmware policies.
tenant-ext-connectivity-util	Used for debugging/monitoring/observer policies such as traceroute, ping, oam, and eptrk.
tenant-ext-protocol-l1	Used for managing tenant external Layer 1 protocols. Generally only used for write access for firmware policies.
tenant-ext-protocol-l2	Used for managing tenant external Layer 2 protocols. Generally only used for write access for firmware policies.
tenant-ext-protocol-l3	Used for managing tenant external Layer 3 protocols such as BGP, OSPF, PIM, and IGMP.
tenant-ext-protocol-mgmt	Used as Write access for firmware policies.
tenant-ext-protocol-util	Used for debugging/monitoring/observer policies such as traceroute, ping, oam, and eptrk.
tenant-network-profile	Used for managing tenant configurations, such as deleting and creating network profiles, and deleting and creating endpoint groups.
tenant-protocol-l1	Used for managing configurations for Layer 1 protocols under a tenant.
tenant-protocol-l2	Used for managing configurations for Layer 2 protocols under a tenant.
tenant-protocol-l3	Used for managing configurations for Layer 3 protocols under a tenant.
tenant-protocol-mgmt	Only used as write access for firmware policies.
tenant-protocol-ops	Used for tenant traceroute policies.
tenant-QoS	Used for QoS-related configurations for a tenant.
tenant-security	Used for contract-related configurations for a tenant.
vmm-connectivity	Used to read all the objects in Cisco APIC's VMM inventory required for virtual machine connectivity.

Role: tenant-admin	
Privilege	**Description**
vmm-ep	Used to read virtual machine and hypervisor endpoints in the Cisco APIC's VMM inventory.
vmm-policy	Used for managing policies for virtual machine networking.
vmm-protocol-ops	Not used by VMM policies.
vmm-security	Used for managing authentication policies for VMM, such as the username and password for VMware vCenter.

Role: tenant-ext-admin	
Privilege	**Description**
tenant-connectivity-util	Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches.
tenant-connectivity-l2	Used for Layer 2 connectivity changes, including bridge domains and subnets.
tenant-connectivity-l3	Used for Layer 3 connectivity changes, including VRFs.
tenant-connectivity-mgmt	Used for tenant in-band and out-of-band management connectivity configurations and for debugging/monitoring policies such as atomic counters and health score.
tenant-epg	Used for managing tenant configurations such as deleting/creating endpoint groups.
tenant-ext-connectivity-l1	Used for write access firmware policies.
tenant-ext-connectivity-l2	Used for managing tenant L2Out configurations.
tenant-ext-connectivity-l3	Used for managing tenant L3Out configurations.
tenant-ext-connectivity-mgmt	Used as write access for firmware policies.
tenant-ext-connectivity-util	Used for debugging/monitoring/observer policies such as traceroute, ping, oam, and eptrk.
tenant-ext-protocol-l1	Used for managing tenant external Layer 1 protocols. Generally only used for write access for firmware policies.
tenant-ext-protocol-l2	Used for managing tenant external Layer 2 protocols. Generally only used for write access for firmware policies.
tenant-ext-protocol-l3	Used for managing tenant external Layer 3 protocols such as BGP, OSPF, PIM, and IGMP.
tenant-ext-protocol-mgmt	Used as Write access for firmware policies.

Role: tenant-ext-admin	
Privilege	**Description**
tenant-ext-protocol-util	Used for debugging/monitoring/observer policies such as traceroute, ping, oam, and eptrk.
tenant-network-profile	Used for managing tenant configurations, such as deleting and creating network profiles, and deleting and creating endpoint groups.
tenant-protocol-l1	Used for managing configurations for Layer 1 protocols under a tenant.
tenant-protocol-l2	Used for managing configurations for Layer 2 protocols under a tenant.
tenant-protocol-l3	Used for managing configurations for Layer 3 protocols under a tenant.
tenant-protocol-mgmt	Only used as write access for firmware policies.
tenant-protocol-ops	Used for tenant traceroute policies.
tenant-QoS	Used for QoS-related configurations for a tenant.
tenant-security	Used for contract-related configurations for a tenant.
vmm-connectivity	Used to read all the objects in Cisco APIC's VMM inventory required for virtual machine connectivity.
vmm-ep	Used to read virtual machine and hypervisor endpoints in the Cisco APIC's VMM inventory.
vmm-policy	Used for managing policies for virtual machine networking.
vmm-protocol-ops	Not used by VMM policies.
vmm-security	Used for managing authentication policies for VMM, such as the username and password for VMware vCenter.

Role: vmm-admin	
Privilege	**Description**
vmm-connectivity	Used to read all the objects in Cisco APIC's VMM inventory required for virtual machine connectivity.
vmm-ep	Used to read virtual machine and hypervisor endpoints in the Cisco APIC's VMM inventory.
vmm-policy	Used for managing policies for virtual machine networking.
vmm-protocol-ops	Not used by VMM policies.
vmm-security	Used for managing authentication policies for a VMM, such as the username and password for VMware vCenter.

Custom Roles

You can create custom roles and assign privileges to the roles. The interface internally assigns one or more privileges to all managed object classes. In an XML model, privileges are assigned in an access attribute. Privilege bits are assigned at compile time and apply per class, and not per instance or object of the class.

In addition to the 45 privilege bits, the "aaa" privilege bit applies to all AAA-subsystem configuration and read operations. The following table provides a matrix of the supported privilege combinations. The rows in the table represent Cisco Application Centric Infrastructure (ACI) modules and the columns represent functionality for a given module. A value of "Yes" in a cell indicates that the functionality for the module is accessible and there exists a privilege bit to access that functionality. An empty cell indicates that the particular functionality for module is not accessible by any privilege bit. See the privilege bit descriptions to learn what each bit does.

	Connectivity	QoS	Security	Application	Fault	Stats	Provider	Service Profile	Service Chain
VMM	Yes		Yes		Yes	Yes	Yes		
Fabric	Yes	Yes	Yes	Yes	Yes	Yes	Yes		
External	Yes	Yes	Yes		Yes	Yes			Yes
Tenant	Yes	Yes	Yes	EPG, NP	Yes	Yes			Yes
Infra	Yes	Yes	Yes	Yes	Yes	Yes			Yes
Ops					Yes	Yes			
Storage	Yes	Yes	Yes	Yes	Yes	Yes			
Network Service	Yes	Yes	Yes	Yes	Yes	Yes		Yes	

Selectively Expose Physical Resources across Security Domains

A fabric-wide administrator uses RBAC rules to selectively expose physical resources to users that otherwise are inaccessible because they are in a different security domain.

For example, if a user in tenant Solar needs access to a virtual machine management (VMM) domain, the fabric-wide admin could create an RBAC rule to allow this. The RBAC rule is comprised of these two parts: the distinguished name (DN) that locates the object to be accessed plus the name of the security domain that contains the user who will access the object. So, in this example, when designated users in the security domain Solar are logged in, this rule gives them access to the VMM domain as well as all its child objects in the tree. To give users in multiple security domains access to the VMM domain, the fabric-wide administrator would create an RBAC rule for each security domain that contains the DN for the VMM domain plus the security domain.

Note While an RBAC rule exposes an object to a user in a different part of the management information tree, it is not possible to use the CLI to navigate to such an object by traversing the structure of the tree. However, as long as the user knows the DN of the object included in the RBAC rule, the user can use the CLI to locate it via an MO find command.

Enable Sharing of Services across Security Domains

A fabric-wide administrator uses RBAC rules to provision trans-tenant EPG communications that enable shared services across tenants.

APIC Local Users

An administrator can choose not to use external AAA servers but rather configure users on the APIC itself. These users are called APIC-local users.

At the time a user sets their password, the APIC validates it against the following criteria:

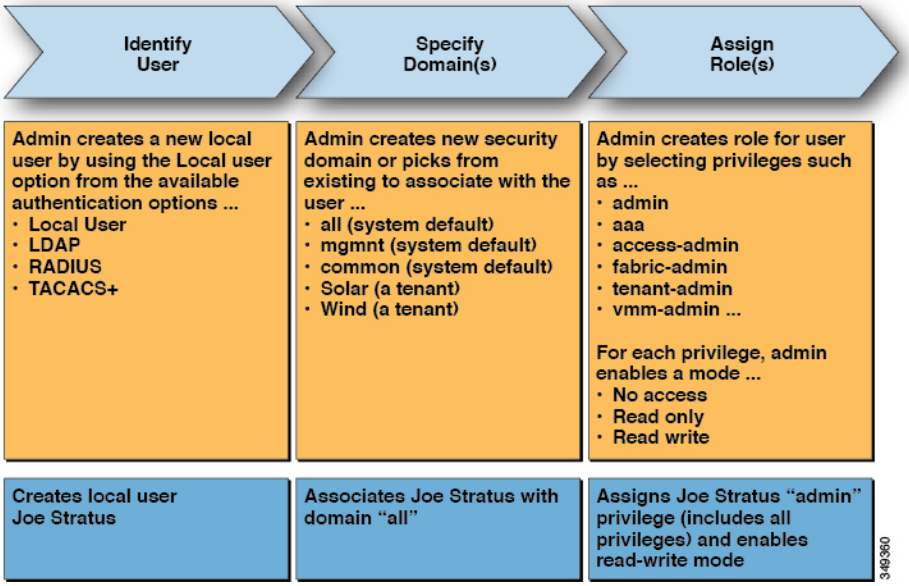
- Minimum password length is 8 characters.
- Maximum password length is 64 characters.
- Has fewer than three consecutive repeated characters.
- Must have characters from at least three of the following characters types: lowercase, uppercase, digit, symbol.
- Does not use easily guessed passwords.
- Cannot be the username or the reverse of the username.
- Cannot be any variation of cisco, isco or any permutation of these characters or variants obtained by changing the capitalization of letters therein.

Cisco ACI uses a crypt library with a SHA256 one-way hash for storing passwords. At rest hashed passwords are stored in an encrypted filesystem. The key for the encrypted filesystem is protected using the Trusted Platform Module (TPM).

The APIC also enables administrators to grant access to users configured on externally managed authentication Lightweight Directory Access Protocol (LDAP), RADIUS, TACACS+, or SAML servers. Users can belong to different authentication systems and can log in simultaneously to the APIC.

The following figure shows how the process works for configuring an admin user in the local APIC authentication database who has full access to the entire ACI fabric.

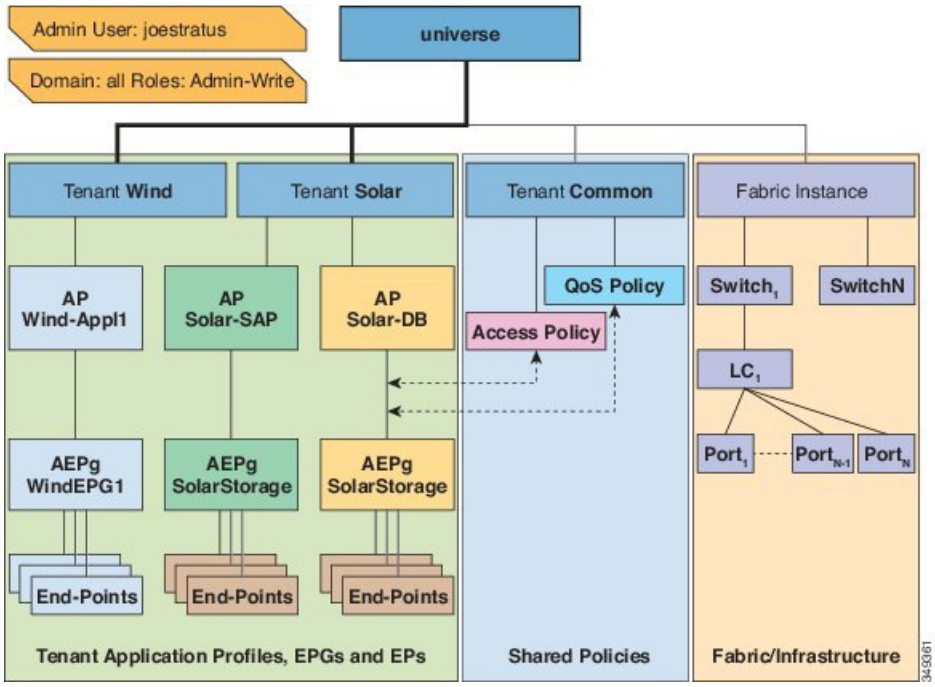
Figure 1: APIC Local User Configuration Process



Note The security domain "all" represents the entire Managed Information Tree (MIT). This domain includes all policies in the system and all nodes managed by the APIC. Tenant domains contain all the users and managed objects of a tenant. Tenant administrators should not be granted access to the "all" domain.

The following figure shows the access that the admin user Joe Stratus has to the system.

Figure 2: Result of Configuring Admin User for "all" Domain



The user Joe Stratus with read-write "admin" privileges is assigned to the domain "all" which gives him full access to the entire system.

OTP-Based Two-Factor Authentication for Local Users

A fabric admin user can enable the one-time password (OTP) feature for a local user. A one-time password changes every 30 seconds for enhanced security. After the admin enables OTP, Cisco Application Policy Infrastructure Controller (APIC) generates a random human-readable 16 binary octet that is a base32 OTP key. This OTP key is used to generate the OTP for the user, which is used for two-factor authentication.

Cisco APIC supports the following security platforms for use with two-factor authentication:

- Duo Security, with the Duo Mobile app
- Google, with the Google Authenticator app (only with Android and Apple iOS smartphones)

Note You must download the indicated app from the appropriate app store.

These security platforms do not act as a repository for user identities. The platforms offer two-factor authentication on top of an organization's existing authentication, which could be on-premises or cloud-based. Two-factor authentication occurs once the user has finished the authentication with the organization's primary authentication source.

The platforms support three types of two-factor authentication methods after you complete authentication with the primary authentication source:

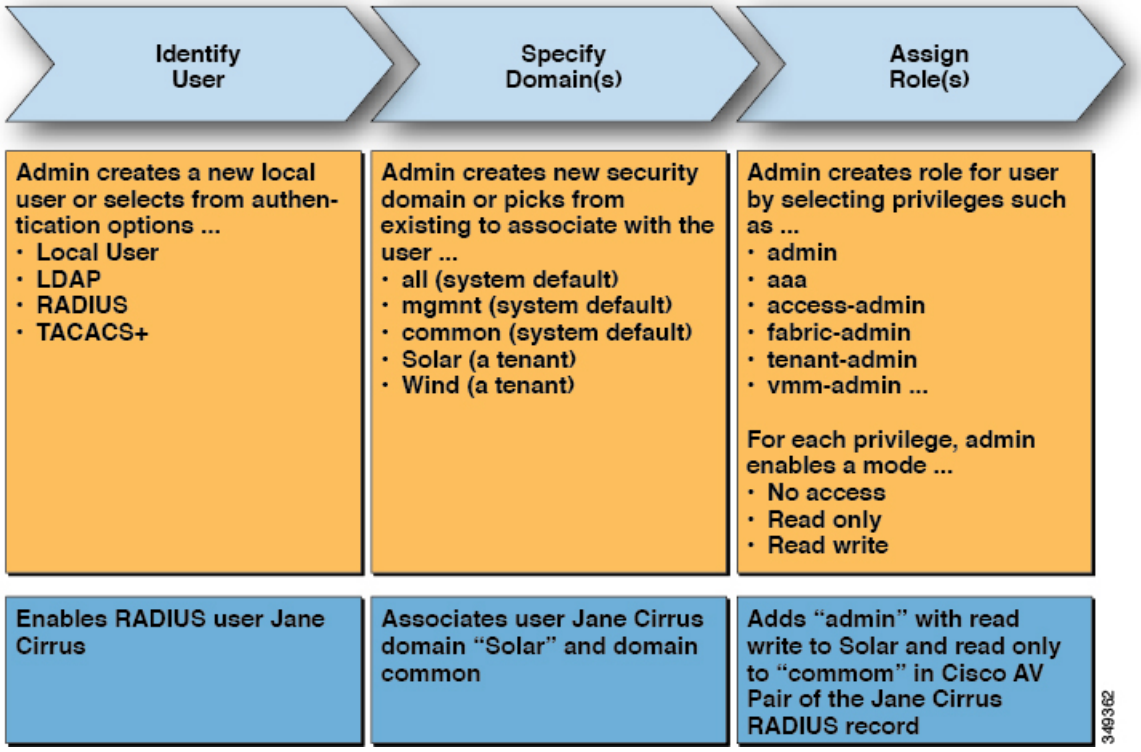
- Notification push on mobile using the appropriate mobile app on smartphones.
- Phone call on your registered phone or mobile numbers.

- Passcode that is generated on the appropriate mobile app.

Externally Managed Authentication Server Users

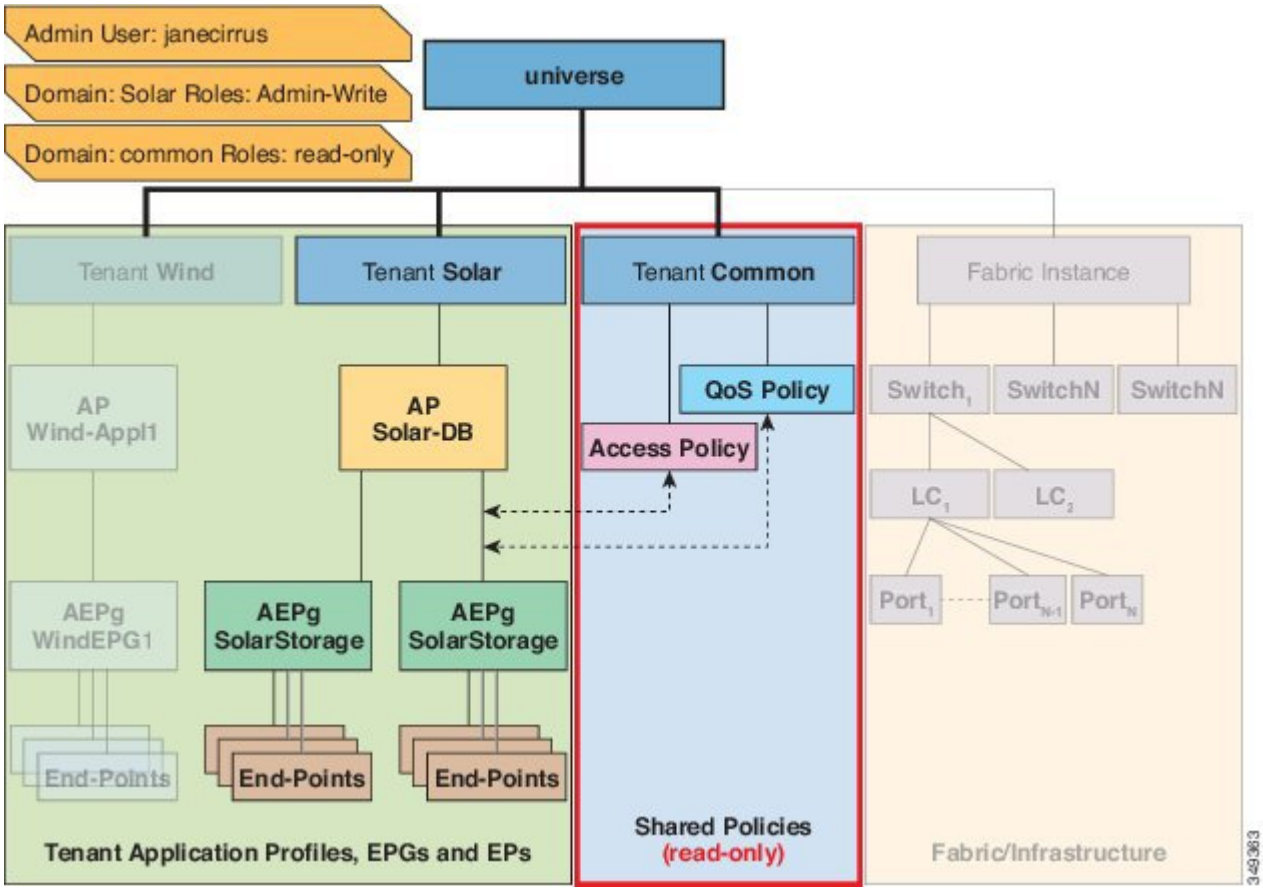
The following figure shows how the process works for configuring an admin user in an external RADIUS server who has full access to the tenant Solar.

Figure 3: Process for Configuring Users on External Authentication Servers



The following figure shows the access the admin user Jane Cirrus has to the system.

Figure 4: Result of Configuring Admin User for Tenant Solar



In this example, the Solar tenant administrator has full access to all the objects contained in the Solar tenant as well as read-only access to the tenant Common. Tenant admin Jane Cirrus has full access to the tenant Solar, including the ability to create new users in tenant Solar. Tenant users are able to modify configuration parameters of the ACI fabric that they own and control. They also are able to read statistics and monitor faults and events for the entities (managed objects) that apply to them such as endpoints, endpoint groups (EPGs) and application profiles.

In the example above, the user Jane Cirrus was configured on an external RADIUS authentication server. To configure an AV Pair on an external authentication server, add a Cisco AV Pair to the existing user record. The Cisco AV Pair specifies the Role-Based Access Control (RBAC) roles and privileges for the user on the APIC. The RADIUS server then propagates the user privileges to the APIC controller.

In the example above, the configuration for an open radius server (/etc/raddb/users) is as follows:

```
janecirrus Cleartext-Password := "<password>"
Cisco-avpair = "shell:domains = solar/admin/,common//read-all(16001)"
```

This example includes the following elements:

- `janecirrus` is the tenant administrator
- `solar` is the tenant
- `admin` is the role with write privileges

- `common` is the tenant-common subtree that all users should have read-only access to
- `read-all` is the role with read privileges

Cisco AV Pair Format

The Cisco APIC requires that an administrator configure a Cisco AV Pair on an external authentication server and only looks for one AV pair string. To do so, an administrator adds a Cisco AV pair to the existing user record. The Cisco AV pair specifies the APIC required RBAC roles and privileges for the user.

In order for the AV pair string to work, it must be formatted as follows:

```
shell:domains =
ACI_Security_Domain_1/ACI_Write_Role_1|ACI_Write_Role_2|ACI_Write_Role_3/ACI_Read_Role_1|ACI_Read_Role_2,
ACI_Security_Domain_2/ACI_Write_Role_1|ACI_Write_Role_2|ACI_Write_Role_3/ACI_Read_Role_1|ACI_Read_Role_2,
ACI_Security_Domain_3/ACI_Write_Role_1|ACI_Write_Role_2|ACI_Write_Role_3/ACI_Read_Role_1|ACI_Read_Role_2
```

- **shell:domains=** - Required so that ACI reads the string correctly. This must always prepend the shell string.
- **ACI_Security_Domain_1//admin** - Grants admin read only access to the tenants in this security domain.
- **ACI_Security_Domain_2/admin** - Grants admin write access to the tenants in this security domain.
- **ACI_Security_Domain_3/read-all** - Grants read-all write access to the tenants in this security domain.

Note /'s separate the security domain, write, read sections of the string. |'s separate multiple write or read roles within the same security domain.

Note Starting with Cisco APIC release 2.1, if no UNIX ID is provided in AV Pair, the APIC allocates the unique UNIX user ID internally.

The APIC supports the following regexes:

```
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})$
```

Examples:

- Example 1: A Cisco AV Pair that contains a single Login domain with only writeRoles:

```
shell:domains=ACI_Security_Domain_1/Write_Role_1|Write_Role_2/
```

- Example 2: A Cisco AV Pair that contains a single Login domain with only readRoles:

```
shell:domains=Security_Domain_1//Read_Role_1|Read_Role_2
```

Note The "/" character is a separator between writeRoles and readRoles per Login domain and is required even if only one type of role is to be used.

The Cisco AVpair string is case sensitive. Although a fault may not be seen, using mismatching cases for the domain name or roles could lead to unexpected privileges being given.

AV Pair GUI Configuration

The security domain is defined in the ACI GUI under **Admin** > **AAA** > **Security Management** > **Security Domains** and assigned to a tenant under **Tenants** > **Tenant_Name** > **Policy**.

A security domain must have either a read or write role. These roles are defined in **APIC** > **Admin** > **Security Management** > **Roles**. If a role is input into the write section it automatically grants read privileges of the same level so there is no need to have ACI_Security_Domain_1/admin/admin.

Change Remote User Role

User-privileges can be modified “dynamically”, which allows the user to request for a role-change, and is allowed or denied the requested role based on information stored locally or remotely.

The role-change is only supported through the Cisco ACS server and can be done by role assignment based on explicit "request".

The ACI fabric supports external authentication using Radius, TACACS+ and LDAP protocols. Both the above-mentioned methods assume that the remote authentication server has components to support the role-change functionality.

The Cisco Secure ACS server provides the remote authentication, authorization and accounting features for the TACACS+ protocol.

Rules are matched, either with **Default Device Admin** or **Default Network Access Service**.

In the Authorization, another set of rules are configured:

- **AVPairOps**: matches the tacacs+ username and AVPair value (cisco-av-pair*newrole). If the rule matches, the ACI_OPS shell-profile is returned
- **NoAVPair**: matches only the tacacs+ username and return ACI_ADMIN shell profile on match
- **opsuser**: matches only the protocol and returns ACI_OPS shell profile

Change the Remote User Role Using the GUI

Before you begin

Roles must first be configured on the Cisco ASC Server to match the AVPairs and selected shell-authorization-profile based on the match.

Procedure

Step 1 Create an ASC Authorization Policy navigate to **Access Policies** > **Access Services** > **Default Device Admin Identity** and perform the following steps:

Note
Shell Profile is configured with CiscoAVPair , which is used to Authorize the User.

a) Add the condition to **TACACS+:AVPair equals cisco-av-pair*** and click **OK**.

Note
The user is authorized with the **cisco-av-pair** role by default.

b) Add the condition to **TACACS+:AVPair equals cisco-av-pair*readall** and click **OK**.

Note
The keyword **readall** is used in APIC to change the Role from **default** Role to **readall** Role (read-all is configured in Shell-Profile).

Step 2 Log in to the APIC GUI, click the **welcome, <login_name>** drop-down list and choose Change Remote User Role.

Step 3 In the Change Remote User Role dialog box, enter the information in the **User Name**, **Password**, and **New Role** fields and click **Submit**.

The GUI will refresh with the new role applied.

Note
To return to the parent role, open the Change Remote User Role dialog box again and enter the information for **User Name** and **Password** but leave the **New Role** field blank.

Change the Remote User Role Using REST API

Before you begin

Roles must first be configured on the Cisco ASC Server to match the AVPairs and selected shell-authorization-profile based on the match.

The user logs in with the user-name **apicadmin** and password.

Procedure

Step 1 Change to a new role:

Example:

```
<!--   api/requestNewRole/json   -->
<aaaChangeRole>
<attributes userName="apic#tacacs” apicadmin="pwd Ins3965!” role=“newrole"/>
```

Step 2 Return to the original role:

Example:

```
<!--   api/requestNewRole/json   -->
<aaaChangeRole>
<attributes userName="apic#tacacs” apicadmin="pwd Ins3965!” role=“"/>
```

About Signature-Based Transactions

The APIC controllers in a Cisco ACI fabric offer different methods to authenticate users.

The primary authentication method uses a username and password and the APIC REST API returns an authentication token that can be used for future access to the APIC. This may be considered insecure in a situation where HTTPS is not available or enabled.

Another form of authentication that is offered utilizes a signature that is calculated for every transaction. The calculation of that signature uses a private key that must be kept secret in a secure location. When the APIC receives a request with a signature rather than a token, the APIC utilizes an X.509 certificate to verify the signature. In signature-based authentication, every transaction to the APIC must have a newly calculated signature. This is not a task that a user should do manually for each transaction. Ideally this function should be utilized by a script or an application that communicates with the APIC. This method is the most secure as it requires an attacker to crack the RSA/DSA key to forge or impersonate the user credentials.

Note Additionally, you must use HTTPS to prevent replay attacks.

Before you can use X.509 certificate-based signatures for authentication, verify that the following pre-requisite tasks are completed:

1. Create an X.509 certificate and private key using OpenSSL or a similar tool.
2. Create a local user on the APIC. (If a local user is already available, this task is optional).
3. Add the X.509 certificate to the local user on the APIC.

Guidelines and Limitations

Follow these guidelines and limitations:

- Local users are supported. Remote AAA users are not supported.
- The APIC GUI does not support the certificate authentication method.
- WebSockets and eventchannels do not work for X.509 requests.
- Certificates signed by a third party are not supported. Use a self-signed certificate.

Accounting

ACI fabric accounting is handled by these two managed objects (MO) that are processed by the same mechanism as faults and events:

- The `aaaSessionLR` MO tracks user account login and logout sessions on the APIC and switches, and token refresh. The ACI fabric session alert feature stores information such as the following:
  - Username
  - IP address initiating the session
  - Type (telnet, https, REST etc.)
  - Session time and length

- Token refresh – a user account login event generates a valid active token which is required in order for the user account to exercise its rights in the ACI fabric.

Note Token expiration is independent of login; a user could log out but the token expires according to the duration of the timer value it contains.

- The `aaaModLR` MO tracks the changes users make to objects and when the changes occurred.
- If the AAA server is not pingable, it is marked unavailable and a fault is seen.

Both the `aaaSessionLR` and `aaaModLR` event logs are stored in APIC shards. Once the data exceeds the pre-set storage allocation size, it overwrites records on a first-in first-out basis.

Note In the event of a destructive event such as a disk crash or a fire that destroys an APIC cluster node, the event logs are lost; event logs are not replicated across the cluster.

The `aaaModLR` and `aaaSessionLR` MOs can be queried by class or by distinguished name (DN). A class query provides all the log records for the whole fabric. All `aaaModLR` records for the whole fabric are available from the GUI at the **Fabric** > **Inventory** > **POD** > **History** > **Audit Log** section, The APIC GUI **History** > **Audit Log** options enable viewing event logs for a specific object identified in the GUI.

The standard syslog, callhome, REST query, and CLI export mechanisms are fully supported for `aaaModLR` and `aaaSessionLR` MO query data. There is no default policy to export this data.

There are no pre-configured queries in the APIC that report on aggregations of data across a set of objects or for the entire system. A fabric administrator can configure export policies that periodically export `aaaModLR` and `aaaSessionLR` query data to a syslog server. Exported data can be archived periodically and used to generate custom reports from portions of the system or across the entire set of system logs.

Routed Connectivity to External Networks as a Shared Service Billing and Statistics

The APIC can be configured to collect byte count and packet count billing statistics from a port configured for routed connectivity to external networks (an `l3extInstP` EPG) as a shared service. Any EPG in any tenant can share an `l3extInstP` EPG for routed connectivity to external networks. Billing statistics can be collected for each EPG in any tenant that uses an `l3extInstP` EPG as a shared service. The leaf switch where the `l3extInstP` is provisioned forwards the billing statistics to the APIC where they are aggregated. Accounting policies can be configured to periodically export these billing statics to a server.

Configuration

Configuring a Local User

In the initial configuration script, the admin account is configured and the admin is the only user when the system starts. The APIC supports a granular, role-based access control system where user accounts can be created with various roles including non-admin users with fewer privileges.

Configuring a Local User Using the GUI

Before you begin

- The ACI fabric is installed, APIC controllers are online, and the APIC cluster is formed and healthy.
- As appropriate, the security domain(s) that the user will access are defined. For example, if the new use account will be restricted to accessing a tenant, the tenant domain is tagged accordingly.
- An APIC user account is available that will enable the following:
  - Creating the TACACS+ provider.
  - Creating the local user account in the target security domain(s). If the target domain is `all`, the login account used to create the new local user must be a fabric-wide administrator that has access to `all`. If the target domain is a tenant, the login account used to create the new local user must be a tenant administrator that has full read write access rights to the target tenant domain.

Procedure

Step 1 On the menu bar, choose **ADMIN** > **AAA**.

Step 2 In the **Navigation** pane, click **Users** and **Local Users** in the **Work** pane.

Step 3 In the **Work** pane, verify that you are in the **Local Users** tab.

The admin user is present by default

Step 4 In the **Work** pane, click on task icon drop-down list and select **Create Local User**.

Step 5 In the **User Identity** dialog box, perform the following actions:

a) In the **Login ID** field, add an ID.

b) In the **Password** field, enter the password.

At the time a user sets their password, the APIC validates it against the following criteria:

c) In the **Confirm Password** field, confirm the password.

d) (Optional) For Certificate based authentication, in the **User Certificate Attribute** field, enter the user identity from the authentication certificate.

e) Click **Finish**.

Step 6 In the **Security** dialog box, choose the desired security domain for the user, and click **Next**.

Step 7 In the **Roles** dialog box, click the radio buttons to choose the roles for your user, and click **Next**.

You can provide read-only or read/write privileges.

Step 8 In the **Navigation** pane, click the name of the user that you created. In the **Work** pane, expand the **+** sign next to your user in the **Security Domains** area.
The access privileges for your user are displayed.

Configuring SSH Public Key Authentication Using the GUI

Before you begin

- Create a local user account in the target security domain(s). If the target domain is `all`, the login account used to create the new local user must be a fabric-wide administrator that has access to `all`. If the target domain is a tenant, the login account used to create the new local user must be a tenant administrator that has full read write access rights to the target tenant domain.
- Generate a public key using the Unix command **ssh-keygen**.

  The default login domain must be set to **local**

Procedure

Step 1 On the menu bar, choose **ADMIN** > **Users** and confirm you are in the **Local Users** tab.

Step 2 In the **Navigation** pane, click the name of the user that you previously created.

Step 3 In the **Work** pane, expand the **SSH Keys** table, and insert the following information:

a) In the **Name** field, enter a name for the key.

b) In the **Key** field, insert the public key previously created. Click **Update**.

Note
To create the SSH Private Key File for downloading to a remote location then in the menu bar, expand **Firmware** > **Download Tasks**.

Configuring a Local User Using the NX-OS Style CLI

Procedure

Step 1 In the NX-OS CLI, start in configuration mode, shown as follows:

Example:

```
apic1# configure
apic1(config)#
```

Step 2 Create a new user, shown as follows:

Example:

```
apic1(config)# username
 WORD       User name (Max Size 28)
 admin
 cli-user
 jigarshah
 test1
 testUser
```

```
apic1(config)# username test
apic1(config-username)#
 account-status      Set The status of the locally-authenticated user account.
 certificate         Create AAA user certificate in X.509 format.
 clear-pwd-history   Clears the password history of a locally-authenticated user
 domain              Create the AAA domain to which the user belongs.
 email               Set The email address of the locally-authenticated user.
 exit                Exit from current mode
 expiration          If expires enabled, Set expiration date of locally-authenticated user
 account.
 expires             Enable expiry for locally-authenticated user account
 fabric              show fabric related information
 first-name          Set the first name of the locally-authenticated user.
 last-name           Set The last name of the locally-authenticated user.
 no                  Negate a command or set its defaults
 password            Set The system user password.
 phone               Set The phone number of the locally-authenticated user.
 pwd-lifetime        Set The lifetime of the locally-authenticated user password.
 pwd-strength-check  Enforces the strength of the user password
 show                Show running system information
 ssh-key             Update ssh key for the user for ssh authentication
 where               show the current mode

apic1(config-username)# exit
```

Configuring a Local User Using the REST API

Procedure

Create a local user.

Example:

```
URL: https://apic-ip-address/api/node/mo/uni/userext.xml
POST CONTENT:
    <aaaUser name="operations" phone="" pwd="<strong_password>" >
        <aaaUserDomain childAction="" descr="" name="all" rn="userdomain-all" status="">
        <aaaUserRole childAction="" descr="" name="Ops" privType="writePriv"/>
        </aaaUserDomain>
    </aaaUser>
```

Generating an X.509 Certificate and a Private Key

Procedure

Step 1 Enter an OpenSSL command to generate an X.509 certificate and private key.

Example:

```
$ openssl req -new -newkey rsa:1024 -days 36500 -nodes -x509 -keyout userabc.key -out
userabc.crt -subj '/CN=User ABC/O=Cisco Systems/C=US'
```

Note

- Once the X.509 certificate is generated, it will be added to the users profile on the APIC, and it is used to verify signatures. The private key is used by the client to generate the signatures.
- The certificate contains a public key but not the private key. The public key is the primary information used by the APIC to verify the calculated signature. The private key is never stored on the APIC. You must keep it secret.

Step 2 Display the fields in the certificate using OpenSSL.

Example:

```
$ openssl x509 -text -in userabc.crt
        Certificate:
            Data:
                Version: 3 (0x2)
                Serial Number:
                    c4:27:6c:4d:69:7c:d2:b6
                Signature Algorithm: sha1WithRSAEncryption
                Issuer: CN=User ABC, O=Cisco Systems, C=US
                Validity
                    Not Before: Jan 12 16:36:14 2015 GMT
                    Not After : Dec 19 16:36:14 2114 GMT
                Subject: CN=User ABC, O=Cisco Systems, C=US
                Subject Public Key Info:
                    Public Key Algorithm: rsaEncryption
                    RSA Public Key: (1024 bit)
                        Modulus (1024 bit):
                            00:92:35:12:cd:2b:78:ef:9d:ca:0e:11:77:77:3a:
                            99:d3:25:42:94:b5:3e:8a:32:55:ce:e9:21:2a:ff:
                            e0:e4:22:58:6d:40:98:b1:0d:42:21:db:cd:44:26:
                            50:77:e5:fa:b6:10:57:d1:ec:95:e9:86:d7:3c:99:
                            ce:c4:7f:61:1d:3c:9e:ae:d8:88:be:80:a0:4a:90:
                            d2:22:e9:1b:25:27:cd:7d:f3:a5:8f:cf:16:a8:e1:
                            3a:3f:68:0b:9c:7c:cb:70:b9:c7:3f:e8:db:85:d8:
                            98:f6:e3:70:4e:47:e2:59:03:49:01:83:8e:50:4a:
                            5f:bc:35:d2:b1:07:be:ec:e1
                        Exponent: 65537 (0x10001)
                X509v3 extensions:
                    X509v3 Subject Key Identifier:
                        0B:E4:11:C7:23:46:10:4F:D1:10:4C:C1:58:C2:1E:18:E8:6D:85:34
                    X509v3 Authority Key Identifier:
                        keyid:0B:E4:11:C7:23:46:10:4F:D1:10:4C:C1:58:C2:1E:18:E8:6D:85:34
                        DirName:/CN=User ABC/O=Cisco Systems/C=US
                        serial:C4:27:6C:4D:69:7C:D2:B6

                    X509v3 Basic Constraints:
                        CA:TRUE
            Signature Algorithm: sha1WithRSAEncryption
                8f:c4:9f:84:06:30:59:0c:d2:8a:09:96:a2:69:3d:cf:ef:79:
                91:ea:cd:ae:80:16:df:16:31:3b:69:89:f7:5a:24:1f:fd:9f:
                d1:d9:b2:02:41:01:b9:e9:8d:da:a8:4c:1e:e5:9b:3e:1d:65:
                84:ff:e8:ad:55:3e:90:a0:a2:fb:3e:3e:ef:c2:11:3d:1b:e6:
                f4:5e:d2:92:e8:24:61:43:59:ec:ea:d2:bb:c9:9a:7a:04:91:
                8e:91:bb:9d:33:d4:28:b5:13:ce:dc:fe:c3:e5:33:97:5d:37:
                cc:5f:ad:af:5a:aa:f4:a3:a8:50:66:7d:f4:fb:78:72:9d:56:
```

```
            91:2c
[snip]
```

Creating a Local User and Adding a User Certificate Using the GUI

Procedure

Step 1 On the menu bar, choose **ADMIN** > **AAA**.

Step 2 In the **Navigation** pane, click **Users** and **Local Users** in the **Work** pane.

Step 3 In the **Work** pane, verify that you in the **Local Users** tab.

The admin user is present by default

Step 4 In the **Work** pane, click on task icon drop-down list and select **Create Local User**.

Step 5 In the **Security** dialog box, choose the desired security domain for the user, and click **Next**.

Step 6 In the **Roles** dialog box, click the radio buttons to choose the roles for your user, and click **Next**.

You can provide read-only or read/write privileges.

Step 7 In the **User Identity** dialog box, perform the following actions:

a) In the **Login ID** field, add an ID.
b) In the **Password** field, enter the password.
c) In the **Confirm Password** field, confirm the password.
d) (Optional) For Certificate based authentication, in the **User Certificate Attribute** field, enter the user identity from the authentication certificate.
e) Click **Finish**.

Step 8 In the **Navigation** pane, click the name of the user that you created. In the **Work** pane, expand the + sign next to your user in the **Security Domains** area.
The access privileges for your user are displayed.

Step 9 In the **Work** pane, in the **User Certificates** area, click the user certificates + sign, and in the **Create X509 Certificate** dialog box, perform the following actions:

a) In the **Name** field, enter a certificate name.
b) In the **Data** field, enter the user certificate details.
c) Click **Submit**.

The X509 certificate is created for the local user.

Creating a Local User and Adding a User Certificate Using the REST API

Procedure

Create a local user and add a user certificate.

Example:

```
method: POST
        url: http://apic/api/node/mo/uni/userext/user-userabc.json
        payload:
        {
            "aaaUser": {
                "attributes": {
                    "name": "userabc",
                    "firstName": "Adam",
                    "lastName": "BC",
                    "phone": "408-525-4766",
                    "email": "userabc@cisco.com",
                },
                "children": [{
                    "aaaUserCert": {
                        "attributes": {
                            "name": "userabc.crt",
                          "data": "-----BEGIN CERTIFICATE-----\nMIICjjCCAfegAwIBAgIJAMQnbE
 <snipped content> ==\n-----END CERTIFICATE-----",
                        },
                        "children": []
                    },
                    "aaaUserDomain": {
                        "attributes": {
                            "name": "all",
                        },
                        "children": [{
                            "aaaUserRole": {
                                "attributes": {
                                    "name": "aaa",
                                    "privType": "writePriv",
                                },
                                "children": []
                            }
                        }, {
                            "aaaUserRole": {
                                "attributes": {
                                    "name": "access-admin",
                                    "privType": "writePriv",
                                },
                                "children": []
                            }
                        }, {
                            "aaaUserRole": {
                                "attributes": {
                                    "name": "admin",
                                    "privType": "writePriv",
                                },
                                "children": []
                            }
                        }, {
                            "aaaUserRole": {
                                "attributes": {
                                    "name": "fabric-admin",
                                    "privType": "writePriv",
                                },
                                "children": []
                            }
                        }, {
                            "aaaUserRole": {
                                "attributes": {
                                    "name": "nw-svc-admin",
                                    "privType": "writePriv",
                                },
                                "children": []
```

```
                }
            }, {
                "aaaUserRole": {
                    "attributes": {
                        "name": "ops",
                        "privType": "writePriv",
                    },
                    "children": []
                }
            }, {
                "aaaUserRole": {
                    "attributes": {
                        "name": "read-all",
                        "privType": "writePriv",
                    },
                    "children": []
                }
            }, {
                "aaaUserRole": {
                    "attributes": {
                        "name": "tenant-admin",
                        "privType": "writePriv",
                    },
                    "children": []
                }
            }, {
                "aaaUserRole": {
                    "attributes": {
                        "name": "tenant-ext-admin",
                        "privType": "writePriv",
                    },
                    "children": []
                }
            }, {
                "aaaUserRole": {
                    "attributes": {
                        "name": "vmm-admin",
                        "privType": "writePriv",
                    },
                    "children": []
                }
            }]
        }
    }]
  }
}
```

Creating a Local User Using Python SDK

Procedure

Create a local user.

Example:

```
#!/usr/bin/env python
```

```
from cobra.model.pol import Uni as PolUni
from cobra.model.aaa import UserEp as AaaUserEp
from cobra.model.aaa import User as AaaUser
from cobra.model.aaa import UserCert as AaaUserCert
from cobra.model.aaa import UserDomain as AaaUserDomain
from cobra.model.aaa import UserRole as AaaUserRole
from cobra.mit.access import MoDirectory
from cobra.mit.session import LoginSession
from cobra.internal.codec.jsoncodec import toJSONStr

APIC = 'http://10.10.10.1'
username = 'admin'
password = 'p@$$w0rd'

session = LoginSession(APIC, username, password)
modir = MoDirectory(session)
modir.login()

def readFile(fileName=None, mode="r"):
   if fileName is None:
      return ""
   fileData = ""
   with open(fileName, mode) as aFile:
      fileData = aFile.read()
   return fileData

# Use a dictionary to define the domain and a list of tuples to define
# our aaaUserRoles (roleName, privType)
# This can further be abstracted by doing a query to get the valid
# roles, that is what the GUI does

userRoles = {'all': [
               ('aaa', 'writePriv'),
               ('access-admin', 'writePriv'),
               ('admin', 'writePriv'),
               ('fabric-admin', 'writePriv'),
               ('nw-svc-admin', 'writePriv'),
               ('ops', 'writePriv'),
               ('read-all', 'writePriv'),
               ('tenant-admin', 'writePriv'),
               ('tenant-ext-admin', 'writePriv'),
               ('vmm-admin', 'writePriv'),
           ],
}

uni = PolUni('') # '' is the Dn string for topRoot
aaaUserEp = AaaUserEp(uni)
aaaUser = AaaUser(aaaUserEp, 'userabc', firstName='Adam',
                  email='userabc@cisco.com')

aaaUser.lastName = 'BC'
aaaUser.phone = '555-111-2222'
aaaUserCert = AaaUserCert(aaaUser, 'userabc.crt')
aaaUserCert.data = readFile("/tmp/userabc.crt")
# Now add each aaaUserRole to the aaaUserDomains which are added to the
# aaaUserCert
for domain,roles in userRoles.items():
   aaaUserDomain = AaaUserDomain(aaaUser, domain)
   for roleName, privType in roles:
       aaaUserRole = AaaUserRole(aaaUserDomain, roleName,
                                 privType=privType)
print toJSONStr(aaaUser, prettyPrint=True)

cr = ConfigRequest()
```

```
cr.addMo(aaaUser)
modir.commit(cr)
# End of Script to create a user
```

Using a Private Key to Calculate a Signature

Before you begin

You must have the following information available:

- HTTP method - GET, POST, DELETE
- REST API URI being requested, including any query options
- For POST requests, the actual payload being sent to the APIC
- The private key used to generate the X.509 certificate for the user
- The distinguished name for the user X.509 certificate on the APIC

Procedure

Step 1 Concatenate the HTTP method, REST API URI, and payload together in this order and save them to a file.

This concatenated data must be saved to a file for OpenSSL to calculate the signature. In this example, we use a filename of payload.txt. Remember that the private key is in a file called userabc.key.

Example:

GET example:

```
GET http://10.10.10.1/api/class/fvTenant.json?rsp-subtree=children
```

POST example:

```
POST http://10.10.10.1/api/mo/tn-test.json{"fvTenant": {"attributes": {"status": "deleted",
 "name": "test"}}}
```

Step 2 Verify that the payload.txt file contains the correct information.

For example, using the GET example shown in the previous step:

```
GET http://10.10.10.1/api/class/fvTenant.json?rsp-subtree=children
```

Your payload.txt file should contain only the following information:

```
GET/api/class/fvTenant.json?rsp-subtree=children
```

Step 3 Verify that you didn't inadvertently create a new line when you created the payload file.

Example:

```
# cat -e payload.txt
```

Determine if there is a **$** symbol at the end of the output, similar to the following:

```
GET/api/class/fvTenant.json?rsp=subtree=children$
```

If so, then that means that a new line was created when you created the payload file. To prevent creating a new line when generating the payload file, use a command similar to the following:

```
echo -n "GET/api/class/fvTenant.json?rsp-subtree=children" >payload.txt
```

Step 4 Calculate a signature using the private key and the payload file using OpenSSL.

Example:

```
openssl dgst -sha256 -sign userabc.key payload.txt > payload_sig.bin
```

The resulting file has the signature printed on multiple lines.

Step 5 Convert the signature to base64 format:

Example:

```
openssl base64 -A -in payload_sig.bin -out payload_sig.base64
```

Step 6 Strip the signature of the new lines using Bash.

Example:

```
$ tr -d '\n' < payload_sig.base64
P+OTqK0CeAZjl7+Gute2R1Ww8OGgtzE0wsLlx8fIXXl4V79Zl7
Ou8IdJH9CB4W6CEvdICXqkv3KaQszCIC0+Bn07o3qF//BsIplZmYChD6gCX3f7q
IcjGX+R6HAqGeK7k97cNhXlWEoobFPe/oajtPjOu3tdOjhf/9ujG6Jv6Ro=
```

Note
This is the signature that will be sent to the APIC for this specific request. Other requests will require to have their own signatures calculated.

Step 7 Place the signature inside a string to enable the APIC to verify the signature against the payload.

This complete signature is sent to the APIC as a cookie in the header of the request.

Example:

```
APIC-Request-Signature=P+OTqK0CeAZjl7+Gute2R1Ww8OGgtzE0wsLlx8f
IXXl4V79Zl7Ou8IdJH9CB4W6CEvdICXqkv3KaQszCIC0+Bn07o3qF//BsIplZmYChD6gCX3f
7qIcjGX+R6HAqGeK7k97cNhXlWEoobFPe/oajtPjOu3tdOjhf/9ujG6Jv6Ro=;
APIC-Certificate-Algorithm=v1.0; APIC-Certificate-Fingerprint=fingerprint;
APIC-Certificate-DN=uni/userext/user-userabc/usercert-userabc.crt
```

Note
The DN used here must match the DN of the user certified object containing the x509 certificate in the next step.

Step 8 Use the CertSession class in the Python SDK to communicate with an APIC using signatures.

The following script is an example of how to use the CertSession class in the ACI Python SDK to make requests to an APIC using signatures.

Example:

```
#!/usr/bin/env python
# It is assumed the user has the X.509 certificate already added to
# their local user configuration on the APIC
from cobra.mit.session import CertSession
from cobra.mit.access import MoDirectory

def readFile(fileName=None, mode="r"):
    if fileName is None:
        return ""
    fileData = ""
```

```
    with open(fileName, mode) as aFile:
        fileData = aFile.read()
    return fileData

pkey = readFile("/tmp/userabc.key")
csession = CertSession("https://ApicIPOrHostname/",
                       "uni/userext/user-userabc/usercert-userabc", pkey)

modir = MoDirectory(csession)
resp = modir.lookupByDn('uni/fabric')
pring resp.dn
# End of script
```

Note
The DN used in the earlier step must match the DN of the user certified object containing the x509 certificate in this step.

Configuring a Local User for OTP-Based Two-Factor Authentication Using the GUI

The following procedure configures OTP-based two-factor authentication for a local user using the Cisco APIC GUI. The procedure assumes that you are a fabric administrator.

Before you begin

You must have already created a local user for which you want to enable OTP-based two-factor authentication.

Procedure

Step 1 On the menu bar, choose **Admin** > **AAA**.

Step 2 In the **Navigation** pane, choose **Users**.

Step 3 Double-click the user for which you want to enable OTP-based two-factor authentication.

Step 4 For **Enable OTP**, choose **Yes**.

Step 5 Click **Submit**.

Step 6 In the **Policy Usage Warning** dialog, click **Submit Changes**.

From this point, you can provide the QR code or OTP key to the user such as through an email, or the user can log into the Cisco APIC GUI to get the code and key directly.

Step 7 (Optional) If you want to provide the QR code or OTP key to the user, continue with the substeps. Otherwise, stop here.

a) Double-click the user again.

b) To the right of **OTP Key**, click the expand button.

A dialog displays that includes the QR code and OTP key.

c) Provide the QR code or OTP key to the user.

d) In the Cisco APIC, click **Close**.

e) Click **Close**.

What to do next

The user for which you enabled OTP must complete the configuration of OTP authentication. See Completing the Configuration of OTP-Based Two-Factor Authentication by a User Using the GUI, on page 39.

Completing the Configuration of OTP-Based Two-Factor Authentication by a User Using the GUI

The following procedure completes the configuration of OTP-based two-factor authentication using the Cisco APIC GUI. The procedure assumes that you are a user for which a fabric administrator enabled OTP-based two-factor authentication.

Before you begin

A fabric administrator must have enabled OTP-based two-factor authentication for your account.

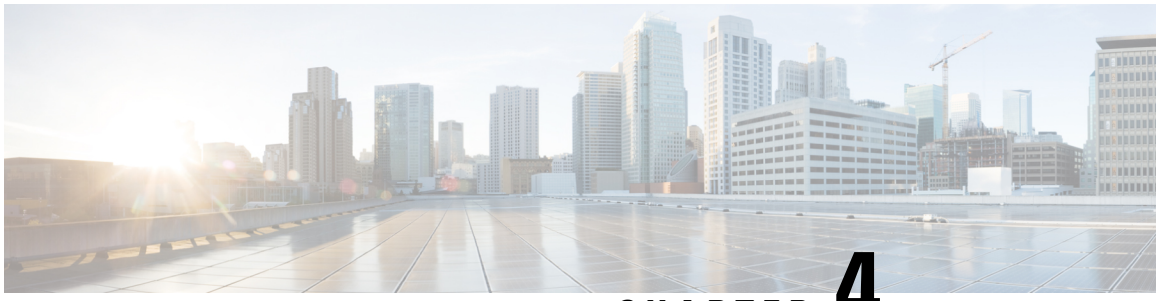
Procedure

Step 1 On your Android or Apple iOS smartphone, download the appropriate two-factor authentication app.

Step 2 Get the QR code or OTP key from the fabric administrator or by logging in to the Cisco APIC GUI.

If you log into the GUI, the QR code and OTP key display after you enter your credentials.

Step 3 Using your smartphone, scan the QR code and follow the two-factor authentication app's directions, or enter the OTP key in the Cisco APIC GUI.



CHAPTER 4

TACACs+, RADIUS, LDAP, RSA, and SAML

This chapter contains the following sections:

- Overview, on page 41
- RADIUS , on page 41
- TACACS+ Authentication, on page 42
- User IDs in the APIC Bash Shell, on page 42
- Login Domains, on page 43
- LDAP/Active Directory Authentication, on page 43
- RSA Secure ID Authentication, on page 44
- Configuring APIC for RSA Access Using the GUI, on page 44
- Configuring a Remote User, on page 45
- About SAML, on page 56

Overview

This article provides step by step instructions on how to enable RADIUS, TACACS+, and LDAP users access the APIC. It assumes the reader is thoroughly familiar with the Cisco Application Centric Infrastructure Fundamentals manual, especially the User Access, Authentication, and Accounting chapter.

Note Remote users for AAA Authentication with shell:domains=all/read-all/ will not be able to access Leaf switches and Spine switches in the fabric for security purposes. This pertains to all version up to 4.0(1h).

RADIUS

To configure users on RADIUS servers, the APIC administrator must configure the required attributes (`shell:domains`) using the `cisco-av-pair` attribute. The default user role is network-operator.

The SNMPv3 authentication protocol options are SHA and MD5. The privacy protocol options are AES-128 and DES. If these options are not specified in the `cisco-av-pair` attribute, MD5 and DES are the default authentication protocols.

For example, SNMPv3 authentication and privacy protocol attributes can be specified as follows:

```
snmpv3:auth=SHA priv=AES-128
```

Similarly, the list of domains would be as follows:

```
shell:domains="domainA domainB …"
```

TACACS+ Authentication

Terminal Access Controller Access Control System Plus (TACACS+) is another remote AAA protocol that is supported by Cisco devices. TACACS+ has the following advantages over RADIUS authentication:

- Provides independent AAA facilities. For example, the Cisco Application Policy Infrastructure Controller (APIC) can authorize access without authenticating.
- Uses TCP to send data between the AAA client and server, enabling reliable transfers with a connection-oriented protocol.
- Encrypts the entire protocol payload between the switch and the AAA server to ensure higher data confidentiality. RADIUS encrypts passwords only.
- Uses the av-pairs that are syntactically and configurationally different than RADIUS but the Cisco APIC supports `shell:domains`.

The following XML example configures the Cisco Application Centric Infrastructure (ACI) fabric to work with a TACACS+ provider at IP address 10.193.208.9:

```
<aaaTacacsPlusProvider name="10.193.208.9"
           key="test123"
           authProtocol="pap"/>
```

Note While the examples provided here use IPv4 addresses, IPv6 addresses could also be used.

The following guidelines and limitations apply when using TACACS+:

- The TACACS server and TACACs ports must be reachable by ping.
- The TACACS server with the highest priority is considered first to be the primary server.

User IDs in the APIC Bash Shell

User IDs on the APIC for the Linux shell are generated within the APIC for local users. Users whose authentication credential is managed on external servers, the user ID for the Linux shell can be specified in the cisco-av-pair. Omitting the (16001) in the above cisco-av-pair is legal, in which case the remote user gets a default Linux user ID of 23999. Linux User IDs are used during bash sessions, allowing standard Linux permissions enforcement. Also, all managed objects created by a user are marked as created-by that user's Linux user ID.

The following is an example of a user ID as seen in the APIC Bash shell:

```
admin@ifav17-ifc1:~> touch myfile
admin@ifav17-ifc1:~> ls -l myfile
-rw-rw-r-- 1 admin admin 0 Apr 13 21:43 myfile
admin@ifav17-ifc1:~> ls -ln myfile
```

```
-rw-rw-r-- 1 15374 15374 0 Apr 13 21:43 myfile
admin@ifav17-ifc1:~> id
uid=15374(admin) gid=15374(admin) groups=15374(admin)
```

Login Domains

A login domain defines the authentication domain for a user. Login domains can be set to the Local, LDAP, RADIUS, or TACACS+ authentication mechanisms. When accessing the system from REST, the CLI, or the GUI, the APIC enables the user to select the correct authentication domain.

For example, in the REST scenario, the username is prefixed with a string so that the full login username looks as follows:

`apic:<domain>\<username>`

If accessing the system from the GUI, the APIC offers a drop-down list of domains for the user to select. If no `apic: domain` is specified, the default authentication domain servers are used to look up the username.

Starting in ACI version 1.0(2x), the login domain fallback of the APIC defaults local. If the default authentication is set to a non-local method and the console authentication method is also set to a non-local method and both non-local methods do not automatically fall back to local authentication, the APIC can still be accessed via local authentication.

To access the APIC fallback local authentication, use the following strings:

- From the GUI, use *apic:fallback\\username*.
- From the REST API, use *apic#fallback\\username*.

Note Do not change the fallback login domain. Doing so could result in being locked out of the system.

LDAP/Active Directory Authentication

Similar to RADIUS and TACACS+, LDAP allows a network element to retrieve AAA credentials that can be used to authenticate and then authorize the user to perform certain actions. An added certificate authority configuration can be performed by an administrator to enable LDAPS (LDAP over SSL) trust and prevent man-in-the-middle attacks.

The XML example below configures the ACI fabric to work with an LDAP provider at IP address 10.30.12.128.

Note While the examples provided here use IPv4 addresses, IPv6 addresses could also be used.

```
<aaaLdapProvider name="10.30.12.128"
          rootdn="CN=Manager,DC=ifc,DC=com"
          basedn="DC=ifc,DC=com"
          SSLValidationLevel="strict"
          attribute="CiscoAVPair"
          enableSSL="yes"
          filter="cn=$userid"
          port="636" />
```

Note For LDAP configurations, best practice is to use `CiscoAVPair` as the attribute string. If customer faces the issue using Object ID 1.3.6.1.4.1.9.22.1, an additional Object ID 1.3.6.1.4.1.9.2742.1-5 can also be used in the LDAP server.

Instead of configuring the Cisco AVPair, you have the option to create LDAP group maps in the APIC.

RSA Secure ID Authentication

RSA Authentication provides a token which can be used in combination with a fixed key in many different ways to create the password. It supports both hardware and software tokens.

Configuring APIC for RSA Access Using the GUI

Before you begin

- The ACI fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The RSA server host name or IP address, port, authorization protocol, and key are available.
- The APIC management endpoint group is available.

Procedure

Step 1 In the APIC, create the RSA provider.

a) On the menu bar, choose **Admin** > **AAA**.
b) In the **Navigation** pane, choose **RSA Management** > **RSA Providers**.
c) In the **Work** pane, choose **Actions** > **Create RSA Provider**.
d) Specify the RSA host name (or IP address), port, protocol, and management endpoint group.

Step 2 Create the login domain for RSA.

a) In the **Navigation** pane, choose **AAA Authentication** > **Login Domains**.
b) In the **Work** pane, choose **Actions** > **Create Login Domain**.
c) Specify the login domain name, description, realm, and provider group as appropriate.

What to do next

This completes the APIC RSA configuration steps. Next, configure the RSA server.

Configuring a Remote User

Instead of configuring local users, you can point the APIC at the centralized enterprise credential datacenter. The APIC supports Lightweight Directory Access Protocol (LDAP), active directory, RADIUS, and TACACS+.

Note When an APIC is in minority (disconnected from the cluster), remote logins can fail because the ACI is a distributed system and the user information is distributed across APICS. Local logins, however, continue to work because they are local to the APIC.

Starting with the 3.1(1) release, **Server Monitoring** can be configured through RADIUS, TACACS+, LDAP, and RSA to determine whether the respective AAA servers are alive or not. Server monitoring feature uses the respective protocol login to check for server aliveness. For example, a LDAP server will use ldap login and a Radius server will use radius login with server monitoring to determine server aliveness.

To configure a remote user authenticated through an external authentication provider, you must meet the following prerequisites:

- The DNS configuration should have already been resolved with the hostname of the RADIUS server.
- You must configure the management subnet.

AV Pair on the External Authentication Server

The Cisco APIC requires that an administrator configure a Cisco AV Pair on an external authentication server. The Cisco AV pair specifies the APIC required RBAC roles and privileges for the user. The Cisco AV Pair format is the same for RADIUS, LDAP, or TACACS+.

To configure a Cisco AV Pair on an external authentication server, an administrator adds a Cisco AV pair to the existing user record. The Cisco AV pair format is as follows:

```
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,
domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,
domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2(16003)
```

Starting with Cisco APIC release 2.1, if no UNIX ID is provided in AV Pair, the APIC allocates the unique UNIX user ID internally.

Note The APIC Cisco AV-pair format is compatible and can co-exist with other Cisco AV-pair formats. APIC will pick up the first matching AV-pair from all the AV-pairs.

The APIC supports the following regexes:

```
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})$
```

Examples:

- Example 1: A Cisco AV Pair that contains a single Login domain with only writeRoles:

```
shell:domains=domainA/writeRole1|writeRole2/
```

- Example 2: A Cisco AV Pair that contains a single Login domain with only readRoles:

```
shell:domains=domainA//readRole1|readRole2
```

Note The "/" character is a separator between writeRoles and readRoles per Login domain and is required even if only one type of role is to be used.

The Cisco AVpair string is case sensitive. Although a fault may not be seen, using mismatching cases for the domain name or roles could lead to unexpected privileges being given.

An example configuration for an open RADIUS server (/etc/raddb/users) is as follows:

```
aaa-network-admin Cleartext-Password := "<password>"
Cisco-avpair = "shell:domains = all/aaa/read-all(16001)"
```

Best Practice for Assigning AV Pairs

As best practice,

Cisco recommends that you assign unique UNIX user ids in the range of 16000 to 23999 for the AV Pairs that are assigned to users when in bash shell (using SSH, Telnet or Serial/KVM consoles). If a situation arises when the Cisco AV Pair does not provide a UNIX user id, the user is assigned a user id of 23999 or similar number from the range that also enables the user's home directories, files, and processes accessible to remote users with a UNIX ID of 23999.

To ensure that your remote authentication server does NOT explicitly assign a UNIX ID in its cisco-av-pair response, open an SSH session to the APIC and login as an administrator (using a remote user account). Once logged in, run the following commands (replace “userid” with the username you logged in with):

```
admin@apic1:remoteuser-userid> cd /mit/uni/userext/remoteuser-userid
admin@apic1:remoteuser-userid> cat summary
```

The Cisco AVpair string is case sensitive. Although a fault may not be seen, using mismatching cases for the domain name or roles could lead to unexpected privileges being given.

Configuring an AV Pair on the External Authentication Server

The numerical value within the parentheses in the attribute/value (AV) pair string is used as the UNIX user ID of the user who is logged in using Secure Shell (SSH) or Telnet.

Procedure

Configure an AV pair on the external authentication server.

The Cisco AV pair definition is as follows (Cisco supports AV pairs with and without UNIX user IDs specified):

Example:

```
* shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2

         * shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2(8101)

These are the boost regexes supported by APIC:
uid_regex("shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$");
regex("shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})$");
```

The following is an example:

```
shell:domains = coke/tenant-admin/read-all,pepsi//read-all(16001)
```

Configuring APIC for TACACS+ Access

Before you begin

- The Cisco Application Centric Infrastructure (ACI) fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The TACACS+ server host name or IP address, port, and key are available.
- The APIC management endpoint group is available.

Procedure

Step 1 In the APIC, create the **TACACS+ Provider**.

a) On the menu bar, choose **Admin** > **AAA**.
b) In the **Navigation** pane, choose **TACACS+ Managment** > **TACACS+ Providers**.
c) In the **Work** pane, choose **Actions** > **Create TACACS+ Provider**.
d) Specify the TACACS+ host name (or IP address), port, authorization protocol, key, and management endpoint group.

Note
If the APIC is configured for in-band management connectivity, out-of-band management does not work for authentication. With the APIC release2.1(1x), you can set a global toggle between In-band and out-of-band as the default management connectivity between the APIC server and other external management devices.

For toggling in-band or out-of-band management in the APIC GUI:

- Prior to Release 2.2(1x): In the **Navigation** pane, choose **Fabric** > **Fabric Policies** > **Global Policies** > **Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.
- For Release 2.2(x) and 2.3(x): In the **Navigation** pane, choose **Fabric** > **Fabric Policies** > **Global Policies** > **APIC Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.

- For Release 3.0(1x) or later: In the **Navigation** pane, choose **System** > **System Settings** > **APIC Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.

Step 2 Create the **TACACS+ Provider Group**.

a) In the **Navigation** pane, choose **TACACS+ Managment** > **TACACS+ Provider Groups**.
b) In the **Work** pane, choose **Actions** > **Create TACACS+ Provider Group**.
c) Specify the TACACS+ provider group name, description, and providers as appropriate.

Step 3 Create the **Login Domain** for TACACS+.

a) In the **Navigation** pane, choose **AAA Authentication** > **Login Domains**.
b) In the **Work** pane, choose **Actions** > **Create Login Domain**.
c) Specify the login domain name, description, realm, and provider group as appropriate.

What to do next

This completes the APIC TACACS+ configuration steps. Next, if a RAIDUS server will also be used, configure the APIC for RADIUS. If only a TACACS+ server will be used, go to the ACS server configuration topic below.

Configuring APIC for RADIUS Access

Before you begin

- The ACI fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The RADIUS server host name or IP address, port, authorization protocol, and key are available.
- The APIC management endpoint group is available.

Procedure

Step 1 In the APIC, create the RADIUS provider.

a) On the menu bar, choose **Admin** > **AAA**.
b) In the **Navigation** pane, click on **Authentication** and then click on the **RADIUS** tab.
c) In the **Work** pane, choose **Actions** > **Create RADIUS Provider**.
d) Specify the RADIUS host name (or IP address), port, protocol, and management endpoint group.

Note
If the APIC is configured for in-band management connectivity, out-of-band management does not work for authentication. With the APIC release2.1(1x), you can set a global toggle between In-band and out-of-band as the default management connectivity between the APIC server and other external management devices.

For toggling in-band or out-of-band management in the APIC GUI:

- Prior to Release 2.2(1x): In the **Navigation** pane, choose **Fabric** > **Fabric Policies** > **Global Policies** > **Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.
- For Release 2.2(x) and 2.3(x): In the **Navigation** pane, choose **Fabric** > **Fabric Policies** > **Global Policies** > **APIC Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.
- For Release 3.0(1x) or later: In the **Navigation** pane, choose **System** > **System Settings** > **APIC Connectivity Preferences**. In the **Work Pane** select either **inband** or **ooband**.

Step 2 Create the login domain for RADIUS.

a) In the **Navigation** pane, choose **AAA Authentication** > **Login Domains**.
b) In the **Work** pane, choose **Actions** > **Create Login Domain**.
c) Specify the login domain name, description, realm, and provider group as appropriate.

What to do next

This completes the APIC RADIUS configuration steps. Next, configure the RADIUS server.

Configuring a Cisco Secure Access Control Server for RADIUS and TACACS+ Access to the APIC

Before you begin

- The Cisco Secure Access Control Server (ACS) version 5.5 is installed and online.

Note ACS v5.5 was used to document these steps. Other versions of ACS might support this task but the GUI procedures might vary accordingly.

- The Cisco Application Policy Infrastructure Controller (Cisco APIC) RADIUS or TACACS+ keys are available (or keys for both if both will be configured).
- The Cisco APICs are installed and online; the Cisco APIC cluster is formed and healthy.
- The RADIUS or TACACS+ port, authorization protocol, and key are available.

Procedure

Step 1 Log in to the ACS server to configure the Cisco APIC as a client.

a) Navigate to **Network Resources** > **Network Devices Groups** > **Network Devices and AAA Clients**.
b) Specify the client name, the Cisco APIC in-band IP address, select the TACACS+ or RADIUS (or both) authentication options.

Note
If the only RADIUS or TACACS+ authentication is needed, select only the needed option.

c) Specify the authentication details such as Shared Secret (key), and port as appropriate for the authentication option(s).

Note
The **Shared Secret**(s) must match the Cisco APIC **Provider** key(s).

Step 2 Create the Identity Group.

a) Navigate to **Users and Identity Stores** > **Internal Groups** option.
b) Specify the **Name**, and **Parent Group** as appropriate.

Step 3 Map users to the Identity Group.

a) In the **Navigation** pane, click the **Users and Identity Stores** > **Internal Identity Stores** > **Users** option.
b) Specify the user **Name**, and **Identity Group** as appropriate.

Step 4 Create the Policy Element.

a) Navigate to the **Policy Elements** option.
b) For RADIUS, specify the Authorization and Permissions > Network Access > Authorization Profiles **Name**. For TACACS+, specify the Authorization and Permissions > Device Administration > Shell Profile **Name** as appropriate.
c) For RADIUS, specify the **Attribute** as `cisco-av-pair`, **Type** as string, and the **Value** as **`shell:domains = <domain>/<role>/,<domain>// role`** as appropriate. For TACACS+, specify the **Attribute** as `cisco-av-pair`, **Requirement** as Mandatory, and the **Value** as **`shell:domains = <domain>/<role>/,<domain>// role`** as appropriate.

For example, if the *cisco-av-pair* has a value of `shell:domains = solar/admin/,common// read-all(16001)`, then `solar` is the security domain, `admin` is the role for this user that gives write privileges to this user in the security domain called `solar`, `common` is the Cisco Application Centric Infrastructure (Cisco ACI) tenant common, and `read-all(16001)` is the role with read privileges that gives this user read privileges to all of the Cisco ACI tenant common.

Step 5 Create a service selection rule.

a) For RADIUS, create a service selection rule to associate the Identity Group with the Policy Element by navigating to **Access Policies** > **Default Device Network Access Identity** > **Authorization** and specifying the rule **Name**, **Status**, and **Conditions** as appropriate, and **Add** the `Internal Users:UserIdentityGroup in ALL Groups:<identity group name>`.
b) For TACACS+, create a service selection rule to associate the Identity Group with the Shell Profile by navigating to **Access Policies** > **Default Device Admin Identity** > **Authorization**. Specify the rule **Name**, **Conditions**, and **Select** the **Shell Profile** as appropriate.

What to do next

Use the newly created RADIUS and TACACS+ users to log in to the Cisco APIC. Verify that the users have access to the correct Cisco APIC security domain according to the assigned RBAC roles and privileges. The users should not have access to items that have not been explicitly permitted. Read and write access rights should match those configured for that user.

Configuring LDAP

There are two options for LDAP configurations: you can configure a Cisco AVPair or configure LDAP group maps in the APIC. This section contains instructions for both configuration options.

Configuring Windows Server 2008 LDAP for APIC Access with Cisco AVPair

Before you begin

- First, configure the LDAP server, then configure the Cisco Application Policy Infrastructure Controller (Cisco APIC) for LDAP access.
- The Microsoft Windows Server 2008 is installed and online.
- The Microsoft Windows Server 2008 Server Manager ADSI Edit tool is installed. To install ADSI Edit, follow the instructions in the Windows Server 2008 Server Manager help.
- `CiscoAVPair` attribute specifications: Common Name = **`CiscoAVPair`**, LDAP Display Name = **`CiscoAVPair`**, Unique X500 Object ID = 1.3.6.1.4.1.9.22.1, Description = **`CiscoAVPair`**, Syntax = **`Case Sensitive String`**.

Note For LDAP configurations, best practice is to use **`CiscoAVPair`** as the attribute string. If customer faces the issue using Object ID 1.3.6.1.4.1.9.22.1, an additional Object ID 1.3.6.1.4.1.9.2742.1-5 can also be used in the LDAP server.

- A Microsoft Windows Server 2008 user account is available that will enable the following:
  - Running ADSI Edit to add the `CiscoAVPair` attribute to the Active Directory (AD) Schema.
  - Configuring an Active Directory LDAP user to have `CiscoAVPair` attribute permissions.
- Port 636 is required for configuring LDAP integration with SSL/TLS.

Procedure

Step 1 Log in to an Active Directory (AD) server as a domain administrator.

Step 2 Add the `CiscoAVPair` attribute to the AD schema.

a) Navigate to **Start** > **Run**, type **`mmc`** and press **Enter**.
The Microsoft Management Console (MMC) opens.

b) Navigate to **File** > **Add/Remove Sanp-in** > **Add**.

c) In the **Add Standalonee Snap-in** dialog box, select the **Active Directory Schema** and click **Add**.
The MMC **Console** opens.

d) Right-click the **Attributes** folder, select the **Create Attribute** option.
The **Create New Attribute** dialog box opens.

e) Enter **`CiscoAVPair`** for the **Common Name** , **`CiscoAVPair`** for the **LDAP Display Name**, **`1.3.6.1.4.1.9.22.1`** for the **Unique X500 Object ID**, and select **`Case Sensitive String`** for the **Syntax**.

f) Click **OK** to save the attribute.

Step 3 Update the **User Properties** class to include the **CiscoAVPair** attribute.

a) In the MMC **Console**, expand the **Classes** folder, right-click the **user** class, and choose **Properties**. The **user Properties** dialog box opens.
b) Click the **Attributes** tab, and click **Add** to open the **Select Schema Object** window.
c) In the **Select a schema object:** list, choose **`CiscoAVPair`**, and click **Apply**.
d) In the MMC **Console**, right-click the **Active Directory Schema**, and select **Reload the Schema**.

Step 4 Configure the **`CiscoAVPair`** attribute permissions.

Now that the LDAP includes the **`CiscoAVPair`** attributes, LDAP users need to be granted Cisco APIC permission by assigning them Cisco APIC RBAC roles.

a) In the ADSI Edit dialog box, locate a user who needs access to the Cisco APIC.
b) Right-click on the user name, and choose **Properties**. The **<user> Properties** dialog box opens.
c) Click the **Attribute Editor** tab, select the *CiscoAVPair* attribute, and enter the *Value* as **`shell:domains = <domain>/<role>/,<domain>// role`**.

For example, if the *CiscoAVPair* has a value of `shell:domains = solar/admin/,common// read-all(16001)`, then `solar` is the security domain, `admin` is the role for this user that gives write privileges to this user in the security domain called `solar`, `common` is the Cisco Application Centric Infrastructure (Cisco ACI) tenant common, and `read-all(16001)` is the role with read privileges that gives this user read privileges to all of the Cisco ACI tenant common.

d) Click **OK** to save the changes and close the **<user> Properties** dialog box.

The LDAP server is configured to access the Cisco APIC.

What to do next

Configure the Cisco APIC for LDAP access.

Configuring APIC for LDAP Access

Before you begin

- The Cisco Application Centric Infrastructure (ACI) fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The LDAP server host name or IP address, port, bind DN, Base DN, and password are available.
- The APIC management endpoint group is available.

Procedure

Step 1 In the APIC, configure the LDAP Provider.

a) On the menu bar, choose **Admin** > **AAA**.
b) In the **Navigation** pane, choose **Authentication** and in the **Work** pane click on the **LDAP** tab.

c) In the **Work** pane, choose **Actions** > **Create LDAP Provider**.
d) Specify the LDAP host name (or IP address), port, bind DN, base DN, password, attribute, and management endpoint group.

Note

- The bind DN is the string that the APIC uses to log in to the LDAP server. The APIC uses this account to validate the remote user attempting to log in. The base DN is the container name and path in the LDAP server where the APIC searches for the remote user account. This is where the password is validated. Filter is used to locate the attribute that the APIC requests to use for the *cisco-av-pair*. This contains the user authorization and assigned RBAC roles for use on the APIC. The APIC requests the attribute from the LDAP server.
- **Attribute** field—Enter one of the following:
  - For LDAP server configurations with a Cisco AVPair, enter **`CiscoAVPair`**.
  - For LDAP server configurations with an LDAP group map, enter **`memberOf`**.
- If the APIC is configured for in-band management connectivity, choosing an out-of-band management endpoint group for LDAP access does not take effect. Alternatively, an out-of-band over an in-band management endpoint group can connect a LDAP server, but requires configuring a static route for the LDAP server. The sample configuration procedures in this document use an APIC in-band management endpoint group.

Step 2 On the APIC, configure the login domain for LDAP.
a) In the **Navigation** pane, choose **Authentication** > **Login Domains**.
b) In the **Work** pane, choose **Actions** > **Create Login Domain**.
c) Specify the login domain name, description, realm, and provider group as appropriate.

What to do next

This completes the APIC LDAP configuration steps. Next, test the APIC LDAP login access.

Configuring LDAP Group Map Rules on the Cisco APIC

Configuring an LDAP group map on the Cisco APIC requires first creating LDAP group map rules. This section explains how to create LDAP group map rules.

Before you begin

An LDAP server is running with a configured group mapping.

Procedure

Step 1 On the menu bar of the Cisco APIC GUI, choose **Admin** > **AAA**.
Step 2 In the **Navigation** pane, expand **LDAP Managment**, right-click **LDAP Group Map Rules**, and click **Create LDAP Group Map Rule**. The **Create LDAP Group Map Rule: Security** dialog appears.

Step 3 Specify the map rule name, description (optional), group DN, and security domain in the appropriate fields then click **Next**. The **Create LDAP Group Map Rule: Roles** dialog appears with security domain options.

Step 4 Click the **+** to access the Role Name and Role Privilege Type fields.

Step 5 Click the **Role Name** drop-down arrow to choose a role name.

Step 6 Click the **Role Privilege Type** drop-down arrow to choose a role privilege type (**Read** or **Write**) .

Repeat Step 4 to 6 to add additional roles to the LDAP group map rule.

Step 7 When finished, click **Finished**.

What to do next

After specifying the LDAP group map rules, create an LDAP group map.

Configuring an LDAP Group Map on the Cisco APIC

Configuring an LDAP group map on the Cisco APIC requires first creating LDAP group map rules. This section explains how to create an LDAP group map.

Before you begin

- A running LDAP server is configured with group mapping.
- LDAP group map rules have been configured.

Procedure

Step 1 On the menu bar of the Cisco APIC GUI, choose **Admin** > **AAA**.

Step 2 In the **Navigation** pane, expand **LDAP Managment**, right-click **LDAP Group Maps**, and click **Create LDAP Group Map**. The **Create LDAP Group Map** dialog appears.

Step 3 Specify the map name and description (optional).

Step 4 From the **Rules** field, click the **+** then click the **Name** drop-down arrow to choose a specified LDAP group map rule then click **Update**.

Repeat Step 4 to add additional rules to the LDAP group map.

Step 5 When finished, click **Submit**.

Configuring a Remote User Using the NX-OS Style CLI

Instead of configuring local users, you can point the APIC at the centralized enterprise credential datacenter. The APIC supports Lightweight Directory Access Protocol (LDAP), active directory, RADIUS, and TACACS+.

To configure a remote user authenticated through an external authentication provider, you must meet the following prerequisites:

- The DNS configuration should have already been resolved with the hostname of the RADIUS server.

- You must configure the management subnet.

Changing the Default Behavior for Remote Users with Missing or Bad Cisco AV Pairs

Procedure

Step 1 On the menu bar, click **ADMIN** > **AAA**.

Step 2 In the **Navigation** pane, click **Users**.

Step 3 In the **Work** pane, in the **Remote Users** area, from the **Remote user login policy** drop-down list, choose **Assign Default Role**.

The default value is **No Login**. The **Assign Default Role** option assigns the minimal read-only privileges to users that have missing or bad Cisco AV Pairs. Bad AV Pairs are those AV Pairs that fail the parsing rules.

Changing Default Behavior for Remote Users with Missing or Bad Cisco AV Pairs Using the NX-OS Style CLI

The Cisco APIC requires that an administrator configure a Cisco AV Pair on an external authentication server. To do so, an administrator adds a Cisco AV pair to the existing user record. The Cisco AV pair specifies the APIC required RBAC roles and privileges for the user. The Cisco AV Pair format is the same for RADIUS, LDAP, or TACACS+. One AV pair format contains a Cisco UNIX user ID and one does not. Both are correct if all remote users have the same role and mutual file access is acceptable. If the UNIX user ID is not specified, ID 23999 is applied by the APIC system, and more than one role/read privilege is specified to any AV Pair user. This can cause users to have higher or lower permissions than configured through the group settings. This topic explains how to change the bahavior if that is not acceptable.

To change the default behavior for remote users with missing or bad Cisco AV pairs using the NX-OS CLI:

Procedure

Step 1 In the NX-OS CLI, start in Configuration mode.

Example:

```
apic1#
apic1# configure
```

Step 2 Configure the aaa user default role.

Example:

```
apic1(config)# aaa user default-role
```

```
 assign-default-role  assign-default-role
 no-login             no-login
```

Step 3 Configure the aaa authentication login methods.

Example:

```
apic1(config)# aaa authentication
 login  Configure methods for login

apic1(config)# aaa authentication login
 console  Configure console methods
 default  Configure default methods
 domain   Configure domain methods

apic1(config)# aaa authentication login console
 <CR>

apic1(config)# aaa authentication login domain
 WORD      Login domain name
 fallback
```

About SAML

SAML is an XML-based open standard data format that enables administrators to access a defined set of Cisco collaboration applications seamlessly after signing into one of those applications. SAML describes the exchange of security related information between trusted business partners. It is an authentication protocol used by service providers to authenticate a user. SAML enables exchange of security authentication information between an Identity Provider (IdP) and a service provider.

SAML SSO uses the SAML 2.0 protocol to offer cross-domain and cross-product single sign-on for Cisco collaboration solutions. SAML 2.0 enables SSO across Cisco applications and enables federation between Cisco applications and an IdP. SAML 2.0 allows Cisco administrative users to access secure web domains to exchange user authentication and authorization data, between an IdP and a Service Provider while maintaining high security levels. The feature provides secure mechanisms to use common credentials and relevant information across various applications.

The authorization for SAML SSO Admin access is based on Role-Based Access Control (RBAC) configured locally on Cisco collaboration applications.

SAML SSO establishes a Circle of Trust (CoT) by exchanging metadata and certificates as part of the provisioning process between the IdP and the Service Provider. The Service Provider trusts the IdP's user information to provide access to the various services or applications.

Note Service providers are no longer involved in authentication. SAML 2.0 delegates authentication away from the service providers and to the IdPs.

The client authenticates against the IdP, and the IdP grants an Assertion to the client. The client presents the Assertion to the Service Provider. Since there is a CoT established, the Service Provider trusts the Assertion and grants access to the client.

Enabling SAML SSO results in several advantages:

- It reduces password fatigue by removing the need for entering different user name and password combinations.
- It transfers the authentication from your system that hosts the applications to a third party system. UsingSAML SSO, you can create a circle of trust between an IdP and a service provider. The service provider trusts and relies on the IdP to authenticate the users.
- It protects and secures authentication information. It provides encryption functions to protect authentication information passed between the IdP, service provider, and user. SAML SSO can also hide authentication messages passed between the IdP and the service provider from any external user.
- It improves productivity because you spend less time re-entering credentials for the same identity.
- It reduces costs as fewer help desk calls are made for password reset, thereby leading to more savings.

Basic Elements of SAML

- Client (the user's client): This is a browser-based client or a client that can leverage a browser instance for authentication. For example, a system administrator's browser.
- Service provider: This is the application or service that the client is trying to access.
- An Identity Provider (IdP) server: This is the entity that authenticates user credentials and issues SAML Assertions.
- Lightweight Directory Access Protocol (LDAP) users: These users are integrated with an LDAP directory, for example Microsoft Active Directory or OpenLDAP. Non-LDAP users reside locally on the Unified Communications server.
- SAML Assertion: It consists of pieces of security information that are transferred from IdPs to the service provider for user authentication. An assertion is an XML document that contains trusted statements about a subject including, for example, a username and privileges. SAML assertions are usually digitally signed to ensure their authenticity.
- SAML Request: This is an authentication request that is generated by a Unified Communications application. To authenticate the LDAP user, Unified Communications application delegates an authentication request to the IdP.
- Circle of Trust (CoT): It consists of the various service providers that share and authenticate against one IdP in common.
- Metadata: This is an XML file generated by an ACI application as well as an IdP. The exchange of SAML metadata builds a trust relationship between the IdP and the service provider.
- Assertion Consumer Service (ACS) URL: This URL instructs the IdPs where to post assertions. The ACS URL tells the IdP to post the final SAML response to a particular URL.

Note All in-scope services requiring authentication use SAML 2.0 as the SSO mechanism.

Supported IdPs and SAML Components

Supported IdPs

Identity Provider (IdP) is an authentication module that creates, maintains, and manages identity information for users, systems, or services and also provides authentication to other applications and service providers within a distributed network.

With SAML SSO, IdPs provide authentication options based on the user role or log in options for each of the Cisco collaboration applications. The IdPs store and validate the user credentials and generate a SAML response that allows the user to access the service provider protected resources.

Note You must be familiar with your IdP service, and ensure that it is currently installed and operational.

The APIC SAML SSO feature has been tested with following IdPs:

- https://technet.microsoft.com/en-us/library/cc772128(WS.10).aspx
- PingFederate: https://docs.pingidentity.com/pingfederate/latest/pf_pf_landing_page.html

SAML Components

A SAML SSO solution is based on a particular combination of assertions, protocols, bindings, and profiles. The various assertions are exchanged among applications and sites using the protocols and bindings, and those assertions authenticate the users among sites. The SAML components are as follows:

- SAML Assertion: It defines the structure and content of the information that is transferred from IdPs to service providers. It consists of packets of security information and contains statements that service providers use for various levels of access-control decisions.SAML SSO provides the following types of statements:
  - Authentication statements- These statements assert to the service provider about the method of authentication that occurs between the IdP and the browser at a particular time.
  - Attribute statements- These statements assert about certain attributes (name-value pairs) that are associated with the user. The attribute assertions contain specific information about the user. The service providers use attributes to make access-control decisions.
- SAML protocol: A SAML protocol defines how the SAML requests for and gets assertions. This protocol is responsible for the SAML request and response elements that consist of certain SAML elements or assertions. The SAML 2.0 contains the following protocols:
  - Assertion Query and Request Protocol
  - Authentication Request Protocol
- SAML binding: A SAML binding specifies the mapping of SAML assertion and/or protocol message exchanges with standard messaging formats or communication protocols like SOAP exchanges. ACI supports the following SAML 2.0 bindings:
  - HTTP Redirect (GET) Binding
  - HTTP POST Binding

- SAML profile: A SAML profile provides a detailed description of the combination of SAML assertions, protocols, and bindings to support well-defined use cases.

NTP Setup

In SAML SSO, Network Time Protocol (NTP) enables clock synchronization between the APIC and IdP. SAML is a time sensitive protocol and the IdP determines the time-based validity of a SAML assertion. If the IdP and the APIC clocks are not synchronized, the assertion becomes invalid and stops the SAML SSO feature. The maximum allowed time difference between the IdP and the APIC is 3 seconds.

Note For SAML SSO to work, you must install the correct NTP setup and make sure that the time difference between the IdP and the APIC does not exceed 3 seconds. If IdP and APIC clocks are not synchronized, the user will be redirected back to the APIC's login page even after successful authentication on IdP.

DNS Setup

Domain Name System (DNS) enables the mapping of host names and network services to IP addresses within a network or networks. DNS server(s) deployed within a network provide a database that maps network services to hostnames and, in turn, hostnames to IP addresses. Devices on the network can query the DNS server and receive IP addresses for other devices in the network, thereby facilitating communication between network devices.

In summary, APIC and Idp should be able to resolve each other's fully qualified domain names to IP addresses and should be resolvable by the client.

Certificate Authority

Cisco recommends using server certificates that are signed by one of the following types of Certificate Authority (CA):

- **Public CA**—A third-party company verifies the server identity and issues a trusted certificate.
- **Private CA**—You create and manage a local CA and issue trusted certificates.

The signing process varies for each product and can vary between server versions. It is beyond the scope of this document to provide detailed steps for every version of each server. Refer the appropriate server documentation for detailed instructions on how to get certificates signed by a CA.

If you get server certificates signed by a public CA, the public CA should already have a root certificate present in the trust store on the client computer. In this case, you do not need to import root certificates on the client computers. You should import root certificates if the certificates are signed by a CA that does not already exist in the trust store, such as a private CA. In SAML SSO, the IdP and service providers must have CA signed certificates with the correct domains in the CN or SAN. If the correct CA certificates are not validated, the browser issues a pop up warning.

If the APIC's trust store does not include the root certificate of the IdP, a new certificate authority should be created. This Certificate Authority should be used later while configuring the SAML Provider on APIC.

Configuring APIC for SAML Access

Note SAML based Authentication is only for APIC GUI and not for CLI/REST. Also, not applicable for LEAF Switches and SPINEs. SAML configuration cannot be done via APIC CLI.

Before you begin

- The Cisco Application Centric Infrastructure (ACI) fabric is installed, Application Policy Infrastructure Controllers (APICs) are online, and the APIC cluster is formed and healthy.
- The SAML server host name or IP address, and the IdP's metadata URL are available..
- The APIC management endpoint group is available.
- Set up the following:
  - Time Synchronization and NTP: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/3-x/basic_config/b_APIC_Basic_Config_Guide_3_x/b_APIC_Basic_Config_Guide_3_x_chapter_011.html#concept_9CE11B84AD78486AA7D83A7DE1CE2A77.
  - Configuring a DNS Service Policy to Connect with DNS Providers Using the Advanced GUI: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/3-x/basic_config/b_APIC_Basic_Config_Guide_3_x/b_APIC_Basic_Config_Guide_3_x_chapter_011.html#task_750E077676704BFBB5B0FE74628D821E.
  - Configuring a Custom Certificate for Cisco ACI HTTPS Access Using the GUI: https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/3-x/basic_config/b_APIC_Basic_Config_Guide_3_x/b_APIC_Basic_Config_Guide_3_x_chapter_011.html#task_F037F1B75FF74ED1BCA4F3C75A16C0FA.

Procedure

Step 1 In the APIC, create the **SAML Provider**.

a) On the menu bar, choose **Admin** > **AAA**.
b) In the **Navigation** pane, choose **SAML Management** > **SAML Providers**.
c) In the **Work** pane, choose **Actions** > **Create SAML Provider**.
d) Specify the SAML host name (or IP address), and IdP metadata URL.
   - In case of AD FS, IdP Metadata URL is of the format *https://<FQDN of ADFS>/FederationMetadata/2007-06/FederationMetadata.xml*.
e) Specify the **Entity ID** for the SAML-based service.
f) Configure the Https Proxy if it is needed to access the IdP metadata URL.
g) Select the Certificate Authority if IdP is signed by a Private CA.
h) Select the signature algorithm authentication type for the user requests from the drop-down.

Step 2 Create the **Login Domain** for SAML.

a) In the **Navigation** pane, choose **AAA Authentication** > **Login Domains**.

b) In the **Work** pane, choose **Actions** > **Create Login Domain**.
c) Specify the login domain name, description, realm, and provider group as appropriate.

Setting Up a Relying Party Trust in AD FS

Add relying party trust in AD FS Management Console:

Procedure

Step 1 Add relying party trust:

a) Login to AD FS Management Console on your AD FS server, Navigate to **ADFS** > **Trust Relationships** > **Relying Party Trusts** and right-click on **Add Relying Party Trust** and click **Start**.
b) Choose **Enter data about the relying party manually** or **Import data about relying party from a file (skip the steps d, e, f and g)** by importing the metadata file generated using the **Download SAML Metadata** option available on the corresponding login domain setup in APIC.
c) Enter your preferred **Display Name** for the relying party trust and click **Next**.
d) Choose AD FS Profile and click **Next**.
e) Click **Next** again.
f) Select **Enable support for the SAML 2.0 Web SSO Protocol** and enter **Relying party SAML2.0 SSO service UR** as *https://<APIC_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>* and click **Next**.
g) Enter the **Relying party trust identifier** – *https://<APIC_hostname>/api/aaaLoginSSO.json*
h) Choose **I do not want to configure multi-factor authentication settings for this relying party trust at this time** and click **Next**.
i) Choose **Permit all users to access this relying party** and click **Next**.
j) Select **Open the Edit Claim rules** dialog for this relying party trust when the wizard closes and click **Close**.

Step 2 Add the following **Claim** rules:

a) Send LDAP Attributes as claims:

- In the **Edit Claim Rules** window, click **Add Rule**.
- Select the **Claim Rule Template** as Send LDAP attributes as **Claims** and click **Next**.
- Enter a **Rule_*Name*** and select **Active Directory** as the Attribute Store.
- Select the reserved User Attribute for storing CiscoAvpair (For Ex: **Department**) as LDAP attribute type and map it to Outgoing Claim Manually Type as **CiscoAvpair**.
- Select **E-Mail-Addresses** on LDAP Attribute and map it to the Outgoing Claim Type **E-mail Address** and click **Finish**.

b) Transform an Incoming Claim:

- Click **Add Rule** again in the **Edit Claim Rules** window, and select **Transform an Incoming Claim as Claim Rule Template** and click **Next**.
- Select **E-Mail Address** as the Incoming claim type.

- Select **Name ID** as Outgoing claim type.
- Select **Transient Identifier** as Outgoing name ID format.

Step 3 To add a cluster of APICs, one can either setup multiple **Relying Party Trusts** or setup single **Relying Party Trust** and add multiple **Relying Party Identifiers** and **SAML Assertion Consumer Endpoints** to it.

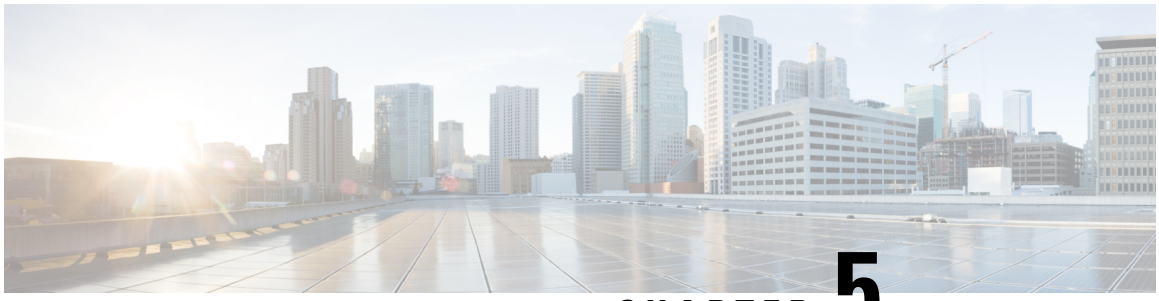
a) Adding other APICs in a cluster with same relying party trusts created above.

1. Navigate to **ADFS Management Console** > **ADFS** > **Trust Relationships** > **Relying Party Trusts** and right-click on **CiscoAPIC** > **Properties**.
2. Click on Identifiers tab and add other APICs in cluster as: *https://<APIC2_hostname>/api/aaaLoginSSO.json*, *https://<APIC3_hostname>/api/aaaLoginSSO.json*
3. Click on **Endpoints** tab and Other two APICs by clicking on **Add SAML**. **Add SAML Post Binding**, Index as 1 and Enter trusted URL as: *https://<APIC2_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>*, and **Add SAML Post Binding** as: *https://<APIC3_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>*.

Step 4 Message and Assertion need to be signed in ADFS from powershell in ADFS server. For Signing Message and Assertion in ADFS Server:

a) Open Windows Powershell (should be run as Administrator) and execute the below command:

b) Set-AdfsRelyingPartyTrust -TargetName **RelyingpartytrustnameOfCiscoAPIC** -SamlResponseSignature **MessageAndAssertion**.



CHAPTER 5

802.1X

This chapter contains the following sections:

- 802.1X Overview, on page 63
- Host Support, on page 63
- Authentication Modes, on page 64
- Guidelines and Limitations, on page 64
- Configuration Overview, on page 65
- Configuring 802.1X Node Authentication Using NX-OS Style CLI, on page 68
- Configuring 802.1X Port Authentication Using the REST API, on page 69
- Configuring 802.1X Node Authentication Using the REST API, on page 69

802.1X Overview

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

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

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

Host Support

The 802.1X feature can restrict traffic on a port with the following modes:

- **Single-host Mode**—Allows traffic from only one endpoint device on the 802.1X port. Once the endpoint device is authenticated, the APIC puts the port in the authorized state. When the endpoint device leaves the port, the APIC put the port back into the unauthorized state. A security violation in 802.1X is defined as a detection of frames sourced from any MAC address other than the single MAC address authorized as a result of successful authentication. In this case, the interface on which this security association violation is detected (EAPOL frame from the other MAC address) will be disabled. Single host mode is

applicable only for host-to-switch topology and when a single host is connected to the Layer 2 (Ethernet access port) or Layer 3 port (routed port) of the APIC.

- **Multi-host Mode**—Allows multiple hosts per port but only the first one gets authenticated. The port is moved to the authorized state after the successful authorization of the first host. Subsequent hosts are not required to be authorized to gain network access once the port is in the authorized state. If the port becomes unauthorized when reauthentication fails or an EAPOL logoff message is received, all attached hosts are denied access to the network. The capability of the interface to shut down upon security association violation is disabled in multiple host mode. This mode is applicable for both switch-to-switch and host-to-switch topologies
- **Multi-Auth Mode**—Allows multiple hosts and all hosts are authenticated separately.

Note Each host must have the same EPG/VLAN information.

- **Multi-Domain Mode**—For separate data and voice domain. For use with IP-Phones.

Authentication Modes

ACI 802.1X supports the following authentication modes:

- **EAP**—The authenticator then sends an EAP-request/identity frame to the supplicant to request its identity (typically, the authenticator sends an initial identity/request frame followed by one or more requests for authentication information). When the supplicant receives the frame, it responds with an EAP-response/identity frame.
- **MAB**—MAC Authentication Bypass (MAB) is supported as the fallback authentication mode. MAB enables port-based access control using the MAC address of the endpoint. A MAB-enabled port can be dynamically enabled or disabled based on the MAC address of the device that connects to it. Prior to MAB, the endpoint's identity is unknown and all traffic is blocked. The switch examines a single packet to learn and authenticate the source MAC address. After MAB succeeds, the endpoint's identity is known and all traffic from that endpoint is allowed. The switch performs source MAC address filtering to help ensure that only the MAB-authenticated endpoint is allowed to send traffic.

Guidelines and Limitations

802.1X port-based authentication has the following configuration guidelines and limitations:

- The Cisco ACI supports 802.1X authentication only on physical ports.
- The Cisco ACI does not support 802.1X authentication on port channels or subinterfaces.
- The Cisco ACI supports 802.1X authentication on member ports of a port channel but not on the port channel itself.
- Member ports with and without 802.1X configuration can coexist in a port channel. However, you must ensure the identical 802.1X configuration on all the member ports in order for channeling to operate with 802.1X

- When you enable 802.1X authentication, supplicants are authenticated before any other Layer 2 or Layer 3 features are enabled on an Ethernet interface.
- 802.1X is supported only on a leaf chassis that is EX or FX type.
- 802.1X is only supported Fabric Access Ports. 802.1X is not supported on Port-Channels, or Virtual-Port-Channels.
- IPv6 is not supported for dot1x clients in the 3.2(1) release.
- While downgrading to earlier releases especially in cases where certain interface config (host mode and auth type) is unsupported in that release, dot1x authentication type defaults to none. Host-mode would need to be manually re-configured to either single host/multi host depending on whatever is desired. This is to ensure that the user configures only the supported modes/auth-types in that release and doesn't run into unsupported scenarios.
- Multi-Auth supports 1 voice client and multiple data clients (all belonging to same data vlan/epg).
- Fail-epg/vlan under 802.1X node authentication policy is a mandatory configuration.
- Multi-domain more than 1 voice and 1 data client puts the port in security disabled state.
- The following platforms are not supported for 802.1X:
  - N9K-C9396PX
  - N9K-M12PQ
  - N9K-C93128TX
  - N9K-M12PQ

Configuration Overview

The 802.1X and RADIUS processes are started only when enabled by APIC. Internally, this means dot1x process is started when 802.1X Inst MO is created and radius process is created when radius entity is created. Dot1x based authentication must be enabled on each interface for authenticating users connected on that interface otherwise the behavior is unchanged.

RADIUS server configuration is done separately from dot1x configuration. RADIUS configuration defines a list of RADIUS servers and a way to reach them. Dot1x configuration contains a reference to RADIUS group (or default group) to use for authentication.

Both 802.1X and RADIUS configuration must be done for successful authentication. Order of configuration is not important but if there is no RADIUS configuration then 802.1X authentication cannot be successful.

Configuring 802.1X Port Authentication Using the APIC GUI

Before you begin

Configure a RADIUS Provider policy.

Procedure

Step 1 On the menu bar, click **Fabric** > **External Access Policies** > **Policies** > **Interface** > **802.1X Port Authentication** and perform the following actions:

a) Right click on **802.1X Port Authentication**, to open **Create 802.1X Port Authentication Policy**.
b) In the **Name** field, enter a name for the policy.
c) In the **Host Mode** field, select the policy mode. The modes are:

- **Multi Auth**—For allowing multiple hosts and all hosts are authenticated separately.

  Note
  Each host must have the same EPG/VLAN information.

- **Multi Domain**—For separate data and voice domain. For use with IP-Phones.
- **Multi Host**—For allowing multiple hosts per port but only the first one gets authenticated.
- **Single Host**—For allowing only one host per port.

d) If your device does not support 802.1X then in the **MAC Auth** field, select **EAP_FALLBACK_MAB** and click **Submit**.

Step 2 To associate the **802.1X Port Authentication Policy** to a Fabric Access Group, navigate to **Fabric** > **External Access Policies** > **Interfaces** > **Leaf Interfaces** > **Policy Groups** > **Leaf Access Port** and perform the following actions:

a) Right click on **Leaf Access Port**, to open **Create Leaf Access Port Policy Group**.
b) In the **Name** field, enter a name for the policy.
c) In the **802.1X Port Authentication Policy** field, select the policy previously created and click **Submit**.

Configuring 802.1X Node Authentication Using the APIC GUI

Before you begin

Configure a RADIUS Provider policy.

Procedure

Step 1 On the menu bar, click **Fabric** > **External Access Policies** > **Policies** > **Switch** > **802.1X Node Authentication** and perform the following actions:

a) Right click on **802.1X Node Authentication**, to open **Create 802.1X Node Authentication Policy**.
b) In the **Name** field, enter a name for the policy.
c) In the **Failed-auth EPG** field, select the tenant, application profile, and EPG to deploy to in the case of failed authentication.
d) In the **Failed-auth VLAN**. select the VLAN to deploy to in the case of failed authentication.

Step 2 To associate the **802.1X Node Authentication Policy** to a Leaf Switch Policy Group, navigate to **Fabric** > **External Access Policies** > **Switches** > **Leaf Switches** > **Policy Groups** and perform the following actions:

a) Right click on **Policy Groups**, to open **Create Access Switch Policy Group**.
b) In the **Name** field, enter a name for the policy.
c) In the **802.1X Node Authentication Policy** field, select the policy previously created and click **Submit**.

Step 3 To associate the **802.1X Node Authentication Policy** to a Leaf Interface Profile, navigate to **Fabric** > **External Access Policies** > **Interfaces** > **Leaf Interfaces** > **Profiles** and perform the following actions:

a) Right click on **Profiles**, to open **Create Leaf Interface Profile**.
b) In the **Name** field, enter a name for the policy.
c) Expand the **Interface Selectors** table, to open the **Create Access Port Selector** dialog box and enter the **Name** and **Interface IDs** information.
d) In the **Interface Policy Group** field, select the policy previously created and click **OK** and **Submit**.

Configuring 802.1X Port Authentication Using the NX-OS Style CLI

Procedure

Step 1 Configure a Policy Group:

Example:

```
apic1# configure
apic1(config)#
apic1(config)# template policy-group mypol
apic1(config-pol-grp-if)# switchport port-authentication mydot1x
apic1(config-port-authentication)# host-mode multi-host
apic1(config-port-authentication)# no shutdown
apic1(config-port-authentication)# exit
apic1(config-pol-grp-if)# exit
```

Step 2 Configure the leaf interface profile:

Example:

```
apic1(config)#
apic1(config)#leaf-interface-profile myprofile
apic1(config-leaf-if-profile)#leaf-interface-group mygroup
apic1(config-leaf-if-group)# interface ethernet 1/10-12
apic1(config-leaf-if-group)# policy-group mypol
apic1(config-leaf-if-group)# exit
apic1(config-leaf-if-profile)# exit
```

Step 3 Configure the leaf profile:

Example:

```
apic1(config)#
apic1(config)# leaf-profile myleafprofile
apic1(config-leaf-profile)# leaf-group myleafgrp
apic1(config-leaf-group)# leaf 101
apic1(config-leaf-group)# exit
```

Step 4 Apply an interface policy on the leaf switch profile:

Example:

```
apic1(config-leaf-profile)# leaf-interface-profile myprofile
apic1(config-leaf-group)# exit
```

Configuring 802.1X Node Authentication Using NX-OS Style CLI

Procedure

Step 1 Configure the radius authentication group:

Example:

```
apic1# configure
apic1(config)#
apic1(config)# aaa group server radius myradiusgrp
apic1(config-radius)#server 192.168.0.100 priority 1
apic1(config-radius)#exit
```

Step 2 Configure node level port authentication policy:

Example:

```
apic1(config)# policy-map type port-authentication mydot1x
apic1(config-pmap-port-authentication)#radius-provider-group myradiusgrp
apic1(config-pmap-port-authentication)#fail-auth-vlan 2001
apic1(config-pmap-port-authentication)#fail-auth-epg tenant tn1 application ap1 epg epg256
apic1(config)# exit
```

Step 3 Configure policy group and specify port authentication policy in the group:

Example:

```
apic1(config)#template leaf-policy-group lpg2
apic1(config-leaf-policy-group)# port-authentication mydot1x
apic1(config-leaf-policy-group)#exit
```

Step 4 Configure the leaf switch profile:

Example:

```
apic1(config)# leaf-profile mylp2
apic1(config-leaf-profile)#leaf-group mylg2
apic1(config-leaf-group)#  leaf-policy-group lpg2
apic1(config-leaf-group)#exit
```

Configuring 802.1X Port Authentication Using the REST API

Procedure

Create a 802.1X port authentication policy:

Example:

```
<polUni>
<infraInfra>
  <l2PortAuthPol adminSt="enabled" annotation="" descr="" dn="uni/infra/portauthpol-test21"
 hostMode="multi-auth" name="test21" nameAlias="" ownerKey="" ownerTag="">
    <l2PortAuthCfgPol annotation="" macAuth="bypass" maxReauthReq="2" maxReq="2"
reAuthPeriod="3600" serverTimeout="30" suppTimeout="30" txPeriod="30"/>
  </l2PortAuthPol>
</infraInfra>
</polUni>

Modify:
<polUni>
<infraInfra>
  <l2PortAuthPol adminSt="enabled" annotation="" descr="" dn="uni/infra/portauthpol-test21"
 hostMode="multi-domain" name="test21" nameAlias="" ownerKey="" ownerTag="" >
    <l2PortAuthCfgPol annotation="" macAuth="eap" maxReauthReq="2" maxReq="2"
reAuthPeriod="3600" serverTimeout="30" suppTimeout="30" txPeriod="30"/>
  </l2PortAuthPol>
</infraInfra>
</polUni>

Delete:
<polUni>
<infraInfra>
  <l2PortAuthPol adminSt="enabled" annotation="" descr="" dn="uni/infra/portauthpol-test21"
 hostMode="multi-host" name="test21" nameAlias="" ownerKey="" ownerTag="" status="deleted">

    <l2PortAuthCfgPol annotation="" macAuth="bypass" maxReauthReq="2" maxReq="2"
reAuthPeriod="3600" serverTimeout="30" suppTimeout="30" txPeriod="30" status="deleted"/>
  </l2PortAuthPol>
</infraInfra>
</polUni>
```

Configuring 802.1X Node Authentication Using the REST API

Procedure

Configure a 802.1X node authentication policy:

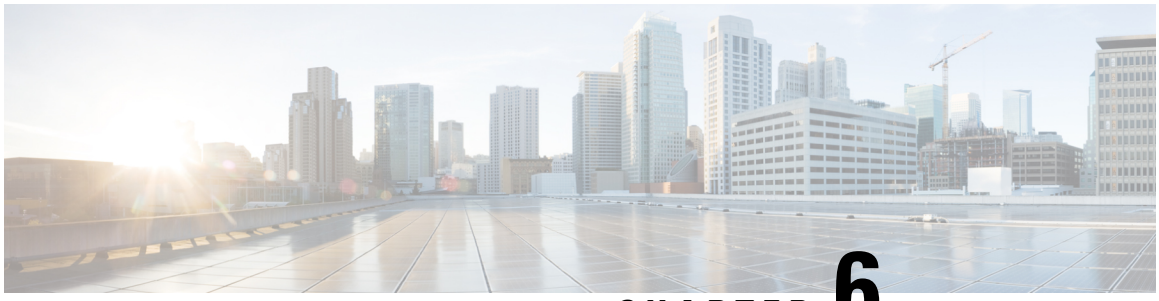
Example:

```
<polUni>
<infraInfra>
```

```
  <l2NodeAuthPol annotation="" descr="" dn="uni/infra/nodeauthpol-802-node-2"
failAuthEpg="tn-t2,ap-ap,epg-epg1" failAuthVlan="vlan-2078" name="802-node-2" nameAlias=""
 ownerKey="" ownerTag="">
<l2RsAaaRadiusProviderGroup annotation=""
tDn="uni/userext/radiusext/radiusprovidergroup-radius-grp"/>
</l2NodeAuthPol>
</infraInfra>
</polUni>

Modify:
<polUni>
<infraInfra>
  <l2NodeAuthPol annotation="" descr="" dn="uni/infra/nodeauthpol-802-node-2"
failAuthEpg="tn-t2,ap-ap,epg-epg1" failAuthVlan="vlan-2066" name="802-node-2" nameAlias=""
 ownerKey="" ownerTag="" status="deleted">
<l2RsAaaRadiusProviderGroup annotation=""
tDn="uni/userext/radiusext/radiusprovidergroup-radius-grp"/>
</l2NodeAuthPol>
</infraInfra>
</polUni>

Delete:
<polUni>
<infraInfra>
  <l2NodeAuthPol annotation="" descr="" dn="uni/infra/nodeauthpol-802-node-2"
failAuthEpg="tn-t2,ap-ap,epg-epg1" failAuthVlan="vlan-2078" name="802-node-2" nameAlias=""
 ownerKey="" ownerTag="" status="deleted">
<l2RsAaaRadiusProviderGroup annotation=""
tDn="uni/userext/radiusext/radiusprovidergroup-radius-grp" status="deleted"/>
</l2NodeAuthPol>
</infraInfra>
</polUni>
```



CHAPTER 6

Port Security

This chapter contains the following sections:

- About Port Security and ACI, on page 71
- Port Security Guidelines and Restrictions, on page 71
- Port Security at Port Level , on page 72
- Port Security and Learning Behavior, on page 75
- Protect Mode, on page 75

About Port Security and ACI

The port security feature protects the ACI fabric from being flooded with unknown MAC addresses by limiting the number of MAC addresses learned per port. The port security feature support is available for physical ports, port channels, and virtual port channels.

Port Security Guidelines and Restrictions

The guidelines and restrictions are as follows:

- Port security is available per port.
- Port security is supported for physical ports, port channels, and virtual port channels (vPCs).
- Static and dynamic MAC addresses are supported.
- MAC address moves are supported from secured to unsecured ports and from unsecured ports to secured ports.
- The MAC address limit is enforced only on the MAC address and is not enforced on a MAC and IP address.
- Port security is not supported with the Fabric Extender (FEX).

Port Security at Port Level

In the APIC, the user can configure the port security on switch ports. Once the MAC limit has exceeded the maximum configured value on a port, all traffic from the exceeded MAC addresses is forwarded. The following attributes are supported:

- **Port Security Timeout**—The current supported range for the timeout value is from 60 to 3600 seconds.
- **Violation Action**—The violation action is available in protect mode. In the protect mode, MAC learning is disabled and MAC addresses are not added to the CAM table. Mac learning is re-enabled after the configured timeout value.
- **Maximum Endpoints**—The current supported range for the maximum endpoints configured value is from 0 to 12000. If the maximum endpoints value is 0, the port security policy is disabled on that port.

Configuring Port Security Using the APIC GUI

Procedure

Step 1 In the menu bar, click **Fabric** > **External Access Policies**, and in the **Navigation** pane, expand **Policies** > **Interface** > **Port Security**.

Step 2 Right-click **Port Security** and click **Create Port Security Policy**.

Step 3 In the **Create Port Security Policy** dialog box, perform the following actions:

a) In the **Name** field, enter a name for the policy.
b) In the **Port Security Timeout** field, choose the desired value for the timeout before re-enabling MAC learning on an interface.
c) In the **Maximum Endpoints** field, choose the desired value for the maximum number of endpoints that can be learned on an interface.
d) In the **Violation Action** field, the option available is **protect**. Click **Submit**.
The port security policy is created.

Step 4

Note
When configuring the interface for a leaf switch, the port security policy can be chosen from the list of available port security policies.

In the **Navigation** pane, click **Fabric** > **Inventory** > **Topology**, and navigate to the desired leaf switch. Choose the appropriate port to configure the interface, and from the port security policy drop-down list, choose the desired port security policy to associate.
This completes the configuration of port security on a port.

Configuring Port Security Using REST API

Procedure

Configure the port security.

Example:

```
<polUni>
 <infraInfra>

    <l2PortSecurityPol name="testL2PortSecurityPol" maximum="10" violation=”protect”
timeout=“300"/>

    <infraNodeP name="test">
        <infraLeafS name="test" type="range">
            <infraNodeBlk name="test" from_="101" to_="102"/>
        </infraLeafS>
        <infraRsAccPortP tDn="uni/infra/accportprof-test"/>
    </infraNodeP>

            <infraAccPortP name="test">
        <infraHPortS name="pselc" type="range">
           <infraPortBlk name="blk"
                                 fromCard="1" toCard="1" fromPort="20" toPort="22">
                 </infraPortBlk>
            <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-testPortG" />
         </infraHPortS>
            </infraAccPortP>

            <infraFuncP>
            <infraAccPortGrp name="testPortG">
               <infraRsL2PortSecurityPol tnL2PortSecurityPolName="testL2PortSecurityPol"/>

                <infraRsAttEntP tDn="uni/infra/attentp-test" />
            </infraAccPortGrp>
            </infraFuncP>

    <infraAttEntityP name="test">
        <infraRsDomP tDn="uni/phys-mininet"/>
    </infraAttEntityP>
 </infraInfra>
</polUni>
```

Configuring Port Security Using the CLI

Procedure

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

	Command or Action	Purpose
	`apic1# configure`	
Step 2	**leaf** *node-id* **Example:** `apic1(config)# leaf 101`	Specifies the leaf to be configured.
Step 3	**interface** *type-or-range* **Example:** `apic1(config-leaf)# interface eth 1/2-4`	Specifies an interface or a range of interfaces to be configured.
Step 4	[**no**] **switchport port-security maximum** *number-of-addresses* **Example:** `apic1(config-leaf-if)# switchport port-security maximum 1`	Sets the maximum number of secure MAC addresses for the interface. The range is 0 to 12000 addresses. The default is 0 address.
Step 5	[**no**] **switchport port-security violation protect** **Example:** `apic1(config-leaf-if)# switchport port-security violation protect`	Sets the action to be taken when a security violation is detected. The **protect** action drops packets with unknown source addresses until you remove a sufficient number of secure MAC addresses to drop below the maximum value.
Step 6	[**no**] **switchport port-security timeout** **Example:** `apic1(config-leaf-if)# switchport port-security timeout 300`	Sets the timeout value for the interface. The range is from 60 to 3600. The default is 60 seconds.

Example

This example shows how to configure port security on an Ethernet interface.

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/2
apic1(config-leaf-if)# switchport port-security maximum 10
apic1(config-leaf-if)# switchport port-security violation protect
apic1(config-leaf-if)# switchport port-security timeout 300
```

This example shows how to configure port security on a port channel.

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po2
apic1(config-leaf-if)# switchport port-security maximum 10
apic1(config-leaf-if)# switchport port-security violation protect
apic1(config-leaf-if)# switchport port-security timeout 300
```

This example shows how to configure port security on a virtual port channel (VPC).

```
apic1# configure
apic1(config)# vpc domain explicit 1 leaf 101 102
apic1(config-vpc)# exit
apic1(config)# template port-channel po4
apic1(config-if)# exit
apic1(config)# leaf 101-102
apic1(config-leaf)# interface eth 1/11-12
apic1(config-leaf-if)# channel-group po4 vpc
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# interface vpc po4
apic1(config-vpc-if)# switchport port-security maximum 10
apic1(config-vpc-if)# switchport port-security violation protect
apic1(config-leaf-if)# switchport port-security timeout 300
```

Port Security and Learning Behavior

For non-vPC ports or port channels, whenever a learn event comes for a new endpoint, a verification is made to see if a new learn is allowed. If the corresponding interface has a port security policy not configured or disabled, the endpoint learning behavior is unchanged with what is supported. If the policy is enabled and the limit is reached, the current supported action is as follows:

- Learn the endpoint and install it in the hardware with a drop action.
- Silently discard the learn.

If the limit is not reached, the endpoint is learned and a verification is made to see if the limit is reached because of this new endpoint. If the limit is reached, and the learn disable action is configured, learning will be disabled in the hardware on that interface (on the physical interface or on a port channel or vPC). If the limit is reached and the learn disable action is not configured, the endpoint will be installed in hardware with a drop action. Such endpoints are aged normally like any other endpoints.

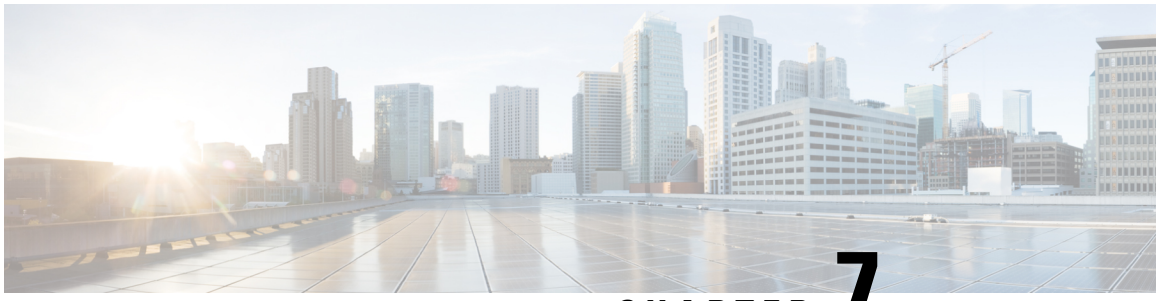
When the limit is reached for the first time, the operational state of the port security policy object is updated to reflect it. A static rule is defined to raise a fault so that the user is alerted. A syslog is also raised when the limit is reached.

In case of vPC, when the MAC limit is reached, the peer leaf switch is also notified so learning can be disabled on the peer. As the vPC peer can be rebooted any time or vPC legs can become unoperational or restart, this state will be reconciled with the peer so vPC peers do not go out of sync with this state. If they get out of sync, there can be a situation where learning is enabled on one leg and disabled on the other leg.

By default, once the limit is reached and learning is disabled, it will be automatically re-enabled after the default timeout value of 60 seconds.

Protect Mode

The protect mode prevents further port security violations from occurring. Once the MAC limit exceeds the maximum configured value on a port, all traffic from excess MAC addresses will be dropped and further learning is disabled.



CHAPTER 7

First Hop Security

This chapter contains the following sections:

- About First Hop Security, on page 77
- ACI FHS Deployment, on page 78
- Guidelines and Limitations, on page 78
- Configuring FHS Using the APIC GUI, on page 79
- Configuring FHS Using the NX-OS CLI, on page 79
- FHS Switch iBASH Commands, on page 86
- Configuring FHS in APIC Using REST API, on page 91

About First Hop Security

First-Hop Security (FHS) features enable a better IPv4 and IPv6 link security and management over the layer 2 links. In a service provider environment, these features closely control address assignment and derived operations, such as Duplicate Address Detection (DAD) and Address Resolution (AR).

The following supported FHS features secure the protocols and help build a secure endpoint database on the fabric leaf switches, that are used to mitigate security threats such as MIM attacks and IP thefts:

- ARP Inspection—allows a network administrator to intercept, log, and discard ARP packets with invalid MAC address to IP address bindings.
- ND Inspection—learns and secures bindings for stateless autoconfiguration addresses in Layer 2 neighbor tables.
- DHCP Inspection—validates DHCP messages received from untrusted sources and filters out invalid messages.
- RA Guard—allows the network administrator to block or reject unwanted or rogue router advertisement (RA) guard messages.
- IPv4 and IPv6 Source Guard—blocks any data traffic from an unknown source.
- Trust Control—a trusted source is a device that is under your administrative control. These devices include the switches, routers, and servers in the Fabric. Any device beyond the firewall or outside the network is an untrusted source. Generally, host ports are treated as untrusted sources.

FHS features provide the following security measures:

- **Role Enforcement**—Prevents untrusted hosts from sending messages that are out the scope of their role.
- **Binding Enforcement**—Prevents address theft.
- **DoS Attack Mitigations**—Prevents malicious end-points to grow the end-point database to the point where the database could stop providing operation services.
- **Proxy Services**—Provides some proxy-services to increase the efficiency of address resolution.

FHS features are enabled on a per tenant bridge domain (BD) basis. As the bridge domain, may be deployed on a single or across multiple leaf switches, the FHS threat control and mitigation mechanisms cater to a single switch and multiple switch scenarios.

ACI FHS Deployment

Most FHS features are configured in a two-step fashion: firstly you define a policy which describes the behavior of the feature, secondly you apply this policy to a "domain" (being the Tenant Bridge Domain or the Tenant Endpoint Group). Different policies that define different behaviors can be applied to different intersecting domains. The decision to use a specific policy is taken by the most specific domain to which the policy is applied.

The policy options can be defined from the Cisco APIC GUI found under the Tenant_*name*>Networking>Protocol Policies>First Hop Security tab.

Guidelines and Limitations

Follow these guidelines and limitations:

- Starting with release 3.1(1), FHS is supported with virtual Endpoints (AVS only).
- FHS feature is not supported when an EPG is deployed with VXLAN encapsulation.
- Any secured endpoint entry in the FHS Binding Table Database in **DOWN** state will get cleared after **18 Hours** of timeout. The entry moves to **DOWN** state when the front panel port where the entry is learned is link down. During this window of **18 Hours**, if the endpoint is moved to a different location and is seen on a different port, the entry will be gracefully moved out of **DOWN** state to **REACHABLE/STALE** as long as the endpoint is reachable from the other port it is moved from.
- When IP Source Guard is enabled, the IPv6 traffic that is sourced using IPv6 Link Local address as IP source address is not subject to the IP Source Guard enforcement (i.e. Enforcement of Source Mac <=> Source IP Bindings secured by IP Inspect Feature). This traffic is permitted by default irrespective of binding check failures.
- FHS is not supported on L3Out interfaces.
- FHS is not supported N9K-M12PQ based TORs.
- FHS in ACI Multi-Site is a site local capability therefore it can only be enabled in a site from the APIC cluster. Also, FHS in ACI Multi-Site only works when the BD and EPG is site local and not stretched across sites. FHS security cannot be enabled for stretched BD or EPGs.
- FHS is not supported on a Layer 2 only bridge domain.

- Enabling FHS feature can disrupt traffic for 50 seconds because the EP in the BD are flushed and EP Learning in the BD is disabled for 50 seconds.

Configuring FHS Using the APIC GUI

Before you begin

- The tenant and Bridge Domain configured.

Procedure

Step 1 On the menu bar, click **Tenants** > **Tenant_name**. In the **Navigation** pane, click **Policies** > **Protocol** > **First Hop Security**. Right click on **First Hop Security** to open **Create Feature Policy** and perform the following actions:

a) In the **Name** field, enter a name for the First Hop Security policy.
b) Verify that the **IP Inspection**, **Source Guard**, and **Router Advertisement** fields are enabled and click **Submit**.

Step 2 In the **Navigation** pane, expand **First Hop Security** and right click on **Trust Control Policies** to open **Create Trust Control Policy** and perform the following actions:

a) In the **Name** field, enter a name for the Trust Control policy.
b) Select the desired features to be allowed on the policy and click **Submit**.

Step 3 (Optional) To apply the Trust Control policy to an EPG, in the **Navigation** pane, expand **Application Profiles** > **Application Profile_*name*** > **Application EPGs** and click on **Application EPG_*name*** and perform the following actions:

a) In the **Work** pane, click on the **General** tab.
b) Click on the down-arrow for **FHS Trust Control Policy** and select the policy you previously created and click **Submit**.

Step 4 In the **Navigation** pane, expand **Bridge Domains** > **Bridge Domain_name** and click on the **Advanced/Troubleshooting** tab and perform the following action:

a) In the **First Hop Security Policy** field, select the policy you just created and click **Submit**. This completes FHS configuration.

Configuring FHS Using the NX-OS CLI

Before you begin

- The tenant and Bridge Domain configured.

Procedure

Step 1 **configure**

Enters configuration mode.

Example:

```
apic1# configure
```

Step 2 Configure FHS policy.

Example:

```
apic1(config)# tenant coke
apic1(config-tenant)# first-hop-security
apic1(config-tenant-fhs)# security-policy pol1
apic1(config-tenant-fhs-secpol)#
apic1(config-tenant-fhs-secpol)# ip-inspection-admin-status enabled-both
apic1(config-tenant-fhs-secpol)# source-guard-admin-status enabled-both
apic1(config-tenant-fhs-secpol)# router-advertisement-guard-admin-status enabled
apic1(config-tenant-fhs-secpol)# router-advertisement-guard
apic1(config-tenant-fhs-raguard)#
apic1(config-tenant-fhs-raguard)# managed-config-check
apic1(config-tenant-fhs-raguard)# managed-config-flag
apic1(config-tenant-fhs-raguard)# other-config-check
apic1(config-tenant-fhs-raguard)# other-config-flag
apic1(config-tenant-fhs-raguard)# maximum-router-preference low
apic1(config-tenant-fhs-raguard)# minimum-hop-limit 10
apic1(config-tenant-fhs-raguard)# maximum-hop-limit 100
apic1(config-tenant-fhs-raguard)# exit
apic1(config-tenant-fhs-secpol)# exit
apic1(config-tenant-fhs)# trust-control tcpol1
pic1(config-tenant-fhs-trustctrl)# arp
apic1(config-tenant-fhs-trustctrl)# dhcpv4-server
apic1(config-tenant-fhs-trustctrl)# dhcpv6-server
apic1(config-tenant-fhs-trustctrl)# ipv6-router
apic1(config-tenant-fhs-trustctrl)#  router-advertisement
apic1(config-tenant-fhs-trustctrl)# neighbor-discovery
apic1(config-tenant-fhs-trustctrl)# exit
apic1(config-tenant-fhs)# exit
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# first-hop-security security-policy pol1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# application ap1
apic1(config-tenant-app)# epg epg1
apic1(config-tenant-app-epg)# first-hop-security trust-control tcpol1
```

Step 3 Show FHS configuration example:

Example:

```
leaf4# show fhs bt all

Legend:
    TR       : trusted-access                      UNRES : unresolved              Age
: Age since creation
    UNTR     : untrusted-access                    UNDTR : undetermined-trust      CRTNG
: creating
    UNKNW    : unknown                             TENTV : tentative               INV
: invalid
```

```
    NDP      : Neighbor Discovery Protocol          STA   : static-authenticated    REACH
: reachable
    INCMP    : incomplete                           VERFY : verify                  INTF
: Interface
    TimeLeft : Remaining time since last refresh    LM    : lla-mac-match           DHCP
: dhcp-assigned

EPG-Mode:
    U : unknown    M : mac    V : vlan    I : ip

BD-VNID             BD-Vlan             BD-Name
15630220            3                   t0:bd200

------------------------------------------------------------------------------------------------------------------
| Origin | IP              | MAC               | INTF   | EPG(sclass)(mode) | Trust-lvl |
State | Age      | TimeLeft |
------------------------------------------------------------------------------------------------------------------
| ARP    | 192.0.200.12    | D0:72:DC:A0:3D:4F | eth1/1 | epg300(49154)(V)  | LM,TR     |
STALE | 00:04:49 | 18:08:13 |
| ARP    | 172.29.205.232  | D0:72:DC:A0:3D:4F | eth1/1 | epg300(49154)(V)  | LM,TR     |
STALE | 00:03:55 | 18:08:21 |
| ARP    | 192.0.200.21    | D0:72:DC:A0:3D:4F | eth1/1 | epg300(49154)(V)  | LM,TR     |
REACH | 00:03:36 | 00:00:02 |
| LOCAL  | 192.0.200.1     | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   | STA       |
REACH | 04:49:41 | N/A      |
| LOCAL  | fe80::200       | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   | STA       |
REACH | 04:49:40 | N/A      |
| LOCAL  | 2001:0:0:200::1 | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   | STA       |
REACH | 04:49:39 | N/A      |
------------------------------------------------------------------------------------------------------------------
```

The trust levels are:

- **TR**— Trusted. Displayed when the endpoint is learned from an EPG where the trust configuration is enabled.
- **UNTR**— Untrusted. Displayed when the endpoint is learned from an EPG where the trust configuration is not enabled.
- **UNDTR**— Undetermined. Displayed in the case of a DHCP relay topology where the DHCP server bridge domain (BD) is on a remote leaf and the DHCP clients are on a local leaf. In this situation, the local leaf will not know whether the DHCP server BD has trust DHCP enabled.

Step 4 Show violations with the different types and reasons example:

Example:

```
leaf4# show fhs violations all

Violation-Type:
    POL  : policy      THR : address-theft-remote
    ROLE : role        TH  : address-theft
    INT  : internal

Violation-Reason:
    IP-MAC-TH   : ip-mac-theft                  OCFG_CHK  : ra-other-cfg-check-fail
ANC-COL     : anchor-collision
    PRF-LVL-CHK : ra-rtr-pref-level-check-fail  INT-ERR   : internal-error
TRUST-CHK   : trust-check-fail
    SRV-ROL-CHK : srv-role-check-fail           ST-EP-COL : static-ep-collision
LCL-EP-COL  : local-ep-collision
    MAC-TH      : mac-theft                     EP-LIM    : ep-limit-reached
MCFG-CHK    : ra-managed-cfg-check-fail
```

```
    HOP-LMT-CHK : ra-hoplimit-check-fail          MOV-COL   : competing-move-collision
RTR-ROL-CHK : rtr-role-check-fail
    IP-TH       : ip-theft

EPG-Mode:
    U : unknown    M : mac    V : vlan    I : ip

BD-VNID             BD-Vlan             BD-Name
15630220            3                   t0:bd200
-------------------------------------------------------------------------------------------------------
| Type | Last-Reason | Proto | IP           | MAC               | Port    | EPG(sclass)(mode)
 | Count |
-------------------------------------------------------------------------------------------------------
| THR  | IP-TH       | ARP   | 192.0.200.21 | D0:72:DC:A0:3D:4F | tunnel5 | epg300(49154)(V)
  | 21    |
-------------------------------------------------------------------------------------------------------
Table Count: 1
```

Step 5 Show FHS configuration:

Example:

```
swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security binding-table

Pod/Node  Type    Family  IP Address            MAC Address        Interface     Level
        State
--------  ------  ------  --------------------  -----------------  ------------
---------------  -----
1/102     local   ipv4    192.0.200.1           00:22:BD:F8:19:FF  vlan3         static-
        reach

authenticated    able
1/102     local   ipv6    fe80::200             00:22:BD:F8:19:FF  vlan3         static-
        reach

authenticated    able
1/102     local   ipv6    2001:0:0:200::1       00:22:BD:F8:19:FF  vlan3         static-
        reach

authenticated    able
1/101     arp     ipv4    192.0.200.23         D0:72:DC:A0:02:61  eth1/2       lla-mac-match
    stale
                                                                                  ,untrusted-

                                                                                     access
1/101     local   ipv4    192.0.200.1           00:22:BD:F8:19:FF  vlan3         static-
        reach

authenticated    able
1/101     nd      ipv6    fe80::d272:dcff:fea0  D0:72:DC:A0:02:61  eth1/2       lla-mac-match
    reach
                          :261                                                    ,untrusted-
      able
                                                                                     access
1/101     nd      ipv6    2001:0:0:200::20      D0:72:DC:A0:02:61  eth1/2       lla-mac-match
    stale
                                                                                  ,untrusted-

                                                                                     access
1/101     nd      ipv6    2001::200:d272:dcff:  D0:72:DC:A0:02:61  eth1/2       lla-mac-match
    stale
                          fea0:261                                                ,untrusted-

                                                                                     access
1/101     local   ipv6    fe80::200             00:22:BD:F8:19:FF  vlan3         static-
```

```
        reach

authenticated    able
1/101     local   ipv6    2001:0:0:200::1       00:22:BD:F8:19:FF  vlan3         static-
        reach

authenticated    able
1/103     local   ipv4    192.0.200.1           00:22:BD:F8:19:FF  vlan4         static-
        reach

authenticated    able
1/103     local   ipv6    fe80::200             00:22:BD:F8:19:FF  vlan4         static-
        reach

authenticated    able
1/103     local   ipv6    2001:0:0:200::1       00:22:BD:F8:19:FF  vlan4         static-
        reach

authenticated    able
1/104     arp     ipv4    192.0.200.10         F8:72:EA:AD:C4:7C  eth1/1       lla-mac-match
    stale

,trusted-access
1/104     arp     ipv4    172.29.207.222       D0:72:DC:A0:3D:4C  eth1/1       lla-mac-match
    stale

,trusted-access
1/104     local   ipv4    192.0.200.1           00:22:BD:F8:19:FF  vlan4         static-
        reach

authenticated    able
1/104     nd      ipv6    fe80::fa72:eaff:fead  F8:72:EA:AD:C4:7C  eth1/1       lla-mac-match
    stale
                            :c47c
,trusted-access
1/104     nd      ipv6    2001:0:0:200::10      F8:72:EA:AD:C4:7C  eth1/1       lla-mac-match
    stale

,trusted-access
1/104     local   ipv6    fe80::200             00:22:BD:F8:19:FF  vlan4         static-
        reach

authenticated    able
1/104     local   ipv6    2001:0:0:200::1       00:22:BD:F8:19:FF  vlan4         static-
        reach

authenticated    able


Pod/Node  Type    IP Address            Creation TS                    Last Refresh TS
            Lease Period
--------  ------  --------------------  -----------------------------
-----------------------------  ------------
1/102     local   192.0.200.1           2017-07-20T04:22:38.000+00:00
2017-07-20T04:22:38.000+00:00
1/102     local   fe80::200             2017-07-20T04:22:56.000+00:00
2017-07-20T04:22:56.000+00:00
1/102     local   2001:0:0:200::1       2017-07-20T04:22:57.000+00:00
2017-07-20T04:22:57.000+00:00
1/101     arp     192.0.200.23          2017-07-27T10:55:20.000+00:00
2017-07-27T16:07:24.000+00:00
1/101     local   192.0.200.1           2017-07-27T10:48:09.000+00:00
2017-07-27T10:48:09.000+00:00
1/101     nd      fe80::d272:dcff:fea0  2017-07-27T10:52:16.000+00:00
```

```
2017-07-27T16:04:29.000+00:00
                    :261
1/101     nd      2001:0:0:200::20      2017-07-27T10:57:32.000+00:00
2017-07-27T16:07:24.000+00:00
1/101     nd      2001::200:d272:dcff:  2017-07-27T11:21:45.000+00:00
2017-07-27T16:07:24.000+00:00
                    fea0:261
1/101     local   fe80::200             2017-07-27T10:48:10.000+00:00
2017-07-27T10:48:10.000+00:00
1/101     local   2001:0:0:200::1       2017-07-27T10:48:11.000+00:00
2017-07-27T10:48:11.000+00:00
1/103     local   192.0.200.1           2017-07-26T22:03:56.000+00:00
2017-07-26T22:03:56.000+00:00
1/103     local   fe80::200             2017-07-26T22:03:57.000+00:00
2017-07-26T22:03:57.000+00:00
1/103     local   2001:0:0:200::1       2017-07-26T22:03:58.000+00:00
2017-07-26T22:03:58.000+00:00
1/104     arp     192.0.200.10          2017-07-27T11:21:13.000+00:00
2017-07-27T16:05:48.000+00:00
1/104     arp     172.29.207.222        2017-07-27T11:54:48.000+00:00
2017-07-27T16:06:38.000+00:00
1/104     local   192.0.200.1           2017-07-27T10:49:13.000+00:00
2017-07-27T10:49:13.000+00:00
1/104     nd      fe80::fa72:eaff:fead  2017-07-27T11:21:13.000+00:00
2017-07-27T16:06:43.000+00:00
                    :c47c
1/104     nd      2001:0:0:200::10      2017-07-27T11:21:13.000+00:00
2017-07-27T16:06:19.000+00:00
1/104     local   fe80::200             2017-07-27T10:49:14.000+00:00
2017-07-27T10:49:14.000+00:00
1/104     local   2001:0:0:200::1       2017-07-27T10:49:15.000+00:00
2017-07-27T10:49:15.000+00:00

swtb23-ifc1#

swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security statistics arp
Pod/Node         : 1/101
Request Received : 4
Request Switched : 2
Request Dropped  : 2
Reply Received   : 257
Reply Switched   : 257
Reply Dropped    : 0

Pod/Node         : 1/104
Request Received : 6
Request Switched : 6
Request Dropped  : 0
Reply Received   : 954
Reply Switched   : 954
Reply Dropped    : 0


swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security statistics dhcpv4
Pod/Node                    : 1/102
Discovery Received          : 5
Discovery Switched          : 5
Discovery Dropped           : 0
Offer Received              : 0
Offer Switched              : 0
Offer Dropped               : 0
Request Received            : 0
Request Switched            : 0
Request Dropped             : 0
```

```
Ack Received                : 0
Ack Switched                : 0
Ack Dropped                 : 0
Nack Received               : 0
Nack Switched               : 0
Nack Dropped                : 0
Decline Received            : 0
Decline Switched            : 0
Decline Dropped             : 0
Release Received            : 0
Release Switched            : 0
Release Dropped             : 0
Information Received        : 0
Information Switched        : 0
Information Dropped         : 0
Lease Query Received        : 0
Lease Query Switched        : 0
Lease Query Dropped         : 0
Lease Active Received       : 0
Lease Active Switched       : 0
Lease Active Dropped        : 0
Lease Unassignment Received : 0
Lease Unassignment Switched : 0
Lease Unassignment Dropped  : 0
Lease Unknown Received      : 0
Lease Unknown Switched      : 0
Lease Unknown Dropped       : 0

swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security statistics
neighbor-discovery
Pod/Node                          : 1/101
Neighbor Solicitation Received  : 125
Neighbor Solicitation Switched  : 121
Neighbor Solicitation Dropped   : 4
Neighbor Advertisement Received : 519
Neighbor Advertisement Switched : 519
Neighbor Advertisement Drop     : 0
Router Solicitation Received    : 4
Router Solicitation Switched    : 4
Router Solicitation Dropped     : 0
Router Adv Received             : 0
Router Adv Switched             : 0
Router Adv Dropped              : 0
Redirect Received               : 0
Redirect Switched               : 0
Redirect Dropped                : 0

Pod/Node                          : 1/104
Neighbor Solicitation Received  : 123
Neighbor Solicitation Switched  : 47
Neighbor Solicitation Dropped   : 76
Neighbor Advertisement Received : 252
Neighbor Advertisement Switched : 228
Neighbor Advertisement Drop     : 24
Router Solicitation Received    : 0
Router Solicitation Switched    : 0
Router Solicitation Dropped     : 0
Router Adv Received             : 53
Router Adv Switched             : 6
Router Adv Dropped              : 47
Redirect Received               : 0
```

```
Redirect Switched               : 0
Redirect Dropped                : 0
```

FHS Switch iBASH Commands

Procedure

Step 1 Show command to display the FHS feature configuration on the BD and the Trust control policy configuration on the EPG:

Example:

```
leaf4# show fhs features all

BD-VNID             BD-Vlan             BD-Name
15630220            4                   t0:bd200
    Feature Policy:
        Feature      Family    Protocol    Operational-State    Options
        ipinspect    IPV4      ARP         UP                   stalelifetime: 180s
        ipinspect    IPV4      DHCP        UP                   -
        ipinspect    IPV4      LOCAL       UP                   -
        ipinspect    IPV4      STATIC      UP                   -
        ipinspect    IPV6      ND          UP                   stalelifetime: 180s
        ipinspect    IPV6      DHCP        UP                   -
        ipinspect    IPV6      LOCAL       UP                   -
        ipinspect    IPV6      STATIC      UP                   -
        raguard      IPV6      -           UP                   ManagedCfgFlag: on
                                                                  OtherCfgFlag: on
                                                                  maxHopLimit: 15
                                                                  minHopLimit: 3
                                                                  routerPref: medium
    ------------------------------------------------------------------------------------
    Trust Policy:
    Epg-id              Epg-type            Epg-name
    49154               Ckt-Vlan            epg300
        Trust-Attribute    Operational-State
        PROTO-ARP          UP
        PROTO-ND           UP
        DHCPV4-SERVER      UP
        DHCPV6-SERVER      UP
        ROUTER             UP
```

Step 2 Show commands to display the FHS secured endpoint database:

Example:

```
leaf1# show fhs bt
all       data      dhcpv4    local     static
arp       detailed  dhcpv6    nd        summary

leaf1# show fhs bt all

Legend:
    DHCP     : dhcp-assigned                        TR    : trusted-access          UNRES
: unresolved
    Age      : Age since creation                   CRTNG : creating                TENTV
: tentative
```

```
    VERFY    : verify                                UNDTR : undetermined-trust      INV
: invalid
    NDP      : Neighbor Discovery Protocol           STA   : static-authenticated    REACH
: reachable
    LM       : lla-mac-match                         UNKNW : unknown                 INTF
: Interface
    TimeLeft : Remaining time since last refresh     INCMP : incomplete              UNTR
: untrusted-access

EPG-Mode:
    U : unknown    M : mac    V : vlan    I : ip

BD-VNID             BD-Vlan             BD-Name
15630220            3                   t0:bd200

_____________________________________________________________________________________________
| Origin | IP                        | MAC               | INTF   | EPG(sclass)(mode) |
Trust-lvl | State | Age      | TimeLeft |
_____________________________________________________________________________________________
| ARP    | 192.0.200.23              | D0:72:DC:A0:02:61 | eth1/2 | epg200(32770)(V)  |
LM,UNTR   | STALE | 00:07:47 | 00:01:33 |
| LOCAL  | 192.0.200.1               | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   | STA
       | REACH | 00:14:58 | N/A      |
| NDP    | fe80::d272:dcff:fea0:261  | D0:72:DC:A0:02:61 | eth1/2 | epg200(32770)(V)  |
LM,UNTR   | STALE | 00:10:51 | 00:00:47 |
| NDP    | 2001:0:0:200::20          | D0:72:DC:A0:02:61 | eth1/2 | epg200(32770)(V)  |
LM,UNTR   | STALE | 00:05:35 | 00:00:42 |
| LOCAL  | fe80::200                 | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   | STA
       | REACH | 00:14:58 | N/A      |
| LOCAL  | 2001:0:0:200::1           | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   | STA
       | REACH | 00:14:57 | N/A      |
_____________________________________________________________________________________________


leaf1# show fhs bt summary all

------------------------------------------------------------------------------
                        FHS Binding Table Summary
------------------------------------------------------------------------------
BD-Vlan: 3        BD-Name: t0:bd200
    Total number of ARP entries       : 1
    Total number of DHCPv4 entries    : 0
    Total number of ND entries        : 2
    Total number of DHCPv6 entries    : 0
    Total number of Data entries      : 0
    Total number of Static entries    : 0
    Total number of Local entries     : 3
    Total number of entries           : 6
------------------------------------------------------------------------------
Total entries across all BDs matching given filters
    Total number of ARP entries       : 1
    Total number of DHCPv4 entries    : 0
    Total number of ND entries        : 2
    Total number of DHCPv6 entries    : 0
    Total number of Data entries      : 0
    Total number of Static entries    : 0
    Total number of Local entries     : 3
    Total number of entries           : 6
------------------------------------------------------------------------------
```

Step 3 Show command to display FHS endpoint violations:

Example:

```
leaf1# show fhs violations all

Violation-Type:
    POL  : policy      THR : address-theft-remote
    ROLE : role        TH  : address-theft
    INT  : internal

Violation-Reason:
    IP-MAC-TH   : ip-mac-theft                    OCFG_CHK  : ra-other-cfg-check-fail
ANC-COL     : anchor-collision
    PRF-LVL-CHK : ra-rtr-pref-level-check-fail    INT-ERR   : internal-error
TRUST-CHK   : trust-check-fail
    SRV-ROL-CHK : srv-role-check-fail             ST-EP-COL : static-ep-collision
LCL-EP-COL  : local-ep-collision
    MAC-TH      : mac-theft                       EP-LIM    : ep-limit-reached
MCFG-CHK    : ra-managed-cfg-check-fail
    HOP-LMT-CHK : ra-hoplimit-check-fail          MOV-COL   : competing-move-collision
RTR-ROL-CHK : rtr-role-check-fail
    IP-TH       : ip-theft

Trust-Level:
    TR   : trusted-access    UNTR : untrusted-access        UNDTR : undetermined-trust
    INV  : invalid           STA  : static-authenticated    LM    : lla-mac-match
    DHCP : dhcp-assigned

EPG-Mode:
    U : unknown    M : mac    V : vlan    I : ip

BD-VNID             BD-Vlan             BD-Name
15630220            4                   t0:bd200
-------------------------------------------------------------------------------------------------------
| Type | Last-Reason | Proto | IP                         | MAC               | Port   |
EPG(sclass)(mode) | Trust-lvl | Count |
-------------------------------------------------------------------------------------------------------
| TH   | IP-TH       | ND    | 2001:0:0:200::20           | D0:72:DC:A0:3D:4F | eth1/1 |
epg300(49154)(V)  | LM,UNTR   | 2     |
| POL  | HOP-LMT-CHK | RD    | fe80::fa72:eaff:fead:c47c | F8:72:EA:AD:C4:7C | eth1/1 |
epg300(49154)(V)  | LM,TR     | 2     |
-------------------------------------------------------------------------------------------------------
Table Count: 2
```

Step 4 Show command to display FHS control packet forwarding counters:

Example:

```
leaf1# show fhs counters
all     arp     dhcpv4  dhcpv6  nd
leaf4# show fhs counters all

BD-VNID             BD-Vlan             BD-Name
15630220            4                   t0:bd200
 ---------------------------------------------------------------------------------------
| Counter Type           |            Received |            Switched |             Dropped
 |
|---------------------------------------------------------------------------------------|
| Arp Request            |                   6 |                   6 |                   0
 |
| Arp Reply              |                  94 |                  94 |                   0
 |
---------------------------------------------------------------------------------------
| Dhcpv4 Ack             |                   0 |                   0 |                   0
 |
| Dhcpv4 Decline         |                   0 |                   0 |                   0
 |
| Dhcpv4 Discover        |                   0 |                   0 |                   0
```

```
 |
| Dhcpv4 Inform          |                  0 |                  0 |                  0
 |
| Dhcpv4 Leaseactive     |                  0 |                  0 |                  0
 |
| Dhcpv4 Leasequery      |                  0 |                  0 |                  0
 |
| Dhcpv4 Leaseunassigned |                  0 |                  0 |                  0
 |
| Dhcpv4 Leaseunknown    |                  0 |                  0 |                  0
 |
| Dhcpv4 Nack            |                  0 |                  0 |                  0
 |
| Dhcpv4 Offer           |                  0 |                  0 |                  0
 |
| Dhcpv4 Release         |                  0 |                  0 |                  0
 |
| Dhcpv4 Request         |                  0 |                  0 |                  0
 |
-------------------------------------------------------------------------------------------
| Dhcpv6 Advertise       |                  0 |                  0 |                  0
 |
| Dhcpv6 Confirm         |                  0 |                  0 |                  0
 |
| Dhcpv6 Decline         |                  0 |                  0 |                  0
 |
| Dhcpv6 Informationreq  |                  0 |                  0 |                  0
 |
| Dhcpv6 Rebind          |                  0 |                  0 |                  0
 |
| Dhcpv6 Reconfigure     |                  0 |                  0 |                  0
 |
| Dhcpv6 Relayforw       |                  0 |                  0 |                  0
 |
| Dhcpv6 Relayreply      |                  0 |                  0 |                  0
 |
| Dhcpv6 Release         |                  0 |                  0 |                  0
 |
| Dhcpv6 Renew           |                  0 |                  0 |                  0
 |
| Dhcpv6 Reply           |                  0 |                  0 |                  0
 |
| Dhcpv6 Request         |                  0 |                  0 |                  0
 |
| Dhcpv6 Solicit         |                  0 |                  0 |                  0
 |
-------------------------------------------------------------------------------------------
| Nd Na                  |                 18 |                 18 |                  0
 |
| Nd Ns                  |                 26 |                 22 |                  4
 |
| Nd Ra                  |                 11 |                  6 |                  5
 |
| Nd Redirect            |                  0 |                  0 |                  0
 |
| Nd Rs                  |                  0 |                  0 |                  0
 |
-------------------------------------------------------------------------------------------
```

Step 5 Display FHS secured endpoint database from the NxOS memory:

Example:

```
leaf1# vsh -c 'show system internal fhs bt'

Binding Table has 7 entries, 4 dynamic

Codes:
L - Local         S - Static        ND - Neighbor Discovery   ARP - Address Resolution
Protocol
DH4 - IPv4 DHCP   DH6 - IPv6 DHCP   PKT - Other Packet        API - API created


Preflevel flags (prlvl):
0001: MAC and LLA match     0002: Orig trunk            0004: Orig access
0008: Orig trusted trunk    0010: Orig trusted access   0020: DHCP assigned
0040: Cga authenticated     0080: Cert authenticated    0100: Statically assigned

EPG types:
V - Vlan Based EPG      M - MAC Based EPG       I - IP Based EPG

-------------------------------------------------------------------------------------------------------------
| Code | Network Layer Address                    | Link Layer Address | Interface      |
Vlan | Epg                | prlvl | Age        | State      | Time left       |
-------------------------------------------------------------------------------------------------------------
| ARP  | 172.29.207.222                           | d0:72:dc:a0:3d:4c  | Eth1/1         |
   4 | 0x40000c002 (V)    | 0011  | 29 s       | STALE      | 157 s           |
| L    | 192.0.200.1                              | 00:22:bd:f8:19:ff  | Vlan4          |
   4 | 0x400004003 (I)    | 0100  | 55 mn      | REACHABLE  |                 |
| ARP  | 192.0.200.10                             | f8:72:ea:ad:c4:7c  | Eth1/1         |
   4 | 0x40000c002 (V)    | 0011  | 156 s      | STALE      | 30 s            |
| L    | 2001:0:0:200::1                          | 00:22:bd:f8:19:ff  | Vlan4          |
   4 | 0x400004003 (I)    | 0100  | 55 mn      | REACHABLE  |                 |
| ND   | 2001:0:0:200::10                         | f8:72:ea:ad:c4:7c  | Eth1/1         |
   4 | 0x40000c002 (V)    | 0011  | 143 s      | STALE      | 47 s            |
| L    | fe80::200                                | 00:22:bd:f8:19:ff  | Vlan4          |
   4 | 0x400004003 (I)    | 0100  | 55 mn      | REACHABLE  |                 |
| ND   | fe80::fa72:eaff:fead:c47c                | f8:72:ea:ad:c4:7c  | Eth1/1         |
   4 | 0x40000c002 (V)    | 0011  | 176 s      | STALE      | 11 s            |
-------------------------------------------------------------------------------------------------------------
```

Step 6 Display FHS feature configuration from the NX-OS FHS process internal memory:

Example:

```
leaf4# vsh -c 'show system internal fhs pol'

Target           Type  Policy           Feature       Target-Range Sub-Feature
epg 0x40000c002  EPG   epg 0x40000c002  Trustctrl      vlan 4      Device-Roles:
DHCPv4-Server, DHCPv6-Server, Router
                                                                    Protocols: ARP ND
vlan 4           VLAN  vlan 4           IP inspect     vlan all    Protocols: ARP, DHCPv4,
 ND, DHCPv6,
vlan 4           VLAN  vlan 4           RA guard       vlan all    Min-HL:3, Max-HL:15,
M-Config-flag:Enable,On
                                                                   O-Config-flag:Enable,On,
 Router-Pref:medium
```

Step 7 Display FHS secured endpoint database from the NX-OS shared database:

Example:

```
leaf1# vsh -c 'show system internal fhs sdb bt'

Preflevel flags (preflvl):
0001: MAC and LLA match     0002: Orig trunk            0004: Orig access
0008: Orig trusted trunk    0010: Orig trusted access   0020: DHCP assigned
0040: Cga authenticated     0080: Cert authenticated    0100: Statically assigned
```

```
Origin      Zone ID     L3 Address                                  MAC Address         VLAN
ID  EPG ID               If-name         Preflvl  State
----------  ----------  ------------------------------------------  ------------------  -------
  ------------------  --------------  -------  ---------
ARP         0x4         172.29.207.222                              d0:72:dc:a0:3d:4c   4
    0x40000c002         Eth1/1          0011     STALE
L           0x4         192.0.200.1                                 00:22:bd:f8:19:ff   4
    0x400004003         Vlan4           0100     REACHABLE
ARP         0x4         192.0.200.10                                f8:72:ea:ad:c4:7c   4
    0x40000c002         Eth1/1          0011     REACHABLE
L           0x4         2001:0:0:200::1                             00:22:bd:f8:19:ff   4
    0x400004003         Vlan4           0100     REACHABLE
ND          0x4         2001:0:0:200::10                            f8:72:ea:ad:c4:7c   4
    0x40000c002         Eth1/1          0011     STALE
L           0x80000004  fe80::200                                   00:22:bd:f8:19:ff   4
    0x400004003         Vlan4           0100     REACHABLE
ND          0x80000004  fe80::fa72:eaff:fead:c47c                   f8:72:ea:ad:c4:7c   4
    0x40000c002         Eth1/1          0011     STALE
```

Step 8 Display FHS feature configurations from the NxOS shared database:

Example:

```
leaf1# vsh -c 'show system internal fhs sdb pol'
Policies:

IP inspect      Vlan 4                    Protocols:ARP DHCPv4 ND DHCPv6
RA guard        Vlan 4                    Min-HL:3 Max-HL:15 M-Config-Flag:enable,on
O-Config-Flag:enable,on Router-Pref:medium
Trustctrl       Epg 0x40000c002           Vlan:4
                                          Device-Roles:DHCPv4-Server DHCPv6-Server Router
                                          Protocols:ARP ND
```

Step 9 Show command to clear a secured database endpoint entry:

Example:

```
leaf1# vsh -c 'clear system internal fhs bt ipv4 172.29.207.222'
```

Configuring FHS in APIC Using REST API

Before you begin

- The tenant and bridge domain must be configured.

Procedure

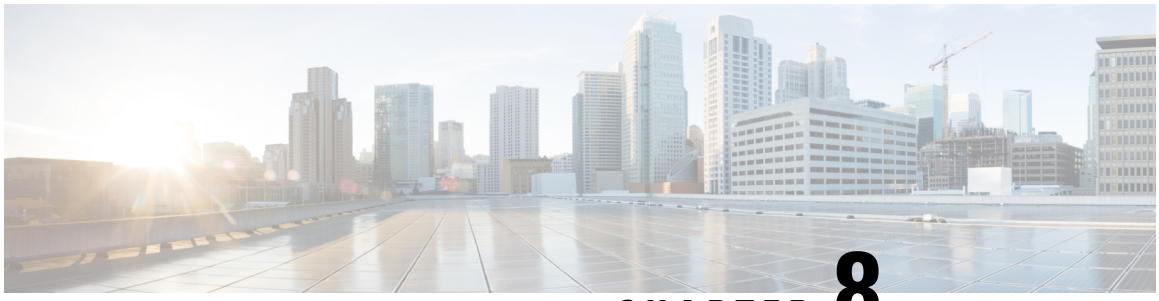
Configure the FHS and Trust Control policies.

Example:

```
<polUni>
    <fvTenant name="Coke">
       <fhsBDPol name="bdpol5" ipInspectAdminSt="enabled-ipv6" srcGuardAdminSt="enabled-both"
 raGuardAdminSt="enabled" status="">
```

```
          <fhsRaGuardPol name="raguard5" managedConfigCheck="true" managedConfigFlag="true"
 otherConfigCheck="true" otherConfigFlag="true" maxRouterPref="medium" minHopLimit="3"
maxHopLimit="15" status=""/>
        </fhsBDPol>
        <fvBD name="bd3">
            <fvRsBDToFhs tnFhsBDPolName="bdpol5" status=""/>
        </fvBD>
    </fvTenant>
</polUni>

<polUni>
<fvTenant name="Coke">
        <fhsTrustCtrlPol name="trustctrl5" hasDhcpv4Server="true" hasDhcpv6Server="true"
hasIpv6Router="true" trustRa="true" trustArp="true" trustNd="true" />
        <fvAp name="wwwCokecom3">
           <fvAEPg name="test966">
           <fvRsTrustCtrl tnFhsTrustCtrlPolName="trustctrl5" status=""/>
        </fvAEPg>
        </fvAp>
</fvTenant>
</polUni>
```



CHAPTER 8

Protocol Authentication

This chapter contains the following sections:

- COOP, on page 93
- EIGRP, on page 95

COOP

Overview

Council of Oracle Protocol (COOP) is used to communicate the mapping information (location and identity) to the spine proxy. A leaf switch forwards endpoint address information to the spine switch 'Oracle' using Zero Message Queue (ZMQ). COOP running on the spine nodes will ensure all spine nodes maintain a consistent copy of endpoint address and location information and additionally maintain the distributed hash table (DHT) repository of endpoint identity to location mapping database.

COOP data path communication provides high priority to transport using secured connections. COOP is enhanced to leverage the MD5 option to protect COOP messages from malicious traffic injection. The APIC controller and switches support COOP protocol authentication.

COOP protocol is enhanced to support two ZMQ authentication modes: strict and compatible.

- Strict mode: COOP allows MD5 authenticated ZMQ connections only.
- Compatible mode: COOP accepts both MD5 authenticated and non-authenticated ZMQ connections for message transportation.

Using COOP with Cisco APIC

To support COOP Zero Message Queue (ZMQ) authentication support across the Cisco Application Centric Infrastructure (ACI) fabric, the Application Policy Infrastructure Controller (APIC) supports the MD5 password and also supports the COOP secure mode.

COOP ZMQ Authentication Type Configuration—A new managed object, `coop:AuthP`, is added to the Data Management Engine (DME)/COOP database (coop/inst/auth). The default value for the attribute type is "compatible", and users have the option to configure the type to be "strict".

COOP ZMQ Authentication MD5 password—The APIC provides a managed object (`fabric:SecurityToken`), that includes an attribute to be used for the MD5 password. An attribute in this managed object, called "token", is a string that changes every hour. COOP obtains the notification from the DME to update the password for ZMQ authentication. The attribute token value is not displayed.

Guidelines and Limitations

Follow these guidelines and limitations:

- During an ACI fabric upgrade, the COOP strict mode is disallowed until all switches are upgraded. This protection prevents the unexpected rejection of a COOP connection that could be triggered by prematurely enabling the strict mode.

Configuring COOP Authentication Using the APIC GUI

Procedure

Step 1 On the menu bar, choose **System** > **System Settings**.

Step 2 In the **Navigation** pane, click on **COOP Group**.

Step 3 In the **Work** pane, under the **Policy Property** area in the **Type** field, choose the desired type from the **Compatible Type** and **Strict Type** options.

Step 4 Click **Submit**.
This completes the COOP authentication policy configuration.

Configuring COOP Authentication Using the Cisco NX-OS-Style CLI

Procedure

Configure the COOP authentication policy using the strict mode option.

Example:

```
apic1# configure
apic1(config)# coop-fabric
apic1(config-coop-fabric)# authentication type ?
compatible  Compatible type
strict      Strict type
apic101-apic1(config-coop-fabric)# authentication type strict
```

Configuring COOP Authentication Using the REST API

Procedure

Configure a COOP authentication policy.

In the example, the strict mode is chosen.

Example:

```
https://172.23.53.xx/api/node/mo/uni/fabric/pol-default.xml

<coopPol type="strict">
</coopPol>
```

EIGRP

Overview

EIGRP combines the benefits of distance vector protocols with the features of link-state protocols. EIGRP sends out periodic Hello messages for neighbor discovery. Once EIGRP learns a new neighbor, it sends a one-time update of all the local EIGRP routes and route metrics. The receiving EIGRP router calculates the route distance based on the received metrics and the locally assigned cost of the link to that neighbor. After this initial full route table update, EIGRP sends incremental updates to only those neighbors affected by the route change. This process speeds convergence and minimizes the bandwidth used by EIGRP.

For Cisco APIC, EIGRP Authentication uses Route-map's keychain infrastructure for MD5 Authentication. It takes two parameters to configure Authentication between two EIGRP peers. The parameters are:

- Mode
- Keychain

Guidelines and Limitations

Follow these guidelines and limitations:

- Only MD5 Authentication is supported. Keychain is the Keychain name configured under RPM.
- When there is authentication mismatch between two EIGRP peers, then neighborship flaps. The reason for the flap can be verified in **`show eigrp internal event-history syslog`**.

Configuring EIGRP Authentication Using the APIC GUI

Procedure

Step 1 On the menu bar, choose **Tenant***tenant-name* .

Step 2 In the **Navigation** pane, expand **Policies** > **Protocol** > **EIGRP**.

Step 3 Expand **EIGRP** and right-click **EIGRP KeyChains** to open **Create Keychain Policy** and perform the following actions:

a) In the **Name** field, enter a name for the policy.
b) In the **KeyID** field, enter a key ID number.
c) In the **Preshared key** field, enter the preshared key information.
d) Optional. In the **Start Time** and **End Time** fields, enter a time.

Step 4 In the **Navigation** pane, right-click on **EIGRP Interface** and perform the following actions:

a) In the **Authentication** field, click the box to enable.
b) In the **Key Chain Policy** field, select the policy just created from the drop-down and click **Submit**.

Configuring EIGRP Authentication Using the NX-OS CLI

Procedure

Step 1 Configure keychain-policy and key-policy under Tenant.

Example:

```
tenant T1
keychain-policy KeyChainPol
key-policy 2
```

Step 2 Optional. Configure Start time.

Example:

```
startime 2018-11-01T08:39:27.000+00:00
exit
```

Step 3 Enter the leaf configuration from APIC. Enable authentication in the interface and configure the key-chain policy.

Example:

```
IFC1(config-leaf)# show run
# Command: show running-config leaf 104
# Time: Thu Nov 8 12:05:45 2018
leaf 104
interface ethernet 1/2.45
vrf member tenant T1 vrf V1 l3out L3Out
ip router eigrp authentication keychain-policy KeyChainPol
ip router eigrp authentication enable
!
```

```
ipv6 router eigrp authentication keychain-policy KeyChainPol
ipv6 router eigrp authentication enable
exit
```

Step 4 To verfiy EIGRP configuration:

Example:

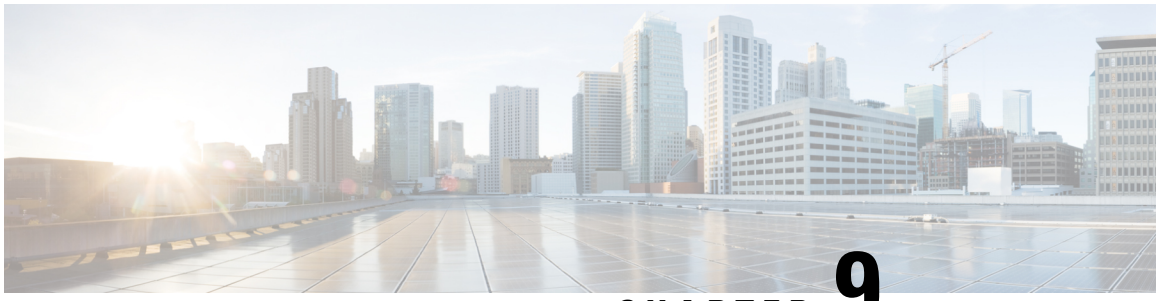
```
fav-blr4-ls-leaf4# show ip eigrp interfaces eth1/2.17
EIGRP interfaces for process 1 VRF T1:V1
Xmit Queue Mean Pacing Time Multicast Pending
Interface Peers Un/Reliable SRTT Un/Reliable Flow Timer Routes
eth1/2.17 0 0/0 0 0/0 50 0
Hello interval is 5 sec
Holdtime interval is 15 sec
Next xmit serial: 0
Un/reliable mcasts: 0/3 Un/reliable ucasts: 6/4
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 1
Retransmissions sent: 0 Out-of-sequence rcvd: 0
Classic/wide metric peers: 0/0
Authentication mode is md5, key-chain is T1:KeyChainPol
ifav-blr4-ls-leaf4#
```

Step 5 For troubleshooting it on a switch, following CLIs can be used. And EIGRP Auth is supported on both IPv4 and IPv6 address families.

Example:

```
(none)# show ip eigrp interface vrf all
EIGRP interfaces for process 100 VRF pepsi
Xmit Queue Mean Pacing Time Multicast Pending
Interface Peers Un/Reliable SRTT Un/Reliable Flow Timer Routes
eth1/1 1 0/0 207 0/0 828 0
Hello interval is 10 sec
Holdtime interval is 15 sec
Next xmit serial: 0
Un/reliable mcasts: 0/7 Un/reliable ucasts: 21/18
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
Retransmissions sent: 4 Out-of-sequence rcvd: 2
Classic/wide metric peers: 0/1
Authentication mode is md5, key-chain is eigrp-auth

(none)# show ipv6 eigrp interface vrf pepsi
IPv6-EIGRP interfaces for process 100 VRF pepsi
Xmit Queue Mean Pacing Time Multicast Pending
Interface Peers Un/Reliable SRTT Un/Reliable Flow Timer Routes
eth1/1 0 0/0 0 0/0 0 0
Hello interval is 10 sec
Holdtime interval is 15 sec
Next xmit serial: 0
Un/reliable mcasts: 0/0 Un/reliable ucasts: 0/0
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
Retransmissions sent: 0 Out-of-sequence rcvd: 0
Classic/wide metric peers: 0/0
Authentication mode is md5, key-chain is eigrp-auth
```



CHAPTER 9

Control Plane Traffic

- About Control Plane Policing, on page 99
- About CoPP Prefilters, on page 107

About Control Plane Policing

Control plane policing (CoPP) protects the control plane, which ensures network stability, reachability, and packet delivery.

This feature allows specification of parameters, for each protocol that can reach the control processor to be rate-limited using a policer. The policing is applied to all traffic destined to any of the IP addresses of the router or Layer 3 switch. A common attack vector for network devices is the denial-of-service (DoS) attack, where excessive traffic is directed at the device interfaces.

The Cisco Application Centric Infrastructure (ACI) leaf and spine switch NX-OS 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 of a Cisco ACI leaf and spine switch CPU or CPU itself.

The supervisor module of Cisco ACI leaf and spine switch switches divides the traffic that it manages into two 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.

The Cisco ACI leaf and spine switch supervisor module has a 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 ACI fabric. Another example is a DoS attack on the Cisco ACI leaf and spine switch supervisor module that could generate IP traffic streams to the control plane at a very high rate, forcing the control plane to spend a large amount of time in handling these packets and preventing the control plane from processing genuine traffic.

Examples of DoS attacks are as follows:

- Internet Control Message Protocol (ICMP) echo requests
- IP fragments
- TCP SYN flooding

These attacks can impact the device performance and have the following negative effects:

- Reduced service quality (such as poor voice, video, or critical applications traffic)
- High route processor or switch processor CPU utilization
- Route flaps due to loss of routing protocol updates or keepalives
- Processor resource exhaustion, such as the memory and buffers
- Indiscriminate drops of incoming packets

Note Cisco ACI leaf and spine switches are by default protected by CoPP with default settings. This feature allows for tuning the parameters on a group of nodes based on customer needs.

Control Plane Protection

To protect the control plane, the Cisco NX-OS running on Cisco ACI leaf and spine switches 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.

Control Plane Packet Types:

Different types of packets can reach the control plane:

- **Receive Packets**—Packets that have the destination address of a router. The destination address can be a Layer 2 address (such as a router MAC address) or a Layer 3 address (such as the IP address of a router interface). These packets include router updates and keepalive messages. Multicast packets can also be in this category where packets are sent to multicast addresses that are used by a router.
- **Exception Packets**—Packets that need special handling by the supervisor module. For example, if a destination address is not present in the Forwarding Information Base (FIB) and results in a miss, the supervisor module sends an ICMP unreachable packet back to the sender. IP packet with IP options are dropped by the supervisor.
- **Redirect Packets**—Packets that are redirected to the supervisor module. Features such as Dynamic Host Configuration Protocol (DHCP) snooping or dynamic Address Resolution Protocol (ARP) inspection redirect some packets to the supervisor module.
- **Glean Packets**—If a Layer 2 MAC address for a destination IP address is not present in the FIB, the supervisor module receives the packet and sends an ARP request to the host.

All of these different packets could be maliciously used to attack the control plane and overwhelm the Cisco ACI fabric. CoPP classifies these packets to different classes and provides a mechanism to individually control the rate at which the Cisco ACI leaf and spine switch supervisor module receives these packets.

Classification for CoPP:

For effective protection, the Cisco ACI leaf and spine switch NX-OS classifies the packets that reach the supervisor modules to allow you to apply different rate controlling policies based on the type of the packet. For example, you might want to be less strict with a protocol packet such as Hello messages but more strict with a packet that is sent to the supervisor module because the IP option is set.

Available Protocols:

Protocol	Description
Glean	With this protocol, when the bridge domain is in proxy mode, unknown unicast traffic received by the leaf switch is sent to the hardware proxy (the spine switch). The spine switch changes the eth-type of the packet to a special eth-type (0xfff2). When these packets reach the leaf switches through the fabric ports, the packets are classified under glean. The packets are sent to the leaf switch's CPU, and the leaf switch's CPU generates an ARP request for the connected external devices.
ToR Glean	ToR glean activates when an endpoint moves or is cleared because of link flap and does not update the source leaf switch's remote IP address endpoint entry. A packet egresses the source leaf switch with the destination leaf switch's TEP address. On the destination leaf switch, because of the missing local IP address entry, the packet gets sent to the leaf switch CPU to generate an ARP request for those IP addresses. These packets are classified under ToR glean.

Rate Controlling Mechanisms:

Once the packets are classified, the Cisco ACI leaf and spine switch NX-OS has different mechanisms to control the rate at which packets arrive at the supervisor module.

You can configure the following parameters for policing:

- **Committed information rate (CIR)**—Desired bandwidth, specified as a bit rate or a percentage of the link rate.
- **Committed burst (BC)**—Size of a traffic burst that can exceed the CIR within a given unit of time and not impact scheduling.

Default Policing Policies:

When the Cisco ACI leaf and spine switch is bootup, the platform setup pre-defined CoPP parameters for different protocols are based on the tests done by Cisco.

Guidelines and Limitations for CoPP

CoPP has the following configuration guidelines and limitations:

- We recommend that you use the 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.
- 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 ACI Leaf/Spine and require a console connection.
- Do not mis-configure CoPP pre-filter entries. CoPP pre-filter entries might impact connectivity to multi-pod configurations, remote leaf switches, and Cisco ACI Multi-Site deployments.
- You can use the APIC UI to be able to tune the CoPP parameters.
- Per interface per protocol is only supported on Leaf switches.
- FEX ports are not supported on per interface per protocol.
- For per interface per protocol the supported protocols are; ARP, ICMP, CDP, LLDP, LACP, BGP, STP, BFD, and OSPF.
- The TCAM entry maximum for per interface per protocol is 256. Once the threshold is exceeded a fault will be raised.

Configuring CoPP Using the APIC GUI

Procedure

Step 1 On the menu bar, click **Fabric** > **External Access Policies**.

Step 2 In the **Navigation** pane, expand **Policies** > **Switch** > **CoPP Leaf**, right click **Create Profiles for CoPP To Be Applied At The Leaf Level** dialog box to perform the following actions in the **Create Profiles for CoPP To Be Applied At The Leaf Level** dialog box:

a) In the **Name** field, add a policy name.

b) In the **Type of Profile** field, select the profile type.

Note
Select **CoPP has custom values** if you wish to set each protocol separately. If you do not select a profile type then default values are applied.

c) Click **Submit** to create the policy.

Step 3 In the **Navigation** pane, expand **Switches** > **Leaf Switches** > **Policy Groups**, right click **Create Access Switch Policy Group** dialog box to perform the following actions in the **Create Access Switch Policy Group** dialog box:

a) In the **Name** field, add a policy name.

b) In the **COPP Leaf Policy** field, select the policy previously created.

c) Click **Submit**.

Step 4 In the **Navigation** pane, expand **Switches** > **Leaf Switches** > **Profiles**, right click **Create Leaf Profile** dialog box to perform the following actions in the **Create Leaf Profile** dialog box:

a) In the **Name** field, add a profile name.
b) Expand the **Leaf Selectors** table, add the Leaf information in the **Name** and **Blocks** fields, and select the **Policy Group** previously created.
c) Click **Next** and **Finish** to complete CoPP configuration.

Configuring CoPP Using the Cisco NX-OS CLI

Procedure

Step 1 Configure a CoPP leaf profile:

Example:

```
# configure copp Leaf Profile
apic1(config)# policy-map type control-plane-leaf leafProfile
apic1(config-pmap-copp-leaf)# profile-type custom
apic1(config-pmap-copp-leaf)# set arpRate 786
# create a policy group to be applied on leaves
apic1(config)# template leaf-policy-group coppForLeaves
apic1(config-leaf-policy-group)# copp-aggr leafProfile
apic1(config-leaf-policy-group)# exit
# apply the leaves policy group on leaves
apic1(config)# leaf-profile applyCopp
apic1(config-leaf-profile)# leaf-group applyCopp
apic1(config-leaf-group)# leaf 101-102
apic1(config-leaf-group)# leaf-policy-group coppForLeaves
```

Step 2 Configure a CoPP Spine profile:

Example:

```
# configure copp Spine Profile
apic1(config)# policy-map type control-plane-spine spineProfile
apic1(config-pmap-copp-spine)# profile-type custom
apic1(config-pmap-copp-spine)# set arpRate 786
# create a policy group to be applied on spines
apic1(config)# template leaf-policy-group coppForSpines
apic1(config-spine-policy-group)# copp-aggr spineProfile
apic1(config-spine-policy-group)# exit
# apply the spine policy group on spines
apic1(config)# spine-profile applyCopp
apic1(config-spine-profile)# spine-group applyCopp
apic1(config-spine-group)# spine 201-202
apic1(config-spine-group)# spine-policy-group coppForSpines
```

Configuring CoPP Using the REST API

Procedure

Step 1 Configure a CoPP leaf profile:

Example:

```
<!-- api/node/mo/uni/.xml -->
<infraInfra>
  <coppLeafProfile type="custom" name="mycustom">                  <!-- define copp leaf
profile -->
    <coppLeafGen1CustomValues bgpBurst="150" bgpRate="300"/>
  </coppLeafProfile>
  <infraNodeP name="leafCopp">
    <infraLeafS name="leafs" type="range">
      <infraNodeBlk name="leaf1" from_="101" to_="101"/>
      <infraNodeBlk name="leaf3" from_="103" to_="103"/>
      <infraRsAccNodePGrp tDn="uni/infra/funcprof/accnodepgrp-myLeafCopp"/>
    </infraLeafS>
  </infraNodeP>
  <infraFuncP>
    <infraAccNodePGrp name="myLeafCopp">
      <infraRsLeafCoppProfile tnCoppLeafProfileName="mycustom"/>   <!-- bind copp leaf
policy to leaf </infraAccNodePGrp>
profile -->
  </infraFuncP>
</infraInfra>
```

Step 2 Configure a CoPP spine profile:

Example:

```
<!-- api/node/mo/uni/.xml -->
<infraInfra>
  <coppSpineProfile type="custom" name="mycustomSpine">             <!-- define copp leaf
profile -->
    <coppSpineGen1CustomValues bgpBurst="150" bgpRate="300"/>
  </coppSpineProfile>
  <infraSpineP name="spineCopp">
    <infraSpineS name="spines" type="range">
      <infraNodeBlk name="spine1" from_="104" to_="104"/>
      <infraRsSpineAccNodePGrp tDn="uni/infra/funcprof/spaccnodepgrp-mySpineCopp"/>
    </infraSpineS>
  </infraSpineP>
  <infraFuncP>
    <infraSpineAccNodePGrp name="mySpineCopp">
     <infraRsSpineCoppProfile tnCoppSpineProfileName="mycustomSpine"/> <!-- bind copp spine
 policy to
    </infraSpineAccNodePGrp>                                                spine profile
 -->
  </infraFuncP>
</infraInfra>
```

Viewing CoPP Statistics Using the GUI

Fine tuning CoPP requires knowing the number of packets dropped/allowed by a given protocol on a given node. The information can be viewed in the GUI using the procedure below:

Procedure

On the menu bar, click **Fabric** > **Inventory** > **Pod*number*** > **Node*name*** > **Control Plane Statistics** > **default**, select from the list of classes to configure the statistics display format.

You can collect statistics about the number of packets allowed or dropped by CoPP.

Configuring Per Interface Per Protocol CoPP Policy Using the APIC GUI

Procedure

Step 1 On the menu bar, click **Fabric** > **External Access Policies**.

Step 2 In the **Navigation** pane, expand **Policies** > **Interface** > **CoPP Interface**, right click **Create Per Interface Per Protocol CoPP Policy** dialog box to perform the following actions in the **Create Per Interface Per Protocol CoPP Policy** dialog box:

a) In the **Name** field, add a policy name.
b) Expand the **CoPP policy Protocol** table, and enter the protocol name, type, rate, and burst information. Click **Update** and **Submit**.

Step 3 In the **Navigation** pane, expand **Interfaces** > **Leaf Interfaces** > **Policy Groups** > **Create Leaf Access Port Policy Group**, right click **Create Leaf Access Port Policy Group** dialog box to perform the following actions in the **Create Leaf Access Port Policy Group** dialog box:

a) In the **Name** field, add a policy name.
b) In the **COPP Leaf Policy** field, select the policy previously created.
c) Click **Submit**.

Step 4 In the **Navigation** pane, expand **Interfaces** > **Leaf Interfaces** > **Profiles** > **Leaf Profiles**, right click **Create Leaf Interface Profile** dialog box to perform the following actions in the **Create Leaf Interface Profile** dialog box:

a) In the **Name** field, add a profile name.
b) Expand the **Interface Selectors** table, add the interface information in the **Name** and **Interface IDs** fields, and select the **Interface Policy Group** previously created.
c) Click **Ok** and **Submit** to complete Per Interface Per Protocol CoPP configuration.

Configuring Per Interface Per Protocol CoPP Policy Using the NX-OS Style CLI

Procedure

Step 1 Define the CoPP class map and policy map:

Example:

```
(config)# policy-map type control-plane-if <name>
        (config-pmap-copp)# protocol bgp bps <value>
        (config-pmap-copp)# protocol ospf bps <value>
```

Step 2 Applying the configuration to an interface on the leaf:

Example:

```
(config)# leaf 101
        (config-leaf)# int eth 1/10
        (config-leaf-if)# service-policy type control-plane-if output<name>
```

Configuring CoPP Per Interface Per Protocol Using REST API

Procedure

Configure a CoPP per interface per protocol:

Example:

```
<polUni>
    <infraInfra>
    <infraNodeP name="default">
        <infraLeafS name="default" type="range">
            <infraNodeBlk name="default" to_="101" from_="101"/>
        </infraLeafS>
        <infraRsAccPortP tDn="uni/infra/accportprof-default"/>
    </infraNodeP>
    <infraAccPortP name="default">
        <infraHPortS name="regularPorts" type="range">
            <infraPortBlk name="blk1" toPort="7" fromPort="1" toCard="1" fromCard="1"/>
                <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-copp"/>
        </infraHPortS>
    </infraAccPortP>

    <infraFuncP>
        <infraAccPortGrp name="copp">
            <infraRsCoppIfPol tnCoppIfPolName="pc"/>
        </infraAccPortGrp>
    </infraFuncP>

    <coppIfPol name = "pc" >
        <coppProtoClassP name = "test" matchProto="lldp,arp" rate="505" burst = "201"/>
        <coppProtoClassP name = "test1" matchProto="bgp" rate="500" burst = "200" />
    </coppIfPol>
```

```
</infraInfra>
</polUni>
```

About CoPP Prefilters

In Cisco Application Centric Infrastructure (ACI), you can use the control plane policing (CoPP) prefilter feature to filter control packets sent to the CPU. A CoPP prefilter is the same as an inftrastructure access control list (iACL).

Before you use this feature, keep in mind the following key points:

1. This feature works leaf switch-wise or spine switch-wise, not per interface, nor per-L3Out.
2. This feature takes effect across VRF instances, meaning that the filters that you define are not specific to a VRF instance. If you enable a CoPP prefilter and you do not specifically allow ICMP traffic in the configuration of the CoPP prefilter, ICMP traffic sent to the bridge domains of any VRF instance of a given leaf switch is dropped.
3. A CoPP prefilter is configured as a permit-list.
4. This feature is activated by entering the first filtering rule. This means that if you do not have any filtering rules configured, everything is allowed. As soon as you enter the first rule, then everything else is dropped except the traffic that you allow in the filtering rules. This means that all the IPv4/IPv6 control plane traffic by default is denied unless you add it to the permit-list.
5. The filter configuration allows you to enter protocols/DIP/SIP/Protocol/L4 port/L4 port range. You can enter the source and destination IP address of the traffic.
6. You must also allow underlay protocols that are not implicitly allowed. For example, you must allow BGP, otherwise the infra BGP sessions to the leaf or spine switch go down. As another example, you must allow OSPF for remote leaf reachability if you enable this feature on the remote leaf switch.
7. Because of point #6, if you configure a CoPP prefilter on leaf or spine switches of a single POD, you must make sure BGP and DHCP traffic is allowed. If the spine switch is also connected to an IPN/ISN, you must consider allowing OSPF.
8. Because of point #6, in Cisco ACI Multi-Pod, Cisco ACI Multi-Site or Cisco Nexus Dashboard Orchestrator, GOLF, or a remote leaf switch, you must add BGP, DHCP, and OSPF to the permit-list for infra connectivity.
9. Enabling the feature does not disconnect the leaf switch from the fabric because Cisco Application Policy Infrastructure Controller (APIC) traffic is automatically allowed. But, be aware that unless you specifically add BGP to the permit-listed, enabling this feature disconnects the infa BGP session to the leaf switch.
10. The following things are automatically allowed: COOP traffic, vPC control plane traffic, protocols such as LACP/LLDP/CDP, ARP, and Neighbour Discovery packets (RS/RA/NS/NA).
11. ICMP, IGMP, and any other protocol must be specifically allowed. If you enable a CoPP prefilter and you want to make sure that servers can ping the bridge domain subnet IP address, you must make sure ICMP is allowed.

12. There is no support for an ICMP sub-type to allow only ICMP replies or requests. Enabling ICMP enables both.

Supported Platforms

This section lists the supported platforms for the CoPP prefilter feature.

Supported leaf switches:

- N9K-C93108TC-EX
- N9K-C93108TC-FX
- N9K-C93108YC-FX
- N9K-C93180LC-EX
- N9K-C93180YC-EX
- N9K-C9348GC-FXP

Supported spine switches:

- N9K-C92300YC
- N9K-C92304QC
- N9K-C9232C
- N9K-C9236C
- N9K-C9272Q
- N9K-C9364C
- N9K-C9508-FM-2
- N9K-C9516-FM-E2

Limitations

- Only Ethernet type IPv4 or IPv6 packets can be matched in the egress TCAM. ARP and ND packets are not matched.
- A total of 128 (wide key) entries can be included in the allowed list. However, some entries are reserved for internal use.

Configuring a CoPP Prefilter, Policy Group, and Profile Using the GUI

Configuring a CoPP Prefilter Using the Cisco APIC GUI

This section explains how to configure a CoPP prefilter at the leaf level and the spine level using the Cisco APIC GUI.

Before you begin

Access to the APIC GUI

Procedure

Step 1 Click **Fabric** > **External Access Policies**.

Step 2 From the **Navigation** pane, click **Policies** > **Switch**.
The **CoPP Pre-Filter for Leaf** and **CoPP Pre-Filter for Spine** nodes appear in the **Navigation** pane.

Step 3 From the **Navigation** pane, choose between the following options:

- **CoPP Pre-Filter for Leaf**–To create a CoPP prefilter for a leaf switch, right-click on **CoPP Pre-Filter for Leaf** and choose **Create Profiles for CoPP Pre-Filter To Be Applied At The Leaf Level**.
- **CoPP Pre-Filter for Spine**–To create a CoPP prefilter for a spine switch, right-click on **CoPP Pre-Filter for Spine** and choose **Create Profiles for CoPP Pre-Filter To Be Applied At The Spine Level**

The respective CoPP prefilter dialog appears.

Step 4 Enter the appropriate values in the dialog fields.

Note
For information about the fields in the dialog, click the help icon to display the Cisco APIC help file.

Step 5 When finished, click **Submit**.

What to do next

Configure a policy group.

Configuring a Leaf Policy Group Using the GUI

This section explains how to create a policy group.

Before you begin

Access to a Cisco APIC GUI.

Procedure

Step 1 Click **Fabric** > **External Access Policies**.

Step 2 From the **Navigation** pane, click **Switches** > **Leaf Switches**.
The **Policy Groups** node appears in the **Navigation** pane.

Step 3 From the **Navigation** pane, **Policy Groups**–To create a leaf policy group, right-click on **Policy Groups** and choose **Create Access Switch Policy Group**.

The respective policy group dialog appears.

Step 4 From the policy group dialog, enter a name in the **Name** field and click the drop-down arrow of the policy type you want to apply. Any configured policies for the chosen policy type will appear in the drop-down list.

Note
For information about the fields in the dialog, click the help icon to display the Cisco APIC help file.

Step 5 When finished, click **Submit**.

What to do next

Configure a profile.

Configuring a Leaf Profile Using the GUI

This section explains how to create a profile.

Before you begin

You should have a configured policy group.

Procedure

Step 1 Click **Fabric** > **External Access Policies**.

Step 2 From the **Navigation** pane, click **Switches** > **Leaf Switches** > **Profiles**.
The **Leaf Profiles** node appears in the **Navigation** pane.

Step 3 From the **Navigation** pane, **Profiles**–To create a profile for a leaf switch, right-click on **Profiles** and choose **Create Leaf Profile**.
The respective profile dialog appears.

Step 4 From the profile dialog, enter a name in the **Name** field and click the **+** to enter the selector information. Click **Update** when finished.

After clicking **Update**, you return to the profile dialog.

Step 5 Click **Next** to enter the interface selector profile information.

Note
For information about the fields in the dialog, click the help icon to display the Cisco APIC help file.

Step 6 When finished, click **Finish**.

Configuring a CoPP Prefilter Using the CLI

Configuring the CoPP Prefilter for a Leaf Switch Using the CLI

This section explains how to configure a CoPP prefilter policy and policy group then associate a switch policy group with a switch profile using the CLI.

Procedure

Step 1 Switch# **configure terminal**

Enters global configuration mode.

Step 2 Switch(config)# **template control-plane-policing-prefilter-leaf** <*name*>

Creates a CoPP prefilter profile for a leaf switch.

Step 3 Switch (config-control-plane-policing-prefilter-leaf)# **permit proto** { **tcp** | **udp** | **eigrp** | **unspecified** | **icmp** | **icmpv6** | **egp** | **igp** | **l2tp** | **ospf** | **pim** }

Permits the specified IP protocol.

Step 4 Switch (config-control-plane-policing-prefilter-leaf)#**exit**

Enters global configuration mode.

Step 5 Switch(config)# **template leaf-policy-group** <*name*>

Creates a CoPP prefilter policy group leaf switches.

Step 6 Switch(config-leaf-policy-group)# **control-plane-policing-prefilter** <*name*>

Associates a leaf policy group with the CoPP prefilter policy.

Step 7 Switch(config-leaf-policy-group)# **exit** <*name*>

Enters global configuration mode.

Step 8 Switch(config)# **leaf-profile** <*name*>

Creates a leaf profile.

Step 9 Switch(config-leaf-profile)# **leaf-group** <*name*>

Associates a leaf group with a leaf profile.

Step 10 Switch(config-leaf-group)# **leaf-policy-group** <*name*>

Associates a leaf policy group with a leaf group.

Configuring the CoPP Prefilter for a Spine Switch Using the CLI

This section explains how to configure a CoPP prefilter policy and policy group then associate a switch policy group with a switch profile using the CLI.

Procedure

Step 1 Switch# **configure terminal**

Enters global configuration mode.

Step 2 Switch(config)# **template control-plane-policing-prefilter-spine** <*name*>

Creates a CoPP prefilter profile for a spine switch.

Step 3 Switch (config-control-plane-policing-prefilter-spine)# **permit proto** { **tcp** | **udp** | **eigrp** | **unspecified** | **icmp** | **icmpv6** | **egp** | **igp** | **l2tp** | **ospf** | **pim** }

Permits the specified IP protocol.

Step 4 Switch (config-control-plane-policing-prefilter-spine)#**exit**

Enters global configuration mode.

Step 5 Switch(config)# **template spine-policy-group** <*name*>

Creates a CoPP prefilter policy group spine switches.

Step 6 Switch(config-spine-policy-group)# **control-plane-policing-prefilter** <*name*>

Associates a spine policy group with the CoPP prefilter policy.

Step 7 Switch(config-spine-policy-group)# **exit** <*name*>

Enters global configuration mode.

Step 8 Switch(config)# **spine-profile** <*name*>

Creates a spine profile.

Step 9 Switch(config-spine-profile)# **spine-group** <*name*>

Associates a spine group with a spine profile.

Step 10 Switch(config-spine-group)# **spine-policy-group** <*name*>

Associates a spine policy group with a spine group.

Configuring a CoPP Prefilter Using the REST API

Configuring a CoPP Prefilter Policy for a Leaf Switch Using the REST API

This section explains how to configure a CoPP prefilter policy for a leaf switch using the REST API.

Procedure

Step 1 Create a switch policy for CoPP Prefilter with entries the allowed list.

```
<iaclLeafProfile descr="" dn="uni/infra/iaclspinep-spine_icmp" name="COPP_PreFilter_BGP_Config
 " ownerKey="" ownerTag="">
<iaclEntry dstAddr="0.0.0.0/0" dstPortFrom="179" dstPortTo="179" ipProto="tcp" name="bgp"
nameAlias="" srcAddr="0.0.0.0/0" srcPortFrom="179" srcPortTo="179"/>
</iaclLeafProfile>
```

Step 2 Create a switch policy group with CoPP prefilter policies.

```
<infraAccNodePGrp descr="" dn="uni/infra/funcprof/accnodepgrp-COPP_PreFilter_BGP_Config "
name="COPP_PreFilter_BGP_Config" nameAlias="" ownerKey="" ownerTag="">
<infraRsIaclLeafProfile tnIaclLeafProfileName="COPP_PreFilter_BGP_Config"/>
</infraAccNodePGrp>
```

Step 3 Associate switch policy group to switch profiles.

```
<infraNodeP descr="" dn="uni/infra/nprof-leafP-103" name="leafP-103" nameAlias="" ownerKey=""
 ownerTag="">
<infraLeafS descr="" name="103_Sel" nameAlias="" ownerKey="" ownerTag="" type="range">
<infraRsAccNodePGrp tDn="uni/infra/funcprof/accnodepgrp-COPP_PreFilter_BGP_Config"/>
<infraNodeBlk descr="" from_="103" name="nblk1" nameAlias="" to_="103"/>
</infraLeafS>
</infraNodeP>
```

Configuring a CoPP Prefilter Policy for a Spine Using the REST API

This section explains how to configure a CoPP prefilter policy for a spine switch using the REST API.

Procedure

Step 1 Create a switch policy for CoPP Prefilter with entries the allowed list.

```
<iaclSpineProfile descr="" dn="uni/infra/iaclspinep-spine_icmp"
name="COPP_PreFilter_OSPF_Config" ownerKey="" ownerTag="">
<iaclEntry dstAddr="0.0.0.0/0" dstPortFrom="unspecified" dstPortTo="unspecified"
ipProto="ospfigp" name="" nameAlias="" srcAddr="0.0.0.0/0" srcPortFrom="unspecified"
srcPortTo="unspecified"/>
</iaclSpineProfile>
```

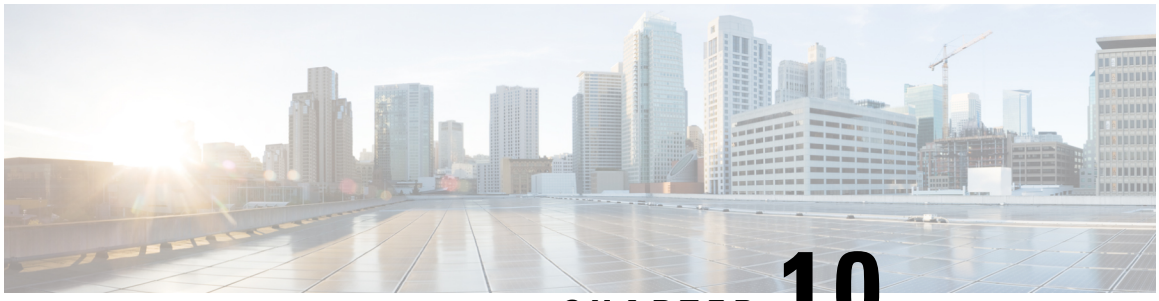
Step 2 Create a switch policy group with CoPP prefilter policies.

```
<infraSpineAccNodePGrp descr=""
dn="uni/infra/funcprof/spaccnodepgrp-COPP_PreFilter_OSPF_Config"
name="COPP_PreFilter_OSPF_Config" nameAlias="" ownerKey="" ownerTag="">
<infraRsIaclSpineProfile tnIaclSpineProfileName="COPP_PreFilter_OSPF_Config"/>
</infraSpineAccNodePGrp>
```

Step 3 Associate switch policy group to switch profiles.

```
<infraSpineP descr="" dn="uni/infra/spprof-204" name="204" nameAlias="" ownerKey=""
ownerTag="">
<infraSpineS descr="" name="204" nameAlias="" ownerKey="" ownerTag="" type="range">
<infraRsSpineAccNodePGrp tDn="uni/infra/funcprof/spaccnodepgrp-COPP_PreFilter_OSPF_Config"/>
<infraNodeBlk descr="" from_="204" name="nodeblock1" nameAlias="" to_="204"/>
</infraSpineS>
<infraRsSpAccPortP tDn="uni/infra/spaccportprof-204"/>
</infraSpineP>
```

What to do next



CHAPTER 10

Fabric Security

This chapter contains the following sections:

- About Federal Information Processing Standards (FIPS), on page 115
- Guidelines and Limitations for FIPS, on page 115
- Configuring FIPS for Cisco APIC Using the GUI, on page 116
- Configuring FIPS for Cisco APIC Using NX-OS Style CLI, on page 116
- Configuring FIPS for Cisco APIC Using REST API, on page 117

About Federal Information Processing Standards (FIPS)

The Federal Information Processing Standards (FIPS) Publication 140-2, Security Requirements for Cryptographic Modules, details the U.S. government requirements for cryptographic modules. FIPS 140-2 specifies that a cryptographic module should be a set of hardware, software, firmware, or some combination that implements cryptographic functions or processes, including cryptographic algorithms and, optionally, key generation, and is contained within a defined cryptographic boundary.

FIPS specifies certain cryptographic algorithms as secure, and it also identifies which algorithms should be used if a cryptographic module is to be called FIPS compliant.

Guidelines and Limitations for FIPS

The following guidelines and limitations apply to FIPS:

- When FIPS is enabled, FIPS is applied across the Cisco Application Policy Infrastructure Controller (APIC).
- When FIPS is enabled, you must disable FIPS before you downgrade the Cisco APIC to a release that does not support FIPS.
- Make your passwords a minimum of eight characters in length.
- Disable Telnet. Log in using only SSH.
- Delete all SSH Server RSA1 keypairs.
- Secure Shell (SSH) and SNMP are supported.

- Disable SNMP v1 and v2. Any existing user accounts on the switch that have been configured for SNMPv3 should be configured only with SHA for authentication and AES for privacy.
- Disable remote authentication through RADIUS/TACACS+. Only local and LDAP users can be authenticated.
- After enabling FIPS on the Cisco APIC, reload the dual supervisor spine switches twice for FIPS to take effect.
- On a dual supervisor spine switch that has FIPS enabled, if a supervisor is replaced, then the spine switch must be reloaded twice for FIPS to take effect on the new supervisor.
- Starting with the 2.3(1) release, FIPS can be configured at the switch level.
- Starting with the 3.1(1) release, when FIPs is enabled, NTP will operate in FIPS mode, Under FIPS mode NTP supports authentication with HMAC-SHA1 and no authentication.

Configuring FIPS for Cisco APIC Using the GUI

When FIPS is enabled, it is applied across Cisco APIC.

Procedure

Step 1 On the menu bar, choose **Admin** > **AAA**.

Step 2 In the **Navigation** pane, expand **AAA** > **Fabric Security**.

Step 3 In the **Work** pane, in the **Properties** area, choose the desired FIPS mode.

The options for FIPS mode are **Disable** and **Enable**. The default value is Disable.

Note
You must reboot to complete the configuration. Anytime you change the mode, you must reboot to complete the configuration.

Configuring FIPS for Cisco APIC Using NX-OS Style CLI

When FIPS is enabled, it is applied across Cisco APIC.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `apic1# configure`	Enters configuration mode.

	Command or Action	Purpose
Step 2	**fips mode enable** **Example:** `apic1(config)# fips mode enable`	Enables FIP. The **no fips mode enable** command disables FIPS. You must reboot to complete the configuration. Anytime you change the mode, you must reboot to complete the configuration.

Configuring FIPS for Cisco APIC Using REST API

When FIPS is enabled, it is applied across Cisco APIC.

Procedure

Configure FIPS for all tenants.

Example:

```
https://apic1.cisco.com/api/node/mo/uni/userext.xml
<aaaFabricSec fipsMode="enable" />
```

Note
You must reboot to complete the configuration. Anytime you change the mode, you must reboot to complete the configuration.

CHAPTER 11

Security Policies

This chapter contains the following sections:

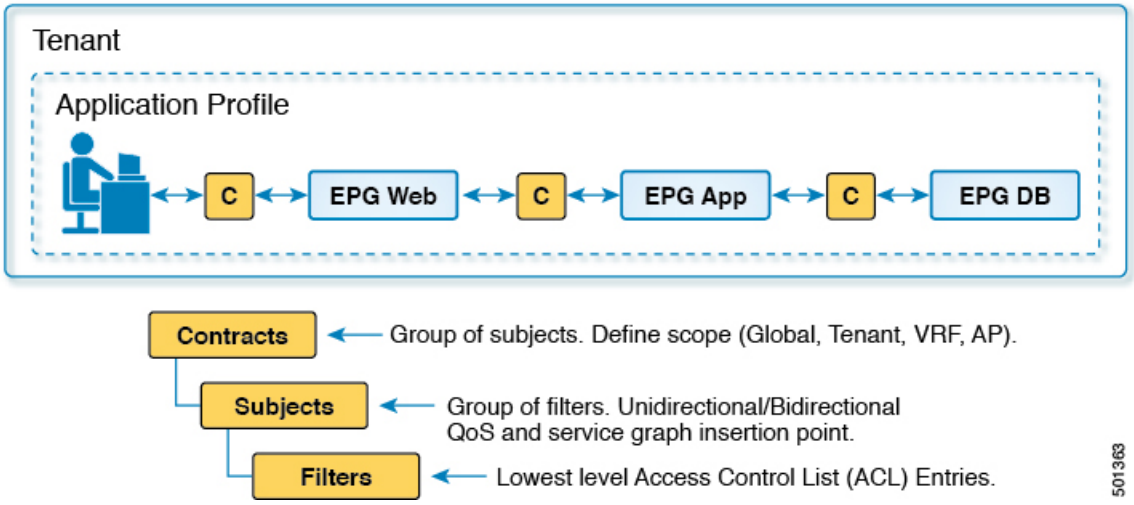
- ACI Fabric Network Access Security Policy Model (Contracts), on page 119
- Enabling and Viewing ACL Contract and Deny Logs, on page 124

ACI Fabric Network Access Security Policy Model (Contracts)

The ACI fabric security policy model is based on contracts. This approach addresses limitations of traditional access control lists (ACLs). Contracts contain the specifications for security policies that are enforced on traffic between endpoint groups.

The following figure shows the components of a contract.

Figure 5: Contract Components



EPG communications require a contract; EPG to EPG communication is not allowed without a contract. The APIC renders the entire policy model, including contracts and their associated EPGs, into the concrete model in each switch. Upon ingress, every packet entering the fabric is marked with the required policy details. Because contracts are required to select what types of traffic can pass between EPGs, contracts enforce security policies. While contracts satisfy the security requirements handled by access control lists (ACLs) in conventional network settings, they are a more flexible, manageable, and comprehensive security policy solution.

Access Control List Limitations

Traditional access control lists (ACLs) have a number of limitations that the ACI fabric security model addresses. The traditional ACL is very tightly coupled with the network topology. They are typically configured per router or switch ingress and egress interface and are customized to that interface and the traffic that is expected to flow through those interfaces. Due to this customization, they often cannot be reused across interfaces, much less across routers or switches.

Traditional ACLs can be very complicated and cryptic because they contain lists of specific IP addresses, subnets, and protocols that are allowed as well as many that are specifically not allowed. This complexity means that they are difficult to maintain and often simply just grow as administrators are reluctant to remove any ACL rules for fear of creating a problem. Their complexity means that they are generally only deployed at specific demarcation points in the network such as the demarcation between the WAN and the enterprise or the WAN and the data center. In this case, the security benefits of ACLs are not exploited inside the enterprise or for traffic that is contained within the data center.

Another issue is the possible huge increase in the number of entries in a single ACL. Users often want to create an ACL that allows a set of sources to communicate with a set of destinations by using a set of protocols. In the worst case, if *N* sources are talking to *M* destinations using *K* protocols, there might be *N*M*K* lines in the ACL. The ACL must list each source that communicates with each destination for each protocol. It does not take many devices or protocols before the ACL gets very large.

The ACI fabric security model addresses these ACL issues. The ACI fabric security model directly expresses the intent of the administrator. Administrators use contract, filter, and label managed objects to specify how groups of endpoints are allowed to communicate. These managed objects are not tied to the topology of the network because they are not applied to a specific interface. They are simply rules that the network must enforce irrespective of where these groups of endpoints are connected. This topology independence means that these managed objects can easily be deployed and reused throughout the data center not just as specific demarcation points.

The ACI fabric security model uses the endpoint grouping construct directly so the idea of allowing groups of servers to communicate with one another is simple. A single rule can allow an arbitrary number of sources to communicate with an equally arbitrary number of destinations. This simplification dramatically improves their scale and maintainability which also means they are easier to use throughout the data center.

Contracts Contain Security Policy Specifications

In the ACI security model, contracts contain the policies that govern the communication between EPGs. The contract specifies what can be communicated and the EPGs specify the source and destination of the communications. Contracts link EPGs, as shown below.

EPG 1 ---------- CONTRACT ---------- EPG 2

Endpoints in EPG 1 can communicate with endpoints in EPG 2 and vice versa if the contract allows it. This policy construct is very flexible. There can be many contracts between EPG 1 and EPG 2, there can be more than two EPGs that use a contract, and contracts can be reused across multiple sets of EPGs, and more.

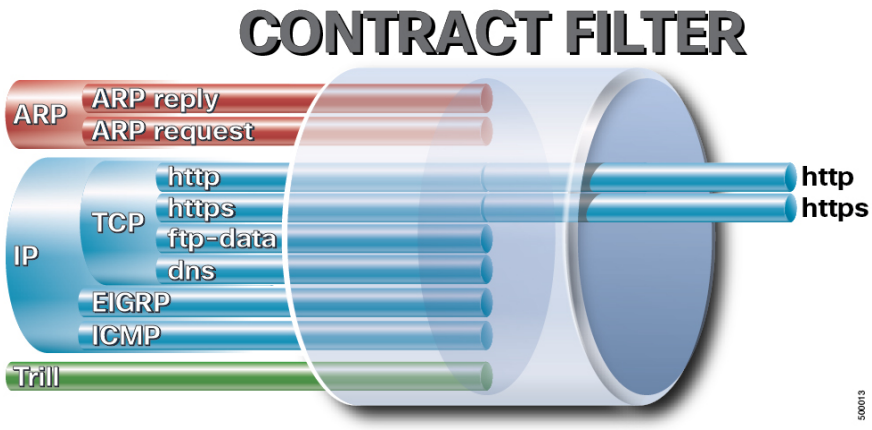
There is also directionality in the relationship between EPGs and contracts. EPGs can either provide or consume a contract. An EPG that provides a contract is typically a set of endpoints that provide a service to a set of client devices. The protocols used by that service are defined in the contract. An EPG that consumes a contract is typically a set of endpoints that are clients of that service. When the client endpoint (consumer) tries to connect to a server endpoint (provider), the contract checks to see if that connection is allowed. Unless otherwise specified, that contract would not allow a server to initiate a connection to a client. However, another contract between the EPGs could easily allow a connection in that direction.

This providing/consuming relationship is typically shown graphically with arrows between the EPGs and the contract. Note the direction of the arrows shown below.

EPG 1 <-------consumes-------- CONTRACT <-------provides-------- EPG 2

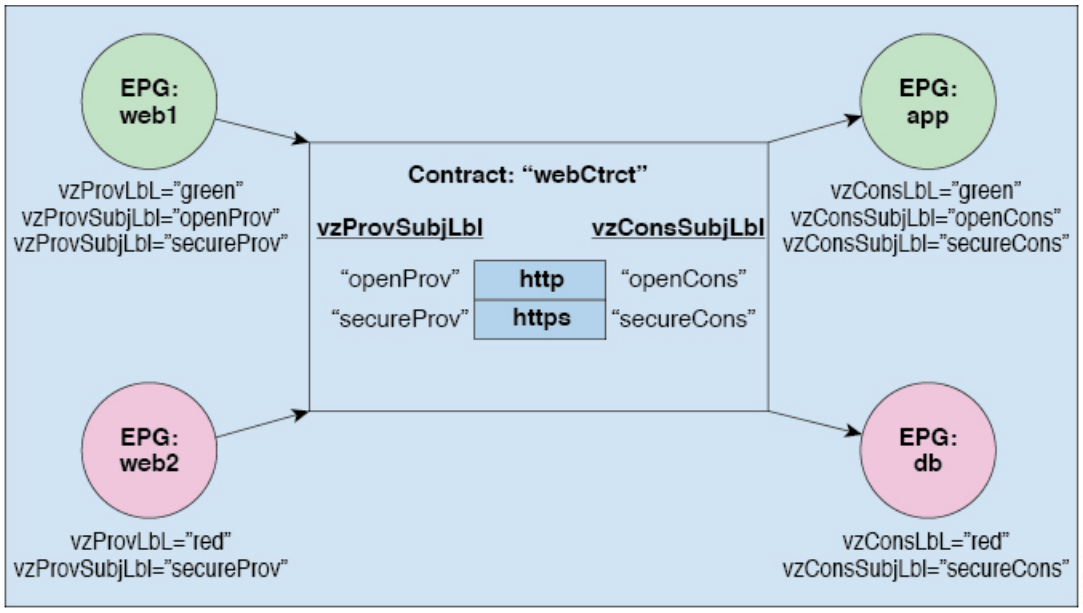
The contract is constructed in a hierarchical manner. It consists of one or more subjects, each subject contains one or more filters, and each filter can define one or more protocols.

Figure 6: Contract Filters



The following figure shows how contracts govern EPG communications.

Figure 7: Contracts Determine EPG to EPG Communications



For example, you may define a filter called HTTP that specifies TCP port 80 and port 8080 and another filter called HTTPS that specifies TCP port 443. You might then create a contract called webCtrct that has two sets of subjects. openProv and openCons are the subjects that contain the HTTP filter. secureProv and secureCons are the subjects that contain the HTTPS filter. This webCtrct contract can be used to allow both secure and non-secure web traffic between EPGs that provide the web service and EPGs that contain endpoints that want to consume that service.

These same constructs also apply for policies that govern virtual machine hypervisors. When an EPG is placed in a virtual machine manager (VMM) domain, the APIC downloads all of the policies that are associated with the EPG to the leaf switches with interfaces connecting to the VMM domain. For a full explanation of VMM domains, see the *Virtual Machine Manager Domains* chapter of *Application Centric Infrastructure Fundamentals*. When this policy is created, the APIC pushes it (pre-populates it) to a VMM domain that specifies which switches allow connectivity for the endpoints in the EPGs. The VMM domain defines the set of switches and ports that allow endpoints in an EPG to connect to. When an endpoint comes on-line, it is associated with the appropriate EPGs. When it sends a packet, the source EPG and destination EPG are derived from the packet and the policy defined by the corresponding contract is checked to see if the packet is allowed. If yes, the packet is forwarded. If no, the packet is dropped.

Contracts consist of 1 or more subjects. Each subject contains 1 or more filters. Each filter contains 1 or more entries. Each entry is equivalent to a line in an Access Control List (ACL) that is applied on the Leaf switch to which the endpoint within the endpoint group is attached.

In detail, contracts are comprised of the following items:

- Name—All contracts that are consumed by a tenant must have different names (including contracts created under the common tenant or the tenant itself).
- Subjects—A group of filters for a specific application or service.
- Filters—Used to classify traffic based upon layer 2 to layer 4 attributes (such as Ethernet type, protocol type, TCP flags and ports).
- Actions—Action to be taken on the filtered traffic. The following actions are supported:
  - Permit the traffic (regular contracts, only)
  - Mark the traffic (DSCP/CoS) (regular contracts, only)
  - Redirect the traffic (regular contracts, only, through a service graph)
  - Copy the traffic (regular contracts, only, through a service graph or SPAN)
  - Block the traffic (taboo contracts)

    With Cisco APIC Release 3.2(x) and switches with names that end in EX or FX, you can alternatively use a subject Deny action or Contract or Subject Exception in a standard contract to block traffic with specified patterns.
  - Log the traffic (taboo contracts and regular contracts)
- Aliases—(Optional) A changeable name for an object. Although the name of an object, once created, cannot be changed, the Alias is a property that can be changed.

Thus, the contract allows more complex actions than just allow or deny. The contract can specify that traffic that matches a given subject can be re-directed to a service, can be copied, or can have its QoS level modified. With pre-population of the access policy in the concrete model, endpoints can move, new ones can come on-line, and communication can occur even if the APIC is off-line or otherwise inaccessible. The APIC is removed from being a single point of failure for the network. Upon packet ingress to the ACI fabric, security policies are enforced by the concrete model running in the switch.

Security Policy Enforcement

As traffic enters the leaf switch from the front panel interfaces, the packets are marked with the EPG of the source EPG. The leaf switch then performs a forwarding lookup on the packet destination IP address within the tenant space. A hit can result in any of the following scenarios:

1. A unicast (/32) hit provides the EPG of the destination endpoint and either the local interface or the remote leaf switch VTEP IP address where the destination endpoint is present.
2. A unicast hit of a subnet prefix (not /32) provides the EPG of the destination subnet prefix and either the local interface or the remote leaf switch VTEP IP address where the destination subnet prefix is present.
3. A multicast hit provides the local interfaces of local receivers and the outer destination IP address to use in the VXLAN encapsulation across the fabric and the EPG of the multicast group.

Note Multicast and external router subnets always result in a hit on the ingress leaf switch. Security policy enforcement occurs as soon as the destination EPG is known by the ingress leaf switch.

A miss result in the forwarding table causes the packet to be sent to the forwarding proxy in the spine switch. The forwarding proxy then performs a forwarding table lookup. If it is a miss, the packet is dropped. If it is a hit, the packet is sent to the egress leaf switch that contains the destination endpoint. Because the egress leaf switch knows the EPG of the destination, it performs the security policy enforcement. The egress leaf switch must also know the EPG of the packet source. The fabric header enables this process because it carries the EPG from the ingress leaf switch to the egress leaf switch. The spine switch preserves the original EPG in the packet when it performs the forwarding proxy function.

On the egress leaf switch, the source IP address, source VTEP, and source EPG information are stored in the local forwarding table through learning. Because most flows are bidirectional, a return packet populates the forwarding table on both sides of the flow, which enables the traffic to be ingress filtered in both directions.

Multicast and EPG Security

Multicast traffic introduces an interesting problem. With unicast traffic, the destination EPG is clearly known from examining the packet's destination. However, with multicast traffic, the destination is an abstract entity: the multicast group. Because the source of a packet is never a multicast address, the source EPG is determined in the same manner as in the previous unicast examples. The derivation of the destination group is where multicast differs.

Because multicast groups are somewhat independent of the network topology, static configuration of the (S, G) and (*, G) to group binding is acceptable. When the multicast group is placed in the forwarding table, the EPG that corresponds to the multicast group is also put in the forwarding table.

Note This document refers to multicast stream as a multicast group.

The leaf switch always views the group that corresponds to the multicast stream as the destination EPG and never the source EPG. In the access control matrix shown previously, the row contents are invalid where the multicast EPG is the source. The traffic is sent to the multicast stream from either the source of the multicast stream or the destination that wants to join the multicast stream. Because the multicast stream must be in the

forwarding table and there is no hierarchical addressing within the stream, multicast traffic is access controlled at the ingress fabric edge. As a result, IPv4 multicast is always enforced as ingress filtering.

The receiver of the multicast stream must first join the multicast stream before it receives traffic. When sending the IGMP Join request, the multicast receiver is actually the source of the IGMP packet. The destination is defined as the multicast group and the destination EPG is retrieved from the forwarding table. At the ingress point where the router receives the IGMP Join request, access control is applied. If the Join request is denied, the receiver does not receive any traffic from that particular multicast stream.

The policy enforcement for multicast EPGs occurs on the ingress by the leaf switch according to contract rules as described earlier. Also, the multicast group to EPG binding is pushed by the APIC to all leaf switches that contain the particular tenant (VRF).

Taboos

While the normal processes for ensuring security still apply, the ACI policy model aids in assuring the integrity of whatever security practices are employed. In the ACI policy model approach, all communications must conform to these conditions:

- Communication is allowed only based on contracts, which are managed objects in the model. If there is no contract, inter-EPG communication is disabled by default.
- No direct access to the hardware; all interaction is managed through the policy model.

Taboo contracts can be used to deny specific traffic that is otherwise allowed by contracts. The traffic to be dropped matches a pattern (such as, any EPG, a specific EPG, or traffic matching a filter). Taboo rules are unidirectional, denying any matching traffic coming toward an EPG that provides the contract.

With Cisco APIC Release 3.2(x) and switches with names that end in EX or FX, you can alternatively use a subject Deny action or Contract or Subject Exception in a standard contract to block traffic with specified patterns.

Enabling and Viewing ACL Contract and Deny Logs

About ACL Contract Permit and Deny Logs

To log and/or monitor the traffic flow for a contract rule, you can enable and view the logging of packets or flows that were allowed to be sent because of contract permit rules or the logging of packets or flows that were dropped because of:

- Taboo contract deny rules
- Deny actions in contract subjects
- Contract or subject exceptions
- ACL contract permit in the ACI fabric is only supported on Nexus 9000 Series switches with names that end in EX or FX, and all later models. For example, N9K-C93180LC-EX or N9K-C9336C-FX.
- Deny logging in the ACI fabric is supported on all platforms.

- Using log directive on filters in management contracts is not supported. Setting the log directive will cause zoning-rule deployment failure.

For information on standard and taboo contracts and subjects, see *Cisco Application Centric Infrastructure Fundamentals* and *Cisco APIC Basic Configuration Guide*.

EPG Data Included in ACL Permit and Deny Log Output

Up to Cisco APIC, Release 3.2(1), the ACL permit and deny logs did not identify the EPGs associated with the contracts being logged. In release 3.2(1) the source EPG and destination EPG are added to the output of ACI permit and deny logs. ACL permit and deny logs include the relevant EPGs with the following limitations:

- Depending on the position of the EPG in the network, EPG data may not be available for the logs.
- When configuration changes occur, log data may be out of date. In steady state, log data is accurate.

The most accurate EPG data in the permit and deny logs results when the logs are focussed on:

- Flows from EPG to EPG, where the ingress policy is installed at the ingress TOR and the egress policy is installed at the egress TOR.
- Flows from EPG to L3Out, where one policy is applied on the border leaf TOR and the other policy is applied on a non-BL TOR.

EPGs in the log output are not supported for uSeg EPGs or for EPGs used in shared services (including shared L3Outs).

Enabling ACL Contract Permit and Deny Logging Using the GUI

The following steps show how to enable contract permit and deny logging using the GUI:

Note The tenant that contains the permit logging is the tenant that contains the VRF that the EPG is associated to. This will not necessarily be the same tenant as the EPG or its associated contracts.

Procedure

Step 1 On the menu bar, choose **Tenants** > **<tenant name>**.

Step 2 In the **Navigation** pane, expand **Contracts**, right-click **Standard**, and choose **Create Contract**.

Step 3 In the **Create Contract** dialog box, perform the following actions:

a) In the **Name** field, type the name for the contract.
b) In the **Scope** field, choose the scope for it (VRF, Tenant, or Global).
c) Optional. Set the target DSCP or QoS class to be applied to the contract.
d) Click the + icon to expand **Subjects**.

Step 4 In the Create Contract Subject dialog box, perform the following actions:

Step 5 Enter the name of the subject and an optional description.

Step 6 Optional. From the drop-down list for the target DSCP, select the DSCP to be applied to the subject.

Step 7 Leave **Apply Both Directions** checked, unless you want the contract to only be applied from the consumer to the provider, instead of in both directions.

Step 8 Leave **Reverse Filter Ports** checked if you unchecked **Apply Both Directions** to swap the Layer 4 source and destination ports so that the rule is applied from the provider to the consumer.

Step 9 Click the + icon to expand **Filters**.

Step 10 In the **Name** drop-down list, choose an option; for example, click **arp**, **default**, **est**, or **icmp**, or choose a previously configured filter.

Step 11 In the **Directives** drop-down list, click **log**.

Step 12 (Optional) Change the Action to be taken with this subject to **Deny** (or leave the action to the default, **Permit**.

With Directive: log enabled, if the action for this subject is **Permit**, ACL permit logs track the flows and packets that are controlled by the subject and contract. If the action for this subject is **Deny**, ACL deny logs track the flows and packets.

Step 13 (Optional) Set the priority for the subject.

Step 14 Click **Update**.

Step 15 Click **OK**.

Step 16 Click **Submit**.
Logging is enabled for this contract.

Enabling ACL Contract Permit Logging Using the NX-OS CLI

The following example shows how to enable Contract permit logging using the NX-OS CLI.

Procedure

Step 1 To enable logging of packets or flows that were allowed to be sent because of Contract permit rules, use the following commands:

```
configure
tenant <tenantName>
contract <contractName> type <permit>
subject <subject Name>
access-group <access-list> <in/out/both> log
```

Example:

For example:

```
apic1# configure
apic1(config)# tenant BDMode1
apic1(config-tenant)# contract Logicmp type permit
apic1(config-tenant-contract)# subject icmp
apic1(config-tenant-contract-subj)# access-group arp both log
```

Step 2 To disable the permit logging use the **no** form of the access-group command; for example, use the `no access-group arp both log` command.

Enabling ACL Contract Permit Logging Using the REST API

The following example shows you how to enable permit and deny logging using the REST API. This example configures ACL permit and deny logging for a contract with subjects that have Permit and Deny actions configured.

Procedure

For this configuration, send a post with XML similar to the following example:

Example:

```
<vzBrCP dn="uni/tn-Tenant64/brc-C64" name="C64" scope="context">
    <vzSubj consMatchT="AtleastOne" name="HTTPSsbj" provMatchT="AtleastOne" revFltPorts="yes"
 rn="subj-HTTPSsbj">
         <vzRsSubjFiltAtt action="permit" directives="log" forceResolve="yes"
priorityOverride="default"
rn="rssubjFiltAtt-PerHTTPS" tDn="uni/tn-Tenant64/flt-PerHTTPS" tRn="flt-PerHTTPS"
tnVzFilterName="PerHTTPS"/>
    </vzSubj>
    <vzSubj consMatchT="AtleastOne" name="httpSbj" provMatchT="AtleastOne" revFltPorts="yes"
 rn="subj-httpSbj">
        <vzRsSubjFiltAtt action="deny" directives="log" forceResolve="yes"
priorityOverride="default"
rn="rssubjFiltAtt-httpFilter" tDn="uni/tn-Tenant64/flt-httpFilter" tRn="flt-httpFilter"
tnVzFilterName="httpFilter"/>
    </vzSubj>
    <vzSubj consMatchT="AtleastOne" name="subj64" provMatchT="AtleastOne" revFltPorts="yes"
 rn="subj-subj64">
        <vzRsSubjFiltAtt action="permit" directives="log" forceResolve="yes"
priorityOverride="default"
rn="rssubjFiltAtt-icmp" tDn="uni/tn-common/flt-icmp" tRn="flt-icmp" tnVzFilterName="icmp"/>

    </vzSubj>
</vzBrCP>
```

Enabling Taboo Contract Deny Logging Using the GUI

The following steps show how to enable Taboo Contract deny logging using the GUI.

Procedure

Step 1 On the menu bar, choose **Tenants** > **<tenant name>**.

Step 2 In the **Navigation** pane, expand **Contracts**.

Step 3 Right-click **Taboos** and choose **Create Taboo Contract**.

Step 4 In the Create Taboo Contract dialog box, perform the following actions to specify the Taboo contract:

a) In the **Name** field, type the name for the contract.

b) Optional. In the **Description** field, type a description of the Taboo contract.

c) Click the + icon to expand **Subjects**.

Step 5 In the **Create Taboo Contract Subject** dialog box, perform the following actions:

a) In the Specify Identity of Subject area, type a name and optional description.
b) Click the + icon to expand **Filters**.
c) From the **Name** drop-down list, choose one of the default values, such as**<tenant_name>/arp**, **<tenant_name>/default**, **<tenant_name>/est**, **<tenant_name>/icmp**, choose a previously created filter, or**Create Filter**.

Note
If you chose **Create Filter**, in the Specify Filter Identity Area, perform the following actions to specify criteria for the ACL Deny rule:

a. Type a name and optional description.

b. Expand **Entries**, type a name for the rule, and choose the criteria to define the traffic you want to deny.

c. In theDirectives drop-down list, choose **log**.

d. Click **Update**.

e. Click **OK**.

Step 6 Click **Submit**.
Logging is enabled for this Taboo contract.

Enabling Taboo Contract Deny Logging Using the NX-OS CLI

The following example shows how to enable Taboo Contract deny logging using the NX-OS CLI.

Procedure

Step 1 To enable logging of packets or flows dropped because of Taboo Contract deny rules, use the following commands:

```
configure
tenant <tenantName>
contract <contractName> type <deny>
subject <subject Name>
access-group <access-list> <both> log
```

Example:

For example:

```
apic1# configure
apic1(config)# tenant BDMode1
apic1(config-tenant)# contract dropFTP type deny
apic1(config-tenant-contract)# subject dropftp
apic1(config-tenant-contract-subj)# access-group ftp both log
```

Step 2 To disable the deny logging use the **no** form of the access-group command; for example, use the `no access-group https both log` command.

Enabling Taboo Contract Deny Logging Using the REST API

The following example shows you how to enable Taboo Contract deny logging using the REST API.

Procedure

To configure taboo contract deny logging, send a post with XML similar to the following example.

Example:

```
<vzTaboo dn="uni/tn-Tenant64/taboo-TCtrctPrefix" name="TCtrctPrefix" scope="context">
    <vzTSubj name="PrefSubj" rn="tsubj-PrefSubj"">
        <vzRsDenyRule directives="log" forceResolve="yes" rn="rsdenyRule-default"
tCl="vzFilter"
tDn="uni/tn-common/flt-default" tRn="flt-default"/>
    </vzTSubj>
</vzTaboo>
```

Viewing ACL Permit and Deny Logs Using the GUI

The following steps show how to view ACL permit and deny logs (if they are enabled) for traffic flows, using the GUI:

Procedure

Step 1 On the menu bar, choose **Tenants** > **<tenant name>**.

Step 2 In the **Navigation** pane, click on **Tenant <tenant name>**.

Step 3 In the **Tenants <tenant name> Work** pane, click the **Operational** tab.

Step 4 Under the **Operational** tab, click the **Flows** tab.
Under the **Flows** tab, click one of the tabs to view log data for Layer 2 permit logs (**L2 Permit**) Layer 3 permit logs (**L3 Permit**, Layer 2 deny logs (**L2 Drop**), or Layer 3 deny logs (**L3 Drop**). On each tab, you can view ACL logging data, if traffic is flowing. The data points differ according to the log type and ACL rule; for example, the following data points are included for **L3 Permit** and **L3 Deny** logs:

- VRF
- Alias
- Source IP address
- Destination IP address
- Protocol
- Source port
- Destination port
- Source MAC address

- Destination MAC address
- Node
- Source interface
- VRF Encap
- Source EPG
- Destination EPG
- Source PC Tag
- Destination PC Tag

Note
You can also use the **Packets** tab (next to the **Flows** tab) to access ACL logs for groups of packets (up to 10) with the same signature, source and destination. You can see what type of packets are being sent and which are being dropped.

Viewing ACL Permit and Deny Logs Using the REST API

The following example shows how to view Layer 2 deny log data for traffic flows, using the REST API. You can send queries using the following MOs:

- acllogDropL2Flow
- acllogPermitL2Flow
- acllogDropL3Flow
- acllogPermitL3Flow
- acllogDropL2Pkt
- acllogPermitL2Pkt
- acllogDropL3Pkt
- acllogPermitL3Pkt

Before you begin

You must enable permit or deny logging, before you can view ACL contract permit and deny log data.

Procedure

To view Layer 3 drop log data, send the following query using the REST API:

```
GET https://apic-ip-address/api/class/acllogDropL3Flow
```

Example:

The following example shows sample output:

```
<?xml version="1.0" encoding="UTF-8"?>
<imdata totalCount="2">
    <acllogPermitL3Flow childAction=""
dn="topology/pod-1/node-101/ndbgs/acllog/tn-common/ctx-inb

/permitl3flow-spctag-333-dpctag-444-sepgname-unknown-depgname-unknown-sip-[100:c000:a00:700:b00:0:f00:0]


-dip-[19.0.2.10]-proto-udp-sport-17459-dport-8721-smac-00:00:15:00:00:28-dmac-00:00:12:00:00:25-sintf-

        [port-channel5]-vrfencap-VXLAN: 2097153" dstEpgName="unknown" dstIp="19.0.2.10"
dstMacAddr="00:00:12:00:00:25"
       dstPcTag="444" dstPort="8721" lcOwn="local" modTs="never" monPolDn="" protocol="udp"
 srcEpgName="unknown"
        srcIntf="port-channel5" srcIp="100:c000:a00:700:b00:0:f00:0"
srcMacAddr="00:00:15:00:00:28" srcPcTag="333"
        srcPort="17459" status="" vrfEncap="VXLAN: 2097153"/>
    <acllogPermitL3Flow childAction=""
dn="topology/pod-1/node-102/ndbgs/acllog/tn-common/ctx-inb

/permitl3flow-spctag-333-dpctag-444-sepgname-unknown-depgname-unknown-sip-[100:c000:a00:700:b00:0:f00:0]-dip-


[19.0.2.10]-proto-udp-sport-17459-dport-8721-smac-00:00:15:00:00:28-dmac-00:00:12:00:00:25-sintf-

        [port-channel5]-vrfencap-VXLAN: 2097153" dstEpgName="unknown" dstIp="19.0.2.10"
dstMacAddr="00:00:12:00:00:25"
       dstPcTag="444" dstPort="8721" lcOwn="local" modTs="never" monPolDn="" protocol="udp"
 srcEpgName="unknown"
        srcIntf="port-channel5" srcIp="100:c000:a00:700:b00:0:f00:0"
srcMacAddr="00:00:15:00:00:28" srcPcTag="333"
        srcPort="17459" status="" vrfEncap="VXLAN: 2097153"/>
</imdata>
```

Viewing ACL Permit and Deny Logs Using the NX-OS CLI

The following steps show how to view ACL log details using the NX-OS-style CLI **show acllog** command.

The syntax for the Layer 3 command is **show acllog {permit | deny} l3 {pkt | flow} tenant <tenant_name> vrf <vrf_name> srcip <source_ip> dstip <destination_ip> srcport <source_port> dstport <destination_port> protocol <protocol> srcintf <source_interface> start-time <startTime> end-time <endTime> detail**

The syntax for the Layer 2 command is **show acllog {permit | deny} l2 {flow | pkt} tenant <tenant_name> vrf <VRF_name> srcintf <source_interface> vlan <VLAN_number> detail**

Note The full syntax of the **show acllog** command is only available on Generation 2 Cisco Nexus 9000 series switches (with names that end in EX or FX or later, such as N9K-C93180LC-EX) and Cisco APIC Release 3.2 or later. With Generation 1 switches (with names that do not end in EX or FX) or Cisco APIC releases before 3.2, the available syntax is as above.

In Cisco APIC 3.2 and later, additional keywords are added to both versions of the command, with the **detail** keyword:**[dstEpgName <destination_EPG_name>| dstmac <destination_MAC_address> | dstpctag <destination_PCTag> | srcEpgName <source_EPG_name> | srcmac <source_MAC_address> | srcpctag <source_PCTag>]**

Procedure

Step 1 The following example shows how to use the **show acllog drop l3 flow tenant common vrf default detail** command to display detailed information about Layer 3 deny logs for the common tenant:

Example:

```
apic1# show acllog deny l3 flow tenant common vrf default detail
SrcPcTag  : 49153
DstPcTag  : 32773
SrcEPG    : uni/tn-TSW_Tenant0/ap-tsw0AP0/epg-tsw0ctx0BD0epg6
DstEPG    : uni/tn-TSW_Tenant0/ap-tsw0AP0/epg-tsw0ctx0BD0epg5
SrcIp     : 16.0.2.10
DstIp     : 19.0.2.10
Protocol  : udp
SrcPort   : 17459
DstPort   : 8721
SrcMAC    : 00:00:15:00:00:28
DstMAC    : 00:00:12:00:00:25
Node      : 101
SrcIntf   : port-channel5
VrfEncap  : VXLAN: 2097153
```

This example shows the output on Generation 2 switches, with Cisco APIC Release 3.2 or later.

Step 2 The following example shows how to use the **show acllog deny l2 flow tenant common vrf tsw0connctx0 detail** command to display detailed information about Layer 3 deny logs for the common tenant:

Example:

```
apic1# show acllog deny l2 flow tenant common vrf tsw0connctx0 detail
SrcPcTag DstPcTag   SrcEPG           DstEPG          SrcMAC            DstMAC         Node
   SrcIntf   vlan
--------- --------- ----------------  --------------  ------------------  ------------------
   -----   ---------  ------
 32773    49153   uni/tn-TSW       uni/tn-TSW  00:00:11:00:00:11  11:00:32:00:00:33   101
    port-     2
                   _Tenant0/ap-     _Tenant0/ap-
    channel8
                   tsw0AP0/epg-     tsw0AP0/epg-
                   tsw0ctx0BD0epg5  tsw0ctx0BD0epg6
```

This example shows the output on Generation 2 switches, with Cisco APIC Release 3.2 or later.

Step 3 The following example shows how to use the **show acllog permit l3 pkt tenant <tenant name> vrf <vrf name> [detail]** command to display detailed information about the common VRF ACL Layer 3 permit packets that were sent:

```
apic1# show acllog permit l3 pkt tenant common vrf default detail acllog permit l3 packets
 detail:
srcIp      : 10.2.0.19
dstIp      : 10.2.0.16
protocol   : udp
srcPort    : 13124
dstPort    : 4386
```

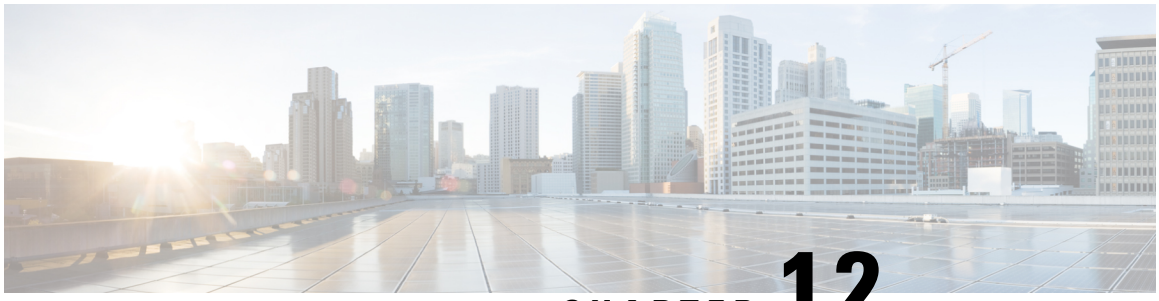
```
srcIntf    : port-channel5
vrfEncap   : VXLAN: 2097153
pktLen     : 112
srcMacAddr : 00:00:15:00:00:28
dstMacAddr : 00:00:12:00:00:25
timeStamp  : 2015-03-17T21:31:14.383+00:00
```

This example shows the output on Generation 1 switches, or with Cisco APIC releases before 3.2.

Step 4 The following example shows how to use the **show acllog permit l2 pkt tenant <tenant name> vrf <vrf name> srcintf <s interface>** command to view information about default VRF Layer 2 packets sent from interface port-channel15:

```
apic1# show acllog permit l2 pkt tenant common vrf default srcintf port-channel5
acllog permit L2 Packets
      Node          srcIntf       pktLen     timeStamp
 --------------  --------------  --------  --------------
                 port-channel5      1      2015-03-17T21:
                                           31:14.383+00:00
```

This example shows the output on Generation 1 switches, or with Cisco APIC releases before 3.2.



CHAPTER 12

Data Plane Policing

This chapter contains the following sections:

- Overview of Data Plane Policing, on page 135
- Guidelines and Limitations, on page 136
- Configuring Data Plane Policing for Layer 2 Interface Using the GUI, on page 137
- Configuring Data Plane Policing for Layer 3 Interface Using the APIC GUI, on page 139
- Configuring Data Plane Policing Using the REST API, on page 140
- Configuring Data Plane Policing Using NX-OS Style CLI, on page 142
- Data Plane Policing at the Endpoint Group Level, on page 147

Overview of Data Plane Policing

Use data plane policing (DPP) to manage bandwidth consumption on Cisco Application Centric Infrastructure (ACI) fabric access interfaces. DPP policies can apply to egress traffic, ingress traffic, or both. DPP monitors the data rates for a particular interface. When the data rate exceeds user-configured values, marking or dropping of packets occurs immediately. Policing does not buffer the traffic; therefore, the transmission delay is not affected. When traffic exceeds the data rate, the Cisco ACI fabric can either drop the packets or mark QoS fields in them.

Before the 3.2 release, the standard behavior for the policer was to be per-EPG member in the case of DPP policy being applied to the EPG, while the same policer was allocated on the leaf switch for the Layer 2 and Layer 3 case. This distinction was done because the DPP policer for Layer 2/Layer 3 case was assumed to be per-interface already, hence it was assumed different interfaces might get different ones. While the per-EPG DPP policy was introduced, it was clear that on a given leaf switch, several members could be present and therefore the policer it made sense to be per-member in order to avoid unwanted drops.

Starting with release 3.2, a clear semantic is given to the Data Plane Policer policy itself, as well as a new flag introducing the sharing-mode setting as presented in the CLI. Essentially, there is no longer an implicit behavior, which is different if the Data Plane Policer is applied to Layer 2/Layer 3 or to per-EPG case. Now the user has the control of the behavior. If the sharing-mode is set to **shared**, then all the entities on the leaf switch referring to the same Data Plane Policer, will share the same hardware policer. If the sharing-mode is set to **dedicated** then there would be a different HW policer allocated for each Layer 2 or Layer 3 or EPG member on the leaf switch. The policer is then dedicated to the entity that needs to be policed.

DPP policies can be single-rate, dual-rate, and color-aware. Single-rate policies monitor the committed information rate (CIR) of traffic. Dual-rate policers monitor both CIR and peak information rate (PIR) of traffic. In addition, the system monitors associated burst sizes. Three colors, or conditions, are determined by

the policer for each packet depending on the data rate parameters supplied: conform (green), exceed (yellow), or violate (red).

Typically, DPP policies are applied to physical or virtual layer 2 connections for virtual or physical devices such as servers or hypervisors, and on layer 3 connections for routers. DPP policies applied to leaf switch access ports are configured in the fabric access (infra) portion of the Cisco ACI fabric, and must be configured by a fabric administrator. DPP policies applied to interfaces on border leaf switch access ports (l3extOut or l2extOut) are configured in the tenant (fvTenant) portion of the Cisco ACI fabric, and can be configured by a tenant administrator.

The data plane policer can also be applied on an EPG so that traffic that enters the Cisco ACI fabric from a group of endpoints are limited per member access interface of the EPG. This is useful to prevent monopolization of any single EPG where access links are shared by various EPGs.

Only one action can be configured for each condition. For example, a DPP policy can to conform to the data rate of 256000 bits per second, with up to 200 millisecond bursts. The system applies the conform action to traffic that falls within this rate, and it would apply the violate action to traffic that exceeds this rate. Color-aware policies assume that traffic has been previously marked with a color. This information is then used in the actions taken by this type of policer.

For information about traffic storm control, see the *Cisco APIC Layer 2 Networking Configuration Guide*.

Guidelines and Limitations

The following are the guidelines and limitations for configuring data plane policing:

- The data plane does not police the packets transmitted from CPU and CPU bound packets on ACI fabric access interfaces.
- The **Dedicated Policer** sharing mode is not supported for Layer 2 interfaces nor Layer 3 interfaces. This limitation is because the mode is only supported for EPGs.

The following are guidelines and limitations for EPG policing:

- Feature support begins with switch models ending in EX/FX (example: N9K-C93180YC-EX) and subsequent models.
- Egress traffic policing is not supported on the EPG level policer.
- Policer mode packet-per-second is not supported.
- Policer type 2R3C is not supported.
- Policer is not supported when **intra-EPG isolation** is enforced in EPG.
- Statistics and considerations for **tuning** include:
  - Awareness of packets that are dropped/allowed is important to know to mitigate issues or for overuse of resources.
  - Statistics are provided in the GUI using the statistics infrastructure. Statistics are exported through the REST API as for any statistic in the Cisco ACI fabric.
  - Statistics are available on per-EPG member, and are useful if the Data Plane Policer policy is of type **dedicated**, otherwise the statistics reflect the statistics of all the ports using it on the leaf switch.

- In certain cases, such as when frames goes through FCoE supported devices, these get classified into the no drop FCoE class. In FCoE devices, this can cause drop off packets when the packet length is higher than the allowed 2184 bytes.

Configuring Data Plane Policing for Layer 2 Interface Using the GUI

Before you begin

The tenant, VRF, and external routed network where you configure the Data Plane Policing policy must be already created.

To apply the Layer 2 Data Plane Policing policy, the policy must be added to a policy group and the policy group must be mapped to an interface profile.

Procedure

Step 1 On the menu bar, choose **Fabric** > **Access Policies**.

Step 2 In the Navigation pane, choose **Policies** > **Interface** > **Data Plane Policing**.

Step 3 Right-click **Data Plane Policing Policing**, and click **Create a Data Plane Policing Policy**.

Step 4 In the **Create a Data Plane Policing Policy** dialog box, in the **Name** field, enter a name for the policy.

Step 5 For **Administrative State**, choose **enabled**.

Step 6 For **BGP Domain Policer Mode**, choose either **Bit Policer** or **Packet Policer**.

Step 7 For **Type**, choose **1 Rate 2 Color** or **2 Rate 3 Color**.

Switch models ending in EX/FX (for example: N9K-C93180YC-EX) and subsequent models do not support **2 Rate 3 Color**.

Step 8 For **Conform Action**, choose an action.

This choice defines an actions for traffic that conforms with certain conditions.

- **Drop**: Drops the packets if the conditions are met.
- **Mark**: Marks the packets if the conditions are met.
- **Transmit**: Transmits the packets if the conditions are met.

Step 9 If for **Conform Action** you chose **Mark**, perform the following substeps:

a) For **Conform mark CoS**, enter the class of service for packets that conformed with the conditions.

b) For **Conform mark dscp**, enter the differentiated services code point (DSCP) for packets that conformed with the conditions.

Step 10 The administrator can configure the CoS and DSCP values in the **Conform** and **Violate** fields.

Step 11 If for **Type** you chose **2 Rate 3 Color**, then for **Exceed Action**, choose an action.

This choice defines an actions for traffic that exceeds certain conditions.

- **Drop**: Drops the packets if the conditions are met.
- **Mark**: Marks the packets if the conditions are met.
- **Transmit**: Transmits the packets if the conditions are met.

Step 12 If for **Exceed Action** you chose **Mark**, perform the following substeps:

a) For **Exceed mark CoS**, enter the class of service for packets that exceeded the conditions.
b) For **Exceed mark dscp**, enter the differentiated services code point (DSCP) for packets that exceeded the conditions.

Step 13 For **Violate Action**, choose an action.

This choice defines an actions for traffic that violates to certain conditions.

- **Drop**: Drops the packets if the conditions are met.
- **Mark**: Marks the packets if the conditions are met.
- **Transmit**: Transmits the packets if the conditions are met.

Step 14 If for **Violate Action** you chose **Mark**, perform the following substeps:

a) For **Violate mark CoS**, enter the class of service for packets that violated the conditions.
b) For **Violate mark dscp**, enter the differentiated services code point (DSCP) for packets that violated the conditions.

Step 15 For **Sharing Mode**, choose **Shared Policer**.

Shared Policer mode allows you to apply the same policing parameters to several interfaces simultaneously. The **Dedicated Policer** mode is not supported for Layer 2 interfaces.

Step 16 For **Rate**, enter the rate at which to allow packets are allowed into the system and choose the unit per packet.

Step 17 For **Burst**, enter the number of packets allowed at the line rate during a burst and choose the unit per packet.

Step 18 If for **Type** you chose **2 Rate 3 Color**, perform the following substeps:

a) For **Peak Rate**, enter the peak information rate, which is the rate above which data traffic is negatively affected, and choose the unit per packet.
b) For **Excessive Burst**, enter the size that a traffic burst can reach before all traffic exceeds the peak information rate, and choose the unit per packet.

Step 19 Click **Submit**.

This completes DPP configuration for Layer 2. The data plane policy can now be mapped to an interface policy group that maps to a Layer 2 interface.

Configuring Data Plane Policing for Layer 3 Interface Using the APIC GUI

Before you begin

The tenant, VRF, and external routed network where you configure the Data Plane Policing policy is already created.

The Data Plane Policing policy must be added to a policy group and the policy group mapped to an interface profile to apply the L3 DPP policy.

Procedure

Step 1 In the **Navigation** pane, click on **Tenant_name** > **Networking** > **External Routed Network** > **Network_name** > **Logical Node Profiles** > **Logical Node Profile_name** > **Logical Interface Profiles**, and perform the following actions.

a) Right-click on **Logical Interface Profiles**, and select **Create Interface Profile**.
b) In the **Create Interface Profile** dialog box, in the **Name** field, enter a name for the profile.
c) Next to **Ingress Data Plane Policing Policy**, select **Create Data Plane Policing Policy**.
d) In the **Name** field, enter a name for the policy.
e) In the **Administrative State** field, click **enabled**.
f) Next to **Policer Mode**, select a button for either **Bit Policer** or **Packet Policer**.
g) Next to **Type**, select a button for **1 Rate 2 Color** or **2 Rate 3 Color**.

Switch models ending in EX/FX (for example: N9K-C93180YC-EX) and subsequent models don't support 2 Rate 3 Color).

a) The administrator can configure the CoS and DSCP values in the **Conform** and **Violate** fields.
b) In the **Sharing Mode** field, select the policer mode.

Note
Shared Policer Mode allows you to apply the same policing parameters to several interfaces simultaneously.

c) Next to the **Burst**, **Excessive Burst** and **Rate** fields, select the drop down arrow to set the per packet rate for **1 Rate 2 Color** policy type.

Note
For **2 Rate 3 Color** policy type, the **Peak Rate** field is added.

d) Click **Submit**.

Step 2 Expand the **Routed Interfaces** table, in the **Path** field navigate to the interface to apply the policy and perform the following actions:

a) Next to **IPv4/Ipv6 Preferred Address**, enter a subnet IP address.
b) Click **OK**.
c) Click on the **SVI** tab and expand, in the **Path** field navigate to the interface to apply the policy.
d) Next to **Encap**, enter the VLAN name.

e) Next to **IPv4/Ipv6 Preferred Address**, enter a subnet IP address.
f) Click **OK**.
g) Expand the **Routed Sub-Interfaces** tab, and follow the same configuration steps as for the Routed Interfaces.
h) Click **OK**. This completes DPP configuration for L3.

Configuring Data Plane Policing Using the REST API

To police the L2 traffic coming in to the Leaf:

```
<!--  api/node/mo/uni/.xml  -->
<infraInfra>
<qosDppPol name="infradpp5" burst="2000" rate="2000" be="400" sharingMode="shared"/>
<!--
 List of nodes. Contains leaf selectors. Each leaf selector contains list of node blocks
-->
<infraNodeP name="leaf1">
<infraLeafS name="leaf1" type="range">
<infraNodeBlk name="leaf1" from_="101" to_="101"/>
</infraLeafS>
<infraRsAccPortP tDn="uni/infra/accportprof-portselector1"/>
</infraNodeP>
<!--
 PortP contains port selectors. Each port selector contains list of ports. It
       also has association to port group policies
-->
<infraAccPortP name="portselector1">
<infraHPortS name="pselc" type="range">
<infraPortBlk name="blk" fromCard="1" toCard="1" fromPort="48" toPort="49"></infraPortBlk>
<infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-portSet2"/>
</infraHPortS>
</infraAccPortP>
<!--  FuncP contains access bundle group policies  -->
<infraFuncP>
<infraAccPortGrp name="portSet2">
<infraRsQosIngressDppIfPol tnQosDppPolName="infradpp5"/>
</infraAccPortGrp>
</infraFuncP>
</infraInfra>
```

To police the L2 traffic going out of the Leaf:

```
<!--  api/node/mo/uni/.xml  -->
<infraInfra>
<qosDppPol name="infradpp2" burst="4000" rate="4000"/>
<!--
 List of nodes. Contains leaf selectors. Each leaf selector contains list of node blocks
-->
<infraNodeP name="leaf1">
<infraLeafS name="leaf1" type="range">
<infraNodeBlk name="leaf1" from_="101" to_="101"/>
</infraLeafS>
<infraRsAccPortP tDn="uni/infra/accportprof-portselector2"/>
</infraNodeP>
<!--
 PortP contains port selectors. Each port selector contains list of ports. It
       also has association to port group policies
```

```
-->
<infraAccPortP name="portselector2">
<infraHPortS name="pselc" type="range">
<infraPortBlk name="blk" fromCard="1" toCard="1" fromPort="37" toPort="38"></infraPortBlk>
<infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-portSet2"/>
</infraHPortS>
</infraAccPortP>
<!--  FuncP contains access bundle group policies  -->
<infraFuncP>
<infraAccPortGrp name="portSet2">
<infraRsQosEgressDppIfPol tnQosDppPolName="infradpp2"/>
</infraAccPortGrp>
</infraFuncP>
</infraInfra>
```

To police the L3 traffic coming in to the Leaf:

```
<!--  api/node/mo/uni/.xml  -->
<fvTenant name="dppTenant">
<qosDppPol name="gmeo" burst="2000" rate="2000"/>
<l3extOut name="Outside">
<l3extInstP name="extroute"/>
<l3extLNodeP name="borderLeaf">
<l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="10.0.0.1">
<ipRouteP ip="0.0.0.0">
<ipNexthopP nhAddr="192.168.62.2"/>
</ipRouteP>
</l3extRsNodeL3OutAtt>
<l3extLIfP name="portProfile">
<l3extRsPathL3OutAtt addr="192.168.40.1/30" ifInstT="l3-port"
tDn="topology/pod-1/paths-101/pathep-[eth1/40]"/>
<l3extRsPathL3OutAtt addr="192.168.41.1/30" ifInstT="l3-port"
tDn="topology/pod-1/paths-101/pathep-[eth1/41]"/>
<l3extRsIngressQosDppPol tnQosDppPolName="gmeo"/>
</l3extLIfP>
</l3extLNodeP>
</l3extOut>
</fvTenant>
```

To police the L3 traffic going out of the Leaf:

```
<!--  api/node/mo/uni/.xml  -->
<fvTenant name="dppTenant">
<qosDppPol name="gmeo" burst="2000" rate="2000"/>
<l3extOut name="Outside">
<l3extInstP name="extroute"/>
<l3extLNodeP name="borderLeaf">
<l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="10.0.0.1">
<ipRouteP ip="0.0.0.0">
<ipNexthopP nhAddr="192.168.62.2"/>
</ipRouteP>
</l3extRsNodeL3OutAtt>
<l3extLIfP name="portProfile">
<l3extRsPathL3OutAtt addr="192.168.40.1/30" ifInstT="l3-port"
tDn="topology/pod-1/paths-101/pathep-[eth1/40]"/>
<l3extRsPathL3OutAtt addr="192.168.41.1/30" ifInstT="l3-port"
tDn="topology/pod-1/paths-101/pathep-[eth1/41]"/>
<l3extRsEgressQosDppPol tnQosDppPolName="gmeo"/>
</l3extLIfP>
</l3extLNodeP>
</l3extOut>
</fvTenant>
```

Configuring Data Plane Policing Using NX-OS Style CLI

Procedure

Step 1 Configure a Layer 2 port to carry one EPG.

Example:

```
apic1# conf t
apic1(config)# vlan-domain test
apic1(config-vlan)# vlan 1000-2000
apic1(config-vlan)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# vlan-domain member test
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# tenant test1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# application ap1
apic1(config-tenant-app)# epg e1
apic1(config-tenant-app-epg)# bridge-domain member bd1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# switchport trunk allowed vlan 1001 tenant test1 application ap1 epg
 e1
apic1(config-leaf-if)# switchport trunk allowed vlan 1501 tenant test1 application ap1 epg
 e1
# Now the port leaf 101 ethernet 1/10 carries two vlan mapped both to the same
Tenant/Application/EPG
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

a) Create a policy-map to apply to the interface.

Example:

```
apic1(config)# policy-map type data-plane qosTest
apic1(config-pmap-dpp)# set burst 2400 mega
apic1(config-pmap-dpp)# set cir 70 mega

apic1(config-pmap-dpp)# set sharing-mode shared
apic1(config-pmap-dpp)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# service-policy type data-plane input qosTest
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# policy-map type data-plane qosTest2
apic1(config-pmap-dpp)# set cir 78 mega
apic1(config-pmap-dpp)# exit
```

```
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# service-policy type data-plane output qosTest2
apic1(config-leaf-if)# end
```

b) Visualize the policy configured.

Example:

```
apic1# show policy-map type data-plane infra
Type data-plane policy-maps
====================
Global Policy
policy-map type data-plane default
    set burst unspecified
    set conform-cos-transmit unspecified
    set conform-dscp-transmit unspecified
    set conform transmit
    set excessive-burst unspecified
    set exceed-cos-transmit unspecified
    set exceed-dscp-transmit unspecified
    set exceed drop
    set mode byte
    set pir 0
    set cir 78 mega
    set type 1R2C
    set violate-cos-transmit unspecified
    set violate-dscp-transmit unspecified
    set violate drop
Global Policy
policy-map type data-plane qosTest
    set burst 2400 mega
    set cir 78 mega
    set conform-cos-transmit unspecified
    set conform-dscp-transmit unspecified
    set conform transmit
    set excessive-burst unspecified
    set exceed-cos-transmit unspecified
    set exceed-dscp-transmit unspecified
    set exceed drop
    set mode byte
    set pir 0
    set type 1R2C
    set violate-cos-transmit unspecified
    set violate-dscp-transmit unspecified
    set violate drop
Global Policy
policy-map type data-plane qosTest2
    set burst unspecified
    set conform-cos-transmit unspecified
    set conform-dscp-transmit unspecified
    set conform transmit
    set excessive-burst unspecified
    set exceed-cos-transmit unspecified
    set exceed-dscp-transmit unspecified
    set exceed drop
    set mode byte
    set pir 0
    set cir 78 mega
    set type 1R2C
    set violate-cos-transmit unspecified
    set violate-dscp-transmit unspecified
    set violate drop
```

c) Show running-config.

Example:

```
apic1# show runn policy-map
# Command: show running-config policy-map
# Time: Fri Jan 29 19:26:18 2016
  policy-map type data-plane default
    exit
  policy-map type data-plane qosTest
    set burst 2400 mega
    set cir 78 mega
    no shutdown
    exit
  policy-map type data-plane qosTest2
    set cir 78 mega
    no shutdown
    exit
apic1# show runn leaf 101
# Command: show running-config leaf 101
# Time: Fri Jan 29 19:26:29 2016
  leaf 101
    interface ethernet 1/10
      vlan-domain member test
      switchport trunk allowed vlan 1501 tenant test1 application ap1 epg e1
      service-policy type data-plane input qosTest
      service-policy type data-plane output qosTest2
      exit
    exit
```

Step 2 Preparation to configure Layer 3 ports.

Example:

```
apic1# conf t
apic1(config)# vlan-domain l3ports
apic1(config-vlan)# vlan 3000-3001
apic1(config-vlan)# exit
apic1(config)# tenant l3test1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant l3test1 vrf v1
apic1(config-leaf-vrf)# exit
# Configure a physical Layer 3 port
apic1(config-leaf)# interface ethernet 1/20
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vlan-domain member l3ports
apic1(config-leaf-if)# vrf member tenant l3test1 vrf v1
apic1(config-leaf-if)# ip address 56.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2000::1/64  preferred
apic1(config-leaf-if)# exit
# Configure base interface for L3 subinterfaces
apic1(config-leaf)# interface ethernet 1/21
apic1(config-leaf-if)# vlan-domain member l3ports
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# exit
# Configure a Layer 3 subinterface
apic1(config-leaf)# interface ethernet 1/21.3001
apic1(config-leaf-if)# vrf member tenant l3test1 vrf v1
apic1(config-leaf-if)# ip address 60.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
apic1(config-leaf-if)# exit
# Configure a Switched Vlan Interface
apic1(config-leaf)# interface vlan 3000
```

```
apic1(config-leaf-if)# vrf member tenant l3test1 vrf v1
apic1(config-leaf-if)# ip address 70.1.1.1/24
apic1(config-leaf-if)# ipv6 address 3000::1/64 preferred
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

a) Configure the policer in the tenant for Layer 3 usage.

Example:

```
apic1(config)# tenant l3test1
apic1(config-tenant)# policy-map type data-plane iPol
apic1(config-tenant-pmap-dpp)# set cir 56 mega
apic1(config-tenant-pmap-dpp)# set burst 2000 kilo
apic1(config-tenant-pmap-dpp)# exit
apic1(config-tenant)# policy-map type data-plane ePol
apic1(config-tenant-pmap-dpp)# set burst 2000 kilo
apic1(config-tenant-pmap-dpp)# set cir 56 mega
apic1(config-tenant-pmap-dpp)# exit
apic1(config-tenant)# exit
```

b) Apply policer on a Layer 3 interface

Example:

```
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 1/20
apic1(config-leaf-if)# service-policy type data-plane input iPol
apic1(config-leaf-if)# service-policy type data-plane output ePol
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/21.3001
apic1(config-leaf-if)# service-policy type data-plane input iPol
apic1(config-leaf-if)# service-policy type data-plane output ePol
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface vlan 3000
apic1(config-leaf-if)# service-policy type data-plane input iPol
apic1(config-leaf-if)# service-policy type data-plane output ePol
apic1(config-leaf-if)# end
```

c) Show commands for policers used on a Layer 3 interface.

Example:

```
apic1# show tenant l3test1 policy-map type data-plane
Type data-plane policy-maps
===================
Policy in Tenant: l3test1
policy-map type data-plane ePol
    set burst 2000 kilo
    set conform-cos-transmit unspecified
    set conform-dscp-transmit unspecified
    set conform transmit
    set excessive-burst unspecified
    set exceed-cos-transmit unspecified
    set exceed-dscp-transmit unspecified
    set exceed drop
    set mode byte
    set pir 0
    set cir 56 mega
    set type 1R2C
    set violate-cos-transmit unspecified
    set violate-dscp-transmit unspecified
    set violate drop
Policy in Tenant: l3test1
policy-map type data-plane iPol
    set burst 2000 kilo
```

```
    set burst unspecified
    set conform-cos-transmit unspecified
    set conform-dscp-transmit unspecified
    set conform transmit
    set excessive-burst unspecified
    set exceed-cos-transmit unspecified
    set exceed-dscp-transmit unspecified
    set exceed drop
    set mode byte
    set pir 0
    set cir 56 mega
    set type 1R2C
    set violate-cos-transmit unspecified
    set violate-dscp-transmit unspecified
    set violate drop
```

d) Show running-config for policers used for Layer 3.

Example:

```
apic1# show runn tenant l3test1
# Command: show running-config tenant l3test1
# Time: Fri Jan 29 19:48:20 2016
  tenant l3test1
    vrf context v1
      exit
    policy-map type data-plane ePol
      set burst 2000 kilo
      set cir 56 mega
      no shutdown
      exit
    policy-map type data-plane iPol
      set burst 2000 kilo
      set cir 56 mega
      no shutdown
      exit
    exit
apic1# show running-config leaf 102
# Command: show running-config leaf 102
# Time: Fri Jan 29 19:48:33 2016
  leaf 102
    vrf context tenant l3test1 vrf v1
      exit
    interface vlan 3000
      vrf member tenant l3test1 vrf v1
      ip address 70.1.1.1/24
      ipv6 address 3000::1/64 preferred
      bfd ip tenant mode
      bfd ipv6 tenant mode
      service-policy type data-plane input iPol
      service-policy type data-plane output ePol
      exit
    interface ethernet 1/20
      vlan-domain member l3ports
      no switchport
      vrf member tenant l3test1 vrf v1
      ip address 56.1.1.1/24
      ipv6 address 2000::1/64 preferred
      bfd ip tenant mode
      bfd ipv6 tenant mode
      service-policy type data-plane input iPol
      service-policy type data-plane output ePol
      exit
    interface ethernet 1/21
      vlan-domain member l3ports
```

```
      no switchport
      bfd ip tenant mode
      bfd ipv6 tenant mode
      exit
    interface ethernet 1/21.3001
      vrf member tenant l3test1 vrf v1
      ip address 60.1.1.1/24
      ipv6 address 2001::1/64 preferred
      bfd ip tenant mode
      bfd ipv6 tenant mode
      service-policy type data-plane input iPol
      service-policy type data-plane output ePol
    exit
    exit
apic1#
```

Data Plane Policing at the Endpoint Group Level

Data Plane Policing (DPP) can be applied to an endpoint group (EPG). The policing of the traffic is applied to all the EPG members on every leaf switch where the EPG is deployed.

Prior to the 3.2(1) release, each EPG member had its own policer. Beginning in the 3.2(1) release, the behavior is dependent on the sharing-mode property (if configured through the CLI or GUI) on the Data Plane Policer. If that is set to **dedicated**, then the situation is similar to before the 3.2(1) release. If the sharing-mode is set to **shared**, then all the members in the same slice using the same Data Plane Policer policy use the hardware policer on the leaf switch.

For example, an EPG has the following members:

- Leaf 101, Eth1/1, vlan-300
- Leaf 101, Eth1/2, vlan-301
- Leaf 102, Eth1/2, vlan-500

In this case, each member will limit the traffic according to the policer, independent from the other members. If the Data Plane Policer has the sharing-mode set to **shared**, then all the members in the same slice above use only one policer on the leaf switch.

The Data Plane Policer works independently on Leaf 101 and Leaf 102 if the sharing-mode is set to **dedicated**. For example:

- Policer-A (100Mbps policing) is applied to EPG1 (Leaf101 e1/1 vlan-300 and e1/2 vlan-301. Leaf 102 e1/2 vlan-500)
- Leaf 101: police traffic at the EPG1 level, which is applied to traffic through E1/1 vlan-300 and E1/2 vlan-301 (100Mbps for each interface).
- Leaf 102: police traffic at the EPG1 level, which is applied to traffic through E1/2 vlan-500 (another 100Mbps for each interface).

The total is up to 300Mbps for EPG1.

If the sharing-mode is set to **shared**, 100Mbps is shared across EPGs using the same policer if the interfaces are in the same slice. For example:

- Policer-A (100Mbps policing) applied to EPG1 and EPG2.
- Leaf 101: police traffic at EPG1 and EPG2 in total.
- Leaf 102: police traffic at EPG1 and EPG2 in total.

The total is up to 200Mbps for EPG1 and EPG2 if the interfaces are in the same slice.

The following are limitations for Data Plane Policing at the EPG level:

- EPG policer feature is supported with switch models that have -EX, -FX, or later suffixes in the product ID.
- Egress traffic policing is not supported for the EPG level policer.
- Policer mode **Packet-per-second** is not supported.
- Policer type 2R3C is not supported in EPG policer.
- Policer is not supported when **intra-EPG isolation-enforced** is applied to the EPG.
- The scale limit allows for 128 EPG policers supported per node.

Configuring Data Plane Policing at the Endpoint Group Level Using CLI

Procedure

Define the policer:

Example:

```
apic1# conf t
apic1(config)# vlan-domain test
apic1(config-vlan)# vlan 1000-2000
apic1(config-vlan)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# vlan-domain member test
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# tenant test1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config)# policy-map type data-plane pol1
apic1(config-pmap-dpp)# set burst 2400 mega
apic1(config-pmap-dpp)# set cir 78 mega
apic1(config-pmap-dpp)# exit
apic1(config-tenant)# application ap1
apic1(config-tenant-app)# epg e1
apic1(config-tenant-app-epg)# bridge-domain member db1
apic1(config-tenant-app-epg)# service-policy type data-plane poll
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
```

```
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# switchport trunk allowed vlan 1001 tenant test1 application ap1 epg
 e1
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

Configuring Data Plane Policing at the Endpoint Group Level Using the APIC GUI

Procedure

In the **Tenants** pane, click on **Tenant_name** > **Policies** > **Protocol** > **Data Plane Policing**. Right-click on **Data Plane Policing** to **Create Data Plane Policing Policy**.

a) In the **Name** field, enter a name for the policy.
b) In the **Administrative State** field, click **enabled**.
c) Next to **Policer Mode**, select a button for either **Bit Policer** or **Packet Policer**.
d) Next to **Type**, select a button for **1 Rate 2 Color**.
e) For **Conform Action**, select **Drop**, **Mark**, or **Transmit**.
f) The administrator can configure the CoS and DSCP values in the **Conform** and **Violate** fields.
g) Next to the **Burst**, **Excessive Burst** and **Rate** fields, click the drop down arrow to select from the following:

- **Bytes/Packets**
- **Kilo Bytes/Packets**
- **Mega Bytes/Packets**
- **Giga Bytes/Packets**
- **Milli Seconds**
- **Micro Seconds**

Configuring Data Plane Policing at the Endpoint Group Level Using Rest API

To police the traffic coming into the leaf switch:

```
<!-- api/node/mo/.xml -->
<polUni>
  <fvTenant  name="t1">

    <qosDppPol name="gmeo" burst="2000" rate="2000"/>
    <fvAp name="ap1">
      <fvAEPg name="ep1">
        <fvRsDppPol tnQosDppPolName="gmeo"/>
      </fvAEPg>
    </fvAp>
```

```
  </fvTenant>
</polUni>
```

Accessing Statistics for the Data Plane Policer at the Endpoint Group Level in the GUI

DPP at the EPG level is used to police traffic at the EPG member level. As such, statistics are integral in ensuring the policer is dropping substantial traffic. Statistics are reported at the EPG member level for fine granularity.

Procedure

Step 1 In the **Tenants** pane, click on **Tenant_name** > **Application EPGs** > **EPG Members** > **Static EPG Members** .

Step 2 Select a node.

Step 3 Click **Select Stats**.

a) Select a **Sampling Interval** unit of time.

b) From the **Available** policer attributes, use the arrows to choose the attributes. You can select up to two attributes.

c) Click **Submit**.

What to do next

You will see a graphical representation of the DPP statistics.

CHAPTER 13

HTTPS Access

This chapter contains the following sections:

- Overview, on page 151
- Configuring Custom Certificate Guidelines, on page 151
- Modifying the SSL Cipher Configuration, on page 152
- Configuring a Custom Certificate for Cisco ACI HTTPS Access Using the GUI, on page 153
- Configuring the Default SSL Protocols and Diffie-Hellman Key Exchange Using the GUI, on page 156
- Enabling Certificate Based Authentication Using the NX-OS CLI, on page 156

Overview

This article provides an example of how to configure a custom certificate for HTTPS access when using Cisco ACI.

Configuring Custom Certificate Guidelines

- Exporting a private key that is used to generate a Certificate Signing Request (CSR) on the Cisco Application Policy Infrastructure Controller (APIC) is not supported. If you want to use the same certificate on multiple servers through a wildcard in the Subject Alternative Name (SAN) field, such as "*cisco.com," by sharing the private key that was used to generate the CSR for the certificate, generate the private key outside of Cisco Application Centric Infrastructure (ACI) fabric and import it to the Cisco ACI fabric.
- You must download and install the public intermediate and root CA certificates before generating a Certificate Signing Request (CSR). Although a root CA Certificate is not technically required to generate a CSR, Cisco requires the root CA certificate before generating the CSR to prevent mismatches between the intended CA authority and the actual one used to sign the CSR. The Cisco APIC verifies that the certificate submitted is signed by the configured CA.
- To use the same public and private keys for a renewed certificate generation, you must satisfy the following guidelines:
  - You must preserve the originating CSR as it contains the public key that pairs with the private key in the key ring.
  - The same CSR used for the originating certificate must be resubmitted for the renewed certificate if you want to re-use the public and private keys on the Cisco APIC.

  - Do not delete the original key ring when using the same public and private keys for the renewed certificate. Deleting the key ring will automatically delete the associated private key used with CSRs.

- Cisco ACI Multi-Site, VCPlugin, VRA, and SCVMM are not supported for certificate-based authentication.
- Only one SSL certificate is allowed per Cisco APIC cluster.
- You must disable certificate-based authentication before downgrading to release 4.0(1) from any later release.
- To terminate the certificate-based authentication session, you must log out and then remove the CAC card.
- The custom certificate configured for the Cisco APIC will be deployed to the leaf and spine switches. If the URL or DN that is used to connect to the fabric node is within the **Subject** or **Subject Alternative Name** field, the fabric node will be covered under the certificate.
- The Cisco APIC GUI can accept a certificate with a maximum size of 4k bytes.
- When a self-signed SSL certificate that you are using for HTTPS access expires, the certificate gets renewed automatically.

Modifying the SSL Cipher Configuration

SSL ciphers can be enabled, disabled, or removed entirely. Depending on the desired cipher settings, you should understand which exact combination is required. Disabling and enabling ciphers in a manner that results in no ciphers remaining is a misconfiguration and will result in NGINX failing validation.

NGINX uses the OpenSSL cipher list format. For information about the format, go to the OpenSSL website.

Mapping the Cisco APIC SSL Configuration Options to the Cipher List Formatting

Enabling a cipher results in the cipher being written to the NGINX configuration file. Disabling a cipher results in the cipher being written in the NGINX configuration file with a preceding exclamation mark (!). For example, disabling "EEDCH" will cause it to be written as "!EEDCH". Removing a cipher will result in the cipher not being written the NGINX configuration file at all.

Note As stated in the OpenSSL cipher list format document, "If ! is used then the ciphers are permanently deleted from the list. The ciphers deleted can never reappear in the list even if they are explicitly stated." This can result in the removal of combination ciphers referencing the one that was set to "Disabled," regardless of the ciphers' "Enabled" state.

Example: Disabling "EEDCH," but enabling "EECDH+aRSA+SHA384." This will cause the following to be written to the NGINX configuration file: "!EEDCH:EECDH+aRSA+SHA384". The "!EEDCH" will prevent "EECDH+aRSA+SHA384" from ever being added. This will result in no ciphers being used, which will fail NGINX validation and prevent NGINX updates from succeeding, such as applying custom HTTPS certificates.

Testing the Cipher List Format Before Modifying the Cisco APIC SSL Configuration

Before making any cipher modifications to the Cisco Application Policy Infrastructure Controller (APIC), validate the results of the planned cipher combination using the `openssl ciphers -V 'cipher_list'` command and ensure that the cipher output matches your desired result.

Example:

```
apic# openssl ciphers -V 'EECDH+aRSA+SHA256:EECDH+aRSA+SHA384'
          0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(128)
Mac=SHA256
          0xC0,0x28 - ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(256)
Mac=SHA384
```

If your tested cipher list results in an error or "no cipher match," do not apply this configuration to the Cisco APIC. Doing so can result in NGINX issues with symptoms including making the Cisco APIC GUI inaccessible and breaking custom certificate application.

Example:

```
apic# openssl ciphers -V '!EECDH:EECDH+aRSA+SHA256:EECDH+aRSA+SHA384'
Error in cipher list
132809172158128:error:1410D0B9:SSL routines:SSL_CTX_set_cipher_list:no cipher
match:ssl_lib.c:1383:
```

Configuring a Custom Certificate for Cisco ACI HTTPS Access Using the GUI

CAUTION: PERFORM THIS TASK ONLY DURING A MAINTENANCE WINDOW AS THERE IS A POTENTIAL FOR DOWNTIME. The downtime affects access to the Cisco Application Policy Infrastructure Controller (APIC) cluster and switches from external users or systems and not the Cisco APIC to switch connectivity. The NGINX process on the switches will also be impacted, but that will be only for external connectivity and not for the fabric data plane. Access to the Cisco APIC, configuration, management, troubleshooting, and such will be impacted. The NGINX web server running on the Cisco APIC and switches will be restarted during this operation.

Before you begin

Determine from which authority you will obtain the trusted certification so that you can create the appropriate Certificate Authority.

Procedure

Step 1 On the menu bar, choose **Admin** > **AAA**.

Step 2 In the **Navigation** pane, choose **Security**.

Step 3 In the **Work** pane, choose **Public Key Management** > **Certificate Authorities** > **Create Certificate Authority**.

Step 4 In the **Create Certificate Authority** dialog box, in the **Name** field, enter a name for the certificate authority.

Step 5 In the **Certificate Chain** field, copy the intermediate and root certificates for the certificate authority that will sign the Certificate Signing Request (CSR) for the Cisco APIC.

The certificate should be in Base64 encoded X.509 (CER) format. The intermediate certificate is placed before the root CA certificate. It should look similar to the following example:

```
-----BEGIN CERTIFICATE-----
<Intermediate Certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Root CA Certificate>
-----END CERTIFICATE-----
```

Step 6 Click **Submit**.

Step 7 In the **Navigation** pane, choose **Public Key Management** > **Key Rings**.

Step 8 In the **Work** pane, choose **Actions** > **Create Key Ring**.

The key ring enables you to manage a private key (imported from external device or internally generated on APIC), a CSR generated by the private key, and the certificate signed via the CSR.

Step 9 In the **Create Key Ring** dialog box, in the **Name** field, enter a name.

Step 10 In the **Certificate** field, do not add any content if you will generate a CSR using the Cisco APIC through the key ring. Alternately, add the signed certificate content if you already have one that was signed by the CA from the previous steps by generating a private key and CSR outside of the Cisco APIC,

Step 11 In the **Modulus** field, click the radio button for the desired key strength.

Step 12 In the **Certificate Authority** field, from the drop-down list, choose the certificate authority that you created earlier, then click **Submit**.

Step 13 In the **Private Key** field, do not add any content if you will generate a CSR using the Cisco APIC through the key ring. Alternately, add the private key used to generate the CSR for the signed certificate that you entered in step 10.

Note
Do not delete the key ring. Deleting the key ring will automatically delete the associated private key used with CSRs.

If you have not entered the signed certificate and the private key, in the **Work** pane, in the **Key Rings** area, the **Admin State** for the key ring created displays **Started**, waiting for you to generate a CSR. Proceed to step 14.

If you entered both the signed certificate and the private key, in the **Key Rings** area, the **Admin State** for the key ring created displays **Completed**. Proceed to step 23.

Step 14 In the **Navigation** pane, choose **Public Key Management** > **Key Rings** > ***key_ring_name***.

Step 15 In the **Work** pane, choose **Actions** > **Create Certificate Request**.

Step 16 In the **Subject** field, enter the common name (CN) of the CSR.

You can enter the fully qualified domain name (FQDN) of the Cisco APICs using a wildcard, but in a modern certificate, we generally recommend that you enter an identifiable name of the certificate and enter the FQDN of all Cisco APICs in the **Alternate Subject Name** field (also known as the *SAN* – Subject Alternative Name) because many modern browsers expect the FQDN in the SAN field.

Step 17 In the **Alternate Subject Name** field, enter the FQDN of all Cisco APICs, such as "DNS:apic1.example.com,DNS:apic2.example.com,DNS:apic3.example.com" or "DNS:*example.com".

Alternatively, if you want SAN to match an IP address, enter the Cisco APICs' IP addresses with the following format:

```
IP:192.168.2.1
```

You can use DNS names, IPv4 addresses, or a mixture of both in this field. IPv6 addresses are not supported.

Step 18 Fill in the remaining fields as appropriate.

Note
Check the online help information available in the **Create Certificate Request** dialog box for a description of the available parameters.

Step 19 Click **Submit**.

Inside the same key ring, the **Associated Certificate Request** area is now displayed with the **Subject**, **Alternate Subject Name** and other fields you entered in the previous steps along with the new field **Request**, which contains the content of the CSR that is tied to this key ring. Copy the content from the **Request** field to submit the content to the same certificate authority that is tied to this key ring for signing.

Step 20 In the **Navigation** pane, choose **Public Key Management** > **Key Rings** > ***key_ring_name***.

Step 21 In the **Work** pane, in the **Certificate** field, paste the signed certificate that you received from the certificate authority.

Step 22 Click **Submit**.

Note
If the CSR was not signed by the Certificate Authority indicated in the key ring, or if the certificate has MS-DOS line endings, an error message is displayed and the certificate is not accepted. Remove the MS-DOS line endings.

The key is verified, and in the **Work** pane, the **Admin State** changes to **Completed** and is now ready for use in the HTTP policy.

Step 23 On the menu bar, choose **Fabric** > **Fabric Policies**.

Step 24 In the Navigation pane, choose **Pod Policies** > **Policies** > **Management Access** > **default**.

Step 25 In the **Work** pane, in the **Admin Key Ring** drop-down list, choose the desired key ring.

Step 26 (Optional) For Certificate based authentication, in the **Client Certificate TP** drop-down list, choose the previously created Local User policy and click **Enabled** for **Client Certificate Authentication state**.

Step 27 Click **Submit**.
All web servers restart. The certificate is activated, and the non-default key ring is associated with HTTPS access.

What to do next

You must remain aware of the expiration date of the certificate and take action before it expires. To preserve the same key pair for the renewed certificate, you must preserve the CSR as it contains the public key that pairs with the private key in the key ring. Before the certificate expires, the same CSR must be resubmitted. Do not delete or create a new key ring as deleting the key ring will delete the private key stored internally on the Cisco APIC.

Configuring the Default SSL Protocols and Diffie-Hellman Key Exchange Using the GUI

This procedure configures the default SSL protocols and Diffie-Hellman key exchange. You must configure these parameters based on the security policy of your organization and the needs of any applications that you use.

Procedure

Step 1 On the menu bar, choose **Fabric** > **Fabric Policies**.

Step 2 In the **Navigation** pane, choose **Policies** > **Pod** > **Management Access** > **default**.

Step 3 In the **Work** pane, find the **HTTPS** section.

a) For **SSL Protocols**, put a check in the boxes for the transport layer security (TLS) versions that your network allows. Leave the box empty for any TLS version that your network does not allow.

b) For **DH Param**, choose the desired key size (in bits).

Choosing one of the key sizes enables the standard Diffie-Hellman (DH) key exchange in addition to the elliptic-curve Diffie-Hellman (ECDH) key exchange and uses the chosen number of bits for the DH key exchange. Choosing **None** instead uses only the elliptic-curve ECDH key exchange. In any case, ECDH always uses 256 bits.

c) Click **Submit**.

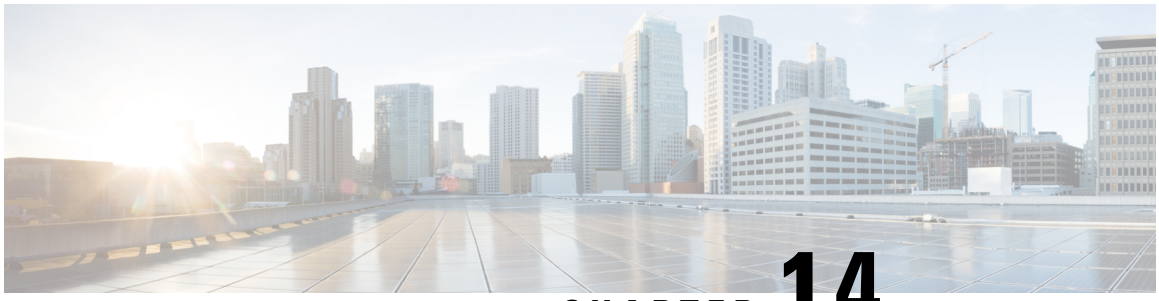
Enabling Certificate Based Authentication Using the NX-OS CLI

Procedure

To enable Certificate Based authentication:

Example:

```
To enable CAC for https access:
  configure terminal
   comm-policy default
    https
     client-cert-ca <ca name>
     client-cert-state-enable
 To disable:
  configure terminal
   comm-policy default
    https
     no client-cert-state-enable
     no client-cert-ca
```



CHAPTER 14

Additional ACI Security Features

This chapter contains the following sections:

- Additional Security Features, on page 157
- Turning Off Generated Session Log Files in APIC, on page 157

Additional Security Features

The following are a list of security features currently supported in ACI but documented in other configuration guides found at https://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html:

- For **Contract** configuration see the *Cisco APIC Basic Configuration Guide, Release 3.x* and the *Operating Cisco Application Centric Infrastructure*.
- For **EPG Communication Rules** see the *Use vzAny to Automatically Apply Communication Rules to all EPGs in a VRF* Knowledge-Based article.
- For **In-Band and Out-of-Band Management Access** see the *Cisco APIC and Static Management Access* Knowledge-Based article, and the *Cisco APIC Layer 4 to Layer 7 Services Deployment Guide, Release 2.2(3)*.
- For **Intra-EPG Isolation Enforcement** see the *Cisco ACI Virtualization Guide, Release 3.0(1)*.
- For **Traffic Storm Control** see the *Cisco APIC Layer 2 Networking Configuration Guide*.

Turning Off Generated Session Log Files in APIC

This section describes how turn off the generated logs in APIC. If you have configured any sort of monitoring for your fabric, you will see the following log file:

```
Body of session record log example:
From-127.0.0.1-client-type-REST-Success
```

To turn off the generated session log files in APIC, perform the following steps:

Procedure

Step 1 On the menu bar, choose **ADMIN** > **AAA**.

Step 2 In the **AAA** pane, click **Security**.

Step 3 In the **User Management – Security** pane, verify that the default **Management Settings** pane is chosen.

Step 4 In the **Include Refresh in Session Records** field, uncheck the box to disable the generated session log files.

Step 5 Click **Submit**.

Step 6 Click **Submit Changes**.