Cisco Nexus 1000 Virtual Edge for VMware vSphere Security Configuration Guide, Release 5.2(1)SV5(1.2)

First Published: 2019-05-03

Last Modified: 2019-12-03

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

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 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.

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/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

CONTENTS

CHAPTER 1 **Overview** 1

- User Accounts 1
- Virtual Service Domain 1
- Authentication, Authorization, and Accounting 1
- RADIUS Security Protocol 2
- TACACS+ Security Protocol 2
- SSH 2
- Telnet 3
- Access Control Lists 3
- Layer 3 Security 3

CHAPTER 2 **Configuring IP ACLs** 5

- Information About ACLs 5
  - ACL Types and Applications 5
  - Active Ports and Services on Nexus 1000V VSM 6
  - Order of ACL Application 6
  - Rules 6
    - Source and Destination 6
    - Protocols 6
    - Implicit Rules 7
    - Additional Filtering Options 7
    - Sequence Numbers 7
  - Statistics 8
  - ACL Logging 8
    - ACL Flows 9
    - Syslog Messages 10

Prerequisites for IP ACLs 11
Guidelines and Limitations for IP ACLs 11
Default Settings for IP ACLs 11
Configuring IP ACLs 11
Creating an IP ACL 11
Changing an IP ACL 12
Removing an IP ACL 14
Changing Sequence Numbers in an IP ACL 15
Adding an IP ACL to a Port Profile 16
Applying an IP ACL to the Management Interface 17
Configuring ACL Logging 19
Disabling ACL Logging 19
Configuring a Time Interval for Accumulating Packet Counters 19
Configuring Flows 20
Syslog Server Severity Levels 20
Setting the Severity Level for a Syslog Message 21
Verifying the IP ACL Configuration 21
Monitoring IP ACLs 22
Configuration Example for IP ACL 22
CHAPTER 3 Configuring MAC ACLs 25
Information About MAC ACLs 25
Prerequisites for MAC ACLs 25
Guidelines and Limitations for MAC ACLs 25
Default Settings for MAC ACLs 25
Configuring MAC ACLs 26
Creating a MAC ACL 26
Changing a MAC ACL 27
Removing a MAC ACL 29
Changing Sequence Numbers in a MAC ACL 29
Applying a MAC ACL as a Port ACL 30
Adding a MAC ACL to a Port Profile 32
Verifying MAC ACL Configurations 33
Monitoring MAC ACLs 33

Configuration Examples for MAC ACLs 34

Configuration Example for Creating a MAC ACL for any Protocol 34

CHAPTER 4 **Configuring Cisco TrustSec** 35

Information About Cisco TrustSec 35

Cisco TrustSec Architecture 35

Security Group-Based Access Control 36

SGACLs and SGTs 36

SGACL Policies 38

Determining the Source Security Group 39

Determining the Destination Security Group 39

SGACL Enforcement 40

Cisco TrustSec With SXPv3 40

SXPv3 Subnet Expansion 40

Cisco TrustSec Subnet-SGT Mapping 41

Overview of Cisco TrustSec with SXPv4 41

SXP Node ID 43

Keepalive and Hold-Time Negotiation with SXPv4 44

Bidirectional SXP Support Overview 45

Guidelines and Limitations for SXPv4 45

SXP Version Negotiation 46

Authorization and Policy Acquisition 48

Licensing Requirements for Cisco TrustSec 49

Prerequisites for Cisco TrustSec 49

Guidelines and Limitations for Cisco TrustSec 49

Default Settings 50

Configuring Cisco TrustSec 50

Enabling the Cisco TrustSec Feature 50

Configuring Cisco TrustSec Device Credentials 52

Enabling Cisco TrustSec SXP 53

Configuring Cisco TrustSec Device Tracking 54

Configuring a Default SXP Password 55

Configuring a Default SXP Source IPv4 Address 56

Configuring Cisco TrustSec SXP Peer Connections 58

Configuring SXPv4 59

Configuring the Node ID of a Network Device 59

Configuring the Hold-Time for the SXPv4 Protocol on a Network Device 60

Configuring the Hold-Time for the SXPv4 Protocol for Each Connection 62

Configuring Bidirectional SXP Support 64

Verifying Cisco TrustSec with SXPv4 65

Configuring Static IP-SGT Bindings 65

Changing the SXP Retry Period 67

Changing the Interface Delete Hold Timer 68

Configuring AAA on the Cisco TrustSec Cisco NX-OS Devices 69

Configuring Cisco TrustSec Authentication in Manual Mode 71

Configuring SGACL Policies 73

Manually Configuring SGACL Policies 73

Enabling SGACL Policy Enforcement 75

Displaying the Downloaded SGACL Policies 76

Refreshing the Downloaded SGACL Policies 76

Clearing Cisco TrustSec SGACL Policies 77

Enabling Statistics for RBACL 77

Configuring RBACL Logging 78

RBACL Logging 78

RBACL Flows 79

Syslog Messages 80

Configuring RBACL Logging 80

Disabling RBACL Logging 81

Configuring a Time Interval for Accumulating Packet Counters 81

Configuring Flows 81

Configuring Syslog Server Severity Levels 83

Verifying the Cisco TrustSec Configuration 84

Secure Login Enhancements 85

Configuring Login Parameters 85

Configuration Examples for Login Parameters 86

Guidelines and Limitations 87

CHAPTER 1

Overview

This chapter contains the following sections:

- User Accounts, on page 1
- Virtual Service Domain, on page 1
- Authentication, Authorization, and Accounting, on page 1
- RADIUS Security Protocol, on page 2
- TACACS+ Security Protocol, on page 2
- SSH, on page 2
- Telnet, on page 3
- Access Control Lists, on page 3
- Layer 3 Security, on page 3

User Accounts

Access to the Cisco Nexus 1000VE is accomplished by setting up user accounts that define the specific actions permitted by each user. You can create up to 256 user accounts. For each user account, you define a role, user name, password, and expiration date.

Virtual Service Domain

A virtual service domain (VSD) allows you to classify and separate traffic for network services, such as firewalls, traffic monitoring, and those in support of compliance goals.

Authentication, Authorization, and Accounting

Authentication, Authorization, and Accounting (AAA) is an architectural framework for configuring a set of three independent, consistent, and modular security functions

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

- Authorization—Provides the method for remote access control, including one-time authorization or authorization for each service, per-user account list and profile, user group support, and support of IP, IPX, ARA, and Telnet. Remote security servers, such as RADIUS and TACACS+, authorize users for specific rights by associating attribute-value (AV) pairs, which define those rights, with the appropriate user. AAA authorization works by assembling a set of attributes that describe what the user is authorized to perform. These attributes are compared with the information contained in a database for a given user, and the result is returned to AAA to determine the user's actual capabilities and restrictions.
- Accounting—Provides the method for collecting and sending security server information used for billing, auditing, and reporting, such as user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes. Accounting enables you to track the services that users are accessing, as well as the amount of network resources that they are consuming.

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

RADIUS Security Protocol

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

TACACS+ Security Protocol

AAA establishes communication between your network access server and your TACACS+ security server.

TACACS+ is a security application implemented through AAA that provides a centralized validation of users who are attempting to gain access to a router or network access server. TACACS+ services are maintained in a database on a TACACS+ daemon that usually runs on a UNIX or Windows NT workstation. TACACS+ provides separate and modular authentication, authorization, and accounting facilities.

SSH

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

The SSH client works with publicly and commercially available SSH servers.

Telnet

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

Access Control Lists

IP ACLs

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

Layer 3 Security

Layer 3 Security (L3Sec) is a framework that secures the internal control plane communications (control and packet traffic) of the Cisco Nexus 1000VE in a robust manner. Layer 3 Security mode is enabled by default.

CHAPTER 2

Configuring IP ACLs

This chapter contains the following sections:

- Information About ACLs , on page 5
- Prerequisites for IP ACLs, on page 11
- Guidelines and Limitations for IP ACLs, on page 11
- Default Settings for IP ACLs, on page 11
- Configuring IP ACLs, on page 11
- Verifying the IP ACL Configuration, on page 21
- Monitoring IP ACLs, on page 22
- Configuration Example for IP ACL, on page 22

Information About ACLs

An access control list (ACL) is an ordered set of rules that you can use to filter traffic. Each rule specifies a set of conditions that a packet must satisfy to match the rule. When the device determines that an ACL applies to a packet, the device tests the packet against the conditions of all rules. The rule determines whether the packet is to be permitted or denied. If there is no match to any of the specified rules, then the device denies the packet. The device continues processing packets that are permitted and drops packets that are denied.

You can use ACLs to protect networks and specific hosts from unnecessary or unwanted traffic. For example, you can use ACLs to disallow HTTP traffic from a high-security network to the Internet. You can also use ACLs to allow HTTP traffic to a specific site using the IP address of the site to identify it in an IP ACL.

ACL Types and Applications

An ACL is considered a port ACL when you apply it to one of the following:

- Ethernet interface
- vEthernet interface

When a port ACL is applied to a trunk port, the ACL filters traffic on all VLANs on that trunk port. Both IPv4 and IPv6 ACLs are supported.

Active Ports and Services on Nexus 1000V VSM

The following table lists the active ports and services on Nexus 1000V VSM:

Port Number	Protocol	Remark
22	SSH / SCP /SFTP (TCP)	
23	TELNET (TCP)	
123	NTP (UDP)	Used by NTP Sever
161	SNMP (UDP)	Used by SNMP Server

Order of ACL Application

When the device processes a packet, it determines the forwarding path of the packet. The device applies the ACLs in the following order:

1. Ingress port ACL
2. Egress port ACL

Rules

Rules are what you create, modify, and remove when you configure how an access control list (ACL) filters network traffic. Rules appear in the running configuration. When you apply an ACL to an interface or change a rule within an ACL that is already applied to an interface, the supervisor module creates ACL entries from the rules in the running configuration and sends those ACL entries to all the Virtual Supervisor Engines (VSEs).

You can create rules in ACLs in access-list configuration mode by using the **permit** or **deny** command. The device allows traffic that matches the criteria in a permit rule and blocks traffic that matches the criteria in a deny rule. You have many options for configuring the criteria that traffic must meet to match the rule.

Source and Destination

In each rule, you specify the source and the destination of the traffic that matches the rule. You can specify both the source and destination as a specific host, a network or group of hosts, or any host.

How you specify the source and destination depends on whether you are configuring IP or MAC ACLs.

Protocols

ACLs allow you to identify traffic by protocol. You can specify some protocols by name. For example, in an IP ACL, you can specify ICMP by name.

In IP ACLs, you can specify protocols by the integer that represents the Internet protocol number. For example, you can use 115 to specify Layer 2 Tunneling Protocol (L2TP) traffic.

You can specify any protocol by number. In MAC ACLs, you can specify protocols by the EtherType number of the protocol, which is a hexadecimal number. For example, you can use 0x0800 to specify IP traffic in a MAC ACL rule.

Implicit Rules

ACLs have implicit rules, which means that although these rules do not appear in the running configuration, the device applies them to traffic when no other rules in an ACL match. When you configure the device to maintain per-rule statistics for an ACL, the device does not maintain statistics for implicit rules. Implicit rules ensure that unmatched traffic is denied, regardless of the protocol specified in the Layer 2 header of the traffic.

All IPv4 ACLs include the following implicit rule that denies unmatched IP traffic:

```
deny ip any any
```

Additional Filtering Options

You can identify traffic by using additional options. These options differ by ACL type. The following list includes most but not all additional filtering options:

- IP ACLs support the following additional filtering options:
  - Layer 4 protocol
  - TCP and UDP ports
  - ICMP types and codes
  - IGMP types
  - Precedence level
  - Differentiated Services Code Point (DSCP) value
  - TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set

See the *Cisco Nexus 1000VE Command Reference* guide for information about filtering options available when using the applicable permit and deny commands.

Sequence Numbers

The device supports sequence numbers for rules. Every rule that you enter receives a sequence number, either assigned by you or assigned automatically by the device. Sequence numbers simplify the following ACL tasks:

- Adding new rules between existing rules—By specifying the sequence number, you specify where in the ACL a new rule should be positioned. For example, if you need to insert a rule between rules numbered 100 and 110, you could assign a sequence number of 105 to the new rule.
- Removing a rule—Without using a sequence number, removing a rule requires that you enter the whole rule, as follows:

  ```
  switch(config-acl)# no permit tcp 10.0.0.0/8 any
  ```

  However, if the same rule had a sequence number of 101, removing the rule requires only the following command:

  ```
  switch(config-acl)# no 101
  ```

- Moving a rule—With sequence numbers, if you need to move a rule to a different position within an ACL, you can add a second instance of the rule by using the sequence number that positions it correctly, and then you can remove the original instance of the rule. This action allows you to move the rule without disrupting traffic.

If you enter a rule without a sequence number, the device adds the rule to the end of the ACL and assigns a sequence number that is 10 greater than the sequence number of the preceding rule to the rule. For example, if the last rule in an ACL has a sequence number of 225 and you add a rule without a sequence number, the device assigns the sequence number 235 to the new rule.

In addition, you can reassign sequence numbers to rules in an ACL. Resequencing is useful when an ACL has rules numbered contiguously, such as 100 and 101, and you need to insert one or more rules between those rules.

Statistics

The device can maintain global statistics for each rule that you configure. If an ACL is applied to multiple interfaces, the maintained rule statistics are the sum of packet matches (hits) on all the interfaces on which that ACL is applied.

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

For each ACL that you configure, you can specify whether the device maintains statistics for that ACL, which allows you to turn ACL statistics on or off as needed to monitor traffic filtered by an ACL or to help troubleshoot the configuration of an ACL.

The device does not maintain statistics for implicit rules in an ACL. For example, the device does not maintain a count of packets that match the implicit **deny ip any any** rule at the end of all IPv4 ACLs. If you want to maintain statistics for implicit rules, you must explicitly configure the ACL with rules that are identical to the implicit rules.

ACL Logging

You can use ACL logging to monitor flows that affect specific ACLs. The ACLs can be configured with the optional log keyword in each of the access control entries (ACEs). When you configure an option, statistics for each flow that match the ACL permit or deny conditions that you enter are logged in the software.

This example shows how to apply the log option to any IPv4 ACL:

```
switch(config)# ip access-list [name]
switch(config-acl)# permit tcp any 156.10.3.44/24 log
```

This example shows how to apply the log option to any IPv6 ACL:

```
switch(config)# ip access-list [name]
switch(config-acl)# permit tcp any 2001:0db8:85a3::/48 log
```

You can enable logging per rule(s) within the ACL. An implicit deny rule is the default action for ACLs. To log any packets that match the implicit deny rule, you must create an explicit deny rule and add the **log** keyword.

Note

- ACL logging is applicable only to IP ACLs that are configured with the **ip access-list** or **ipv6 access-list** commands. MAC ACL logging is not supported. Other traffic, such as the Virtual Supervisor Module (VSM) management interface or the selectors (aaa authen match, qos match, and so on), are not logged.
- ACL logging does not use the VSM management IP address. When in Layer 3 mode, ACL logging uses the Layer 3 vmk IP address. When in Layer 2 mode, ACL logging uses the vmk0 IP address.

Statistics and logging are provided for each flow. A flow is defined by the following IP flows:

- VSM ID
- Virtual Services Engine (VSE) ID
- Source interface
- Protocol
- Source IP address
- Source port
- Destination IP address
- Destination port

Scalability is provided through the following functionality:

- Each Cisco Nexus 1000VE switch can support up to 64 VSEs.
- Each VSE can support up to 5000 permits and 5000 denies flows. The maximum number of permit/deny flows is a configurable option.
- The flow reporting interval can be set from 5 up to 86,400 seconds (1 day).
- The configuration flow syslog level can be from 0 to 7.
- Up to three syslog servers are supported.

ACL Flows

An ACL flow as it pertains to ACL logging has the following characteristics:

- It represents a stream of IPv4/IPv6 packets with the same packet headers (SrcIP, DstIP, Protocol, SrcPort, DstPort) for which an identical ACL action is enforced. Each flow entry tracks the count of packets that match the flow.
- It is created only if logging is enabled on the corresponding ingress/egress ACL policy. Ingress and egress flows are tracked separately.
- Each VSE tracks a maximum of 10,000 ACL flows; a flow space is shared between permit/deny flows, and each has a configurable maximum of 5000.
- Each flow entry contains the following:
    - Packet tuple

    - ACL action
    - Direction
    - Packet count

- The ACL flow life cycle is as follows:
    - A flow is created when the first packet of a unidirectional stream matches a Layer 3 ACL policy. A new flow notification is sent to the syslog server.
    - For all subsequent packets with a tuple that matches the flow tuple, the per flow packet counter is incremented.
    - Each flow is tracked periodically based on the configured reporting interval. Within each periodic report, all the active flows and the corresponding packet count seen since the last periodic report are reported to the syslog server
    - If no packets match a flow for one full periodic interval, the flow entry is purged. This process is the only flow-aging scheme.
    - A flow is not stateful. There is no connection tracking for TCP flows.
- The flow reporting process occurs in the following manner:
    - For each flow created, a new flow notification message is sent to the syslog server.
    - A periodic report for each active flow comes next. A flow is active if packets that match the flow are seen since the last periodic report.
    - The flow information is exported to the syslog server and contains the following: packet tuple, ACL action, direction, VSE-ID, VSM-ID, packet count.
    - The periodic time can be as low as 5 seconds with the default setting of 5 minutes. A new user space ACL-logging thread handles the periodic poll and report functionality.
    - Syslog messages that identify the flow space usage are sent at 75 percent, 90 percent, and 100 percent of the threshold maximum to the syslog server once during each interval.

Syslog Messages

Syslog message characteristics are as follows:

- Syslog messages that contain flow information are exported from each Virtual Service Engine (VSE).
- The syslog client functionality is RFC-5424 compliant and communicates to servers over a UDP port (514).
- Any host that contains a VSE must be configured with a vmknic interface that can reach the remote syslog server.
- On a ESXi-6.x host, syslog messages are blocked by a firewall. The Cisco Nexus 1000VE has installation scripts that open the firewall for port 514.

Prerequisites for IP ACLs

- You must be familiar with IP addressing and protocols to configure IP ACLs.
- You must be familiar with the interface types that you want to configure with ACLs.

Guidelines and Limitations for IP ACLs

ACLs are not supported in port channels.

Default Settings for IP ACLs

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

Configuring IP ACLs

Creating an IP ACL

You can create an IPv4 or IPv6 ACL on the device and add rules to it.

Before you begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **{ip | ipv6} access-list** *name*
3. switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*
4. (Optional) switch(config-acl)# **statistics per-entry**
5. (Optional) switch(config-acl)# **show** {**ip**|**ipv6**} **access-lists** *name*
6. (Optional) switch(config-acl)# **copy running-config startup-config**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	switch(config)# **{ip | ipv6} access-list** *name*	Creates the named IP ACL (up to 64 characters) and enters IP ACL configuration mode.
Step 3	switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*	Creates a rule in the IP ACL. You can create many rules. The sequence-number argument can be a whole number from 1 to 4294967295. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000VE Command Reference*.
Step 4	(Optional) switch(config-acl)# **statistics per-entry**	Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	(Optional) switch(config-acl)# **show** {**ip**|**ipv6**} **access-lists** *name*	Displays the IP ACL configuration.
Step 6	(Optional) switch(config-acl)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create an IPv4 ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# statistics per-entry
switch(config-acl)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        10 permit ip 192.168.2.0/24 any
switch(config-acl)# copy running-config startup-config
```

This example shows how to create an IPv6 ACL:

```
switch# configure terminal
switch(config)# ipv6 access-list acl-01-ipv6
switch(config-ipv6-acl)# permit tcp 2001:0db8:85a3::/48 2001:0db8:be03:2112::/64
```

Changing an IP ACL

You can add and remove rules in an existing IP ACL. You cannot change existing rules. Instead, to change a rule, you can remove it and create it again with the desired changes.

If you need to add more rules between existing rules than the current sequence numbering allows, you can use the **resequence** command to reassign sequence numbers.

Before you begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# {**ip**|**ipv6**} **access-list** *name*
3. (Optional) switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*
4. (Optional) switch(config-acl)# **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination}*
5. (Optional) switch(config-acl)# [**no**] **statistics per-entry**
6. (Optional) switch(config-acl)# **show ip access-lists** *name*
7. (Optional) switch(config-acl)# **copy running-config startup-config**
8. (Optional) switch(config-acl)# **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# {**ip**|**ipv6**} **access-list** *name*	Places you in IP ACL configuration mode for the specified ACL.
Step 3	(Optional) switch(config-acl)# [*sequence-number*] {**permit** | **deny**} *protocol source destination*	Creates a rule in the IP ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The *sequence-number* argument can be a whole number from 1 to 4294967295. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000VE Command Reference*.
Step 4	(Optional) switch(config-acl)# **no** {*sequence-number* | {**permit** | **deny**} *protocol source destination}*	Removes the rule that you specified from the IP ACL. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000VE Command Reference*.
Step 5	(Optional) switch(config-acl)# [**no**] **statistics per-entry**	Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 6	(Optional) switch(config-acl)# **show ip access-lists** *name*	Displays the IP ACL configuration.
Step 7	(Optional) switch(config-acl)# **copy running-config startup-config**	Copies the running configuration to the startup configuration
Step 8	(Optional) switch(config-acl)# **exit**	Exit the ACL configuration mode for the new rules to take effect. **Note** If you are not executing this step, then wait for 5 seconds for the new rules to take effect.

Example

This example shows how to change an IP ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# statistics per-entry
switch(config-acl)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        10 permit ip 192.168.2.0/24 any
switch(config-acl)# ip access-list acl-01
switch(config-acl)# no 10
switch(config-acl)# no statistics per-entry
switch(config-acl)# show ip access-lists acl-01

IPV4 ACL acl-01
switch(config-acl)# copy running-config startup-config
```

Removing an IP ACL

Before you remove an IP ACL from the switch, ensure that you know whether the ACL is applied to an interface. The switch allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where you have applied the ACL. Instead, the switch considers the removed ACL to be empty, that is, empty ACL with implicit rule of deny IP any .Use the **show ip access-lists** command with the summary keyword to find the interfaces that an IP ACL is configured on.

Before you begin

- Log in to the CLI in EXEC mode
- Know whether the ACL is applied to an interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no** {**ip**|**ipv6**} **access-list** *name*
3. (Optional) switch(config)# **show** {**ip**|**ipv6**} **access-list** *name* **summary**
4. switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no** {**ip**|**ipv6**} **access-list** *name*	Removes the IP ACL that you specified by name from the running configuration.
Step 3	(Optional) switch(config)# **show** {**ip**|**ipv6**} **access-list** *name* **summary**	Displays the IP ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.

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

Example

This example shows how to remove an IP ACL:

```
switch# configure terminal
switch(config)# no ip access-list acl-01
switch(config)# show ip access-lists acl-01 summary
switch(config)# copy running-config startup-config
```

Changing Sequence Numbers in an IP ACL

You can change all the sequence numbers assigned to the rules in an IP ACL.

Before you begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **resequence ip access-list** *name starting-sequence-number increment*
3. switch(config)# **show ip access-lists** *name*
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **resequence ip access-list** *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the starting sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment that you specify. The *starting-sequence-number* argument and the increment argument can be a whole number from 1 to 4294967295.
Step 3	switch(config)# **show ip access-lists** *name*	Displays the IP ACL configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to change sequence numbers in an IP ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        10 permit ip 192.168.2.0/24 any
        20 permit ip 192.168.5.0/24 any
switch(config)# resequence ip access-list acl-01 100 10
switch(config)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        100 permit ip 192.168.2.0/24 any
        110 permit ip 192.168.5.0/24 any
switch(config)# copy running-config startup-config
```

Adding an IP ACL to a Port Profile

You can add an IPv4 or IPv6 ACL to a port profile.

You must know the following information:

- If you want to create a new port profile, you must know the interface type (Ethernet or vEthernet) and the name you want to give the profile.
- The name of the IP access control list that you want to configure for this port profile.
- The direction of the packet flow for the access list.

Before you begin

- Log in to the CLI in EXEC mode.
- Create the IP ACL to add to this port profile and you know its name.
- If you are using an existing port profile, you have created it and you know its name.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*
3. switch(config-port-prof)# {**ip port** | **ipv6 port**} {**access-group** | **traffic-filter**} *name* { **in** | **out**}
4. (Optional) switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]
5. (Optional) switch(config-port-prof)# **copy running-config startup-config**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*	Enters port profile configuration mode for the named port profile.
Step 3	switch(config-port-prof)# {**ip port** | **ipv6 port**} {**access-group** | **traffic-filter**} *name* { **in** | **out**}	Adds the named IPv4 or IPv6 ACL to the port profile for either inbound or outbound traffic.
Step 4	(Optional) switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	Displays the configuration for verification.
Step 5	(Optional) switch(config-port-prof)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to add an IP ACL to a port profile:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# port-profile type vethernet vm_eth1
switch(config-port-prof)# ip port access-group acl-01 out
switch(config-port-prof)# show port-profile name vm_eth1
port-profile vm_eth1
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  ip port access-group acl-01 out
  no shutdown
 evaluated config attributes:
  ip port access-group acl-01 out
  no shutdown
 assigned interfaces:
 port-group: vm_eth1
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vn-service: no
 port-profile role: none
 port-binding: static

switch(config-port-prof)# copy running-config startup-config
```

Applying an IP ACL to the Management Interface

You can apply an IP ACL to the management interface, mgmt0.

Before you begin

Log in to the CLI in EXEC mode.

Be sure that the ACL that you want to apply exists and that it is configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface mgmt0**
3. switch(config-if)# {**ip**|**ipv6**} **access-group | traffic-filter** *access-list* [**in** | **out**]
4. (Optional) switch(config-if)# **show** {**ip**|**ipv6**} **access-group | traffic-filter** *access-list*
5. switch(config-if)# {**ip**|**ipv6**} **access-list match-local-traffic**
6. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface mgmt0**	Places you into interface configuration mode for the management interface.
Step 3	switch(config-if)# {**ip**|**ipv6**} **access-group | traffic-filter** *access-list* [**in** | **out**]	Applies a specified inbound or outbound IP ACL to the interface.
Step 4	(Optional) switch(config-if)# **show** {**ip**|**ipv6**} **access-group | traffic-filter** *access-list*	Displays the ACL configuration.
Step 5	switch(config-if)# {**ip**|**ipv6**} **access-list match-local-traffic**	The **match-local-traffic** option enables matching for locally-generated traffic. **Note** This global command has to be enabled for ACL rules to take effect when the ACL is applied in the egress direction on the mgmt 0 interface.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to apply an IP ACL to the management interface:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip access-list acl-01
switch(config-acl)# permit tcp any any
switch(config-acl)# show ip access-lists acl-01
IPV4 ACL acl-01
        10 permit tcp any any
switch(config-acl)# interface mgmt 0
switch(config-if)# ip access-group acl-01 out
switch(config-if)# show ip access-lists acl-01 summary
IPV4 ACL acl-01
        Total ACEs Configured:1
        Configured on interfaces:
                mgmt0 -  egress (Router ACL)
        Active on interfaces:
                mgmt0 -  egress (Router ACL)
switch(config-if)# ip access-list match-local-traffic
switch(config)# copy running-config startup-config
```

Configuring ACL Logging

ACL logging is enabled by default on all Virtual Service Engines (VSEs). In addition, the following guidelines apply to ACL logging configuration:

- Any rule can be enabled for logging by adding the **log** keyword.
- Only packets that have a rule with the **log** keyword enabled are logged.

Disabling ACL Logging

You can disable ACL logging on a VSE by entering the following command:

Command	Purpose
[**no**] **logging ip access-list cache module** *vse*	Disables ACL logging on the specified VSE.

Configuring a Time Interval for Accumulating Packet Counters

You can configure the time interval for accumulating packet counters before they are reported to the syslog servers. You enter the time range in seconds from 5 to 86,400 seconds (1 day). The default is 300 seconds (5 minutes).

You can configure the amount of time to accumulate packet counters by entering one of the following commands:

Command	Purpose
logging ip access-list cache interval *secs*	Sets the time interval in seconds to accumulate packet counters before they are reported to the syslog servers, where *secs* is the number of seconds.
[**no**] **logging ip access-list cache interval** *secs*	Reverts the configuration to the default time interval configuration 300 seconds (5 minutes), where *secs* is the number of seconds.

These examples show the time interval syslog message format that is sent periodically when the time interval expires:

```
Jul  4 23:36:02 17.241.17.234 1 1988-01-03T01:24:26.026 10.197.135.202 n1k-acllog -
ACLLOG-PERMIT-FLOW-CREATE VSM ID: 10.197.135.202, VSE ID: 42023F60-CBBB-A530-1D16-8E6C17199203
SGT :1 DGT :741 Source IP: 74.1.0.1, Destination IP: 74.1.0.108
Source Port: 0, Destination Port: 0
Source Interface: Veth148, Protocol: "ICMP"(1), Hit-count = 1
```

Configuring Flows

You can configure the number of deny and permit flows per VSE. The range is from 0 to 5000 flows. The default is 3000. A syslog message is sent when the flow is near the maximum threshold. The first message is sent when the number of flows has reached 75 percent of the maximum threshold and the next message is sent when the number of flows has reached 90 percent of the maximum threshold. The last message is sent when the number of flows reaches the maximum threshold of 100 percent.

Configuring Permit Flows

You can configure permit flows by entering one of the following commands:

Command	Purpose
logging ip access-list cache max-permit-flows *num*	Sets the number of permit flows where *num* is the number of flows.
[**no**] **logging ip access-list cache max-permit-flows**	Reverts the configuration to the default permit flow value 3000.

Configuring Deny Flows

You can configure deny flows by entering one of the following commands:

Command	Purpose
logging ip access-list cache max-deny-flows *num*	Sets the number of deny flows, where *num* is the number of flows.
[**no**] **logging ip access-list cache max-deny-flows**	Reverts the configuration back to the default deny flow value 3000.

Syslog Server Severity Levels

You can configure severity levels of the ACL logging syslog messages for up to three remote syslog servers. The range is from 0 to 7. The default severity level is 6.

Severity Code	Severity Level	Description
0	Emergency	System is unusable
1	Alert	Action must be taken immediately
2	Critical	Critical conditions
3	Error	Error conditions
4	Warning	Warning conditions
5	Notice	Normal but significant condition
6	Informational	informational messages
7	Debug	Debug-level messages

Setting the Severity Level for a Syslog Message

You can set the severity level of a syslog message and the server to which you want the message to be sent by entering one of the following commands:

Command	Purpose
[**no**] **acllog match-log-level** *level*	Sets the severity level at which syslog messages are sent, where *level* is the severity code from 0 to 7. The no acllog match-log-level level command will revert the ACL log level back to the default severity level 6.
[**no**] **logging ip access-list cache max-deny-flows** *number*	Sets the maximum number of deny flows to *number* per module. The **no logging ip access-list cache max-deny-flows** *number* sets the maximum number of deny-flows to default value of 3000.
[**no**] **logging ip access-list cache max-permit-flows** *number*	Set the max-permit-flows to a specified number per module. The **no logging ip access-list cache max-permit-flows** *number* sets the maximum number of permit-flows to default value of 3000.
logging server { *A.B.C.D* | *x:x:x:x:x:x:x:x*}*-level*	Specifies the syslog server on which you want to set a severity level, where *A.B.C.D* is the syslog server IPv4 address and *x:x:x:x:x:x:x:x* is the syslog server IPv6 address. The severity levels are between 0 to 7.

Note For ACL logging to work, ACL Logging level should be equal or less than that of Syslog level.

Verifying the IP ACL Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show running-config aclmgr	Displays the ACL configuration, including the IP ACL configuration and interfaces that IP ACLs are applied to.
show {**ip**|**ipv6**} **access-lists** [*name*]	Displays all IP ACLs or a named IP ACL.
show {**ip**|**ipv6**} **access-lists** [*name*] su**mmary**	Displays a summary of all configured IP ACLs or a named IP ACL.
show running-config interface	Displays the configuration of an interface to which you have applied an ACL.
show logging ip access-list status	Displays the ACL logging configuration for a VSM.

Command	Purpose
vemcmd show acllog config	Displays the VSE ACL logging configuration.

Monitoring IP ACLs

Use one of the following commands for IP ACL monitoring:

Command	Purpose
show {**ip**|**ipv6**} **access-lists**	Displays the IPv4 or IPv6 ACL configuration. If the IP ACL includes the **statistics per-entry** command, the output includes the number of packets that have matched each rule.
clear {**ip**|**ipv6**} **access-list** [*name*] **counters**	Clears statistics for all IPv4 or IPv6 ACLs or for a specific IPv4 or IPv6 ACL.

Configuration Example for IP ACL

This example shows how to create an IPv4 ACL named acl-01 and apply it as a port ACL on physical ethernet interface which is not a member of port-channel and configuration verification with match counters:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# ip access-list acl-01
switch(config-acl)# permit ip 192.168.2.0/24 any
switch(config-acl)# permit ip 192.168.5.0/24 any
switch(config-acl)# permit 22 any 10.105.225.225/27
switch(config-acl)# permit ip any 10.105.225.225/27
switch(config-acl)# statistics per-entry
switch(config-acl)# interface ethernet 3/5
switch(config-if)# ip port access-group acl-01 in
switch(config-if)# show ip access-lists acl-01 summary
IPV4 ACL acl-01
        statistics per-entry
        Total ACEs Configured:4
        Configured on interfaces:
                Ethernet3/5 - ingress (Port ACL)
        Active on interfaces:
                Ethernet3/5 - ingress (Port ACL)
switch(config-if)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        100 permit ip 192.168.2.0/24 any [match=0]
        110 permit ip 192.168.5.0/24 any [match=0]
        120 permit 22 any 10.105.225.225/27 [match=0]
        130 permit ip any 10.105.225.225/27 [match=44]
switch(config-if)# clear ip access-list counters acl-01
switch(config-if)# show ip access-lists acl-01
IPV4 ACL acl-01
        statistics per-entry
        100 permit ip 192.168.2.0/24 any [match=0]
        110 permit ip 192.168.5.0/24 any [match=0]
        120 permit 22 any 10.105.225.225/27 [match=0]
        130 permit ip any 10.105.225.225/27 [match=0]
switch(config-if)#
```

This example shows how to enable access list matching for locally generated traffic:

```
switch# ip access-list match-local-traffic
```

This example shows how to verify VSM ACL logging configuration:

```
switch# show logging ip access-list status
Max deny flows = 3000
Max permit flows = 3000
Alert interval = 300
Match log level = 6
VSM IP = 192.168.1.1
Syslog IP = 10.1.1.1
Syslog IP = 0.0.0.0
Syslog IP = 0.0.0.0
ACL Logging enabled on module(s):
4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
ACL Logging disabled on module(s):
3
```

This example shows how to verify VSE ACL logging configuration:

```
cisco-vse:~# vemcmd show acllog config
ACL-Log Config:
Status: enabled
Reporting Interval: 300
Max Permit Flows: 3000
Max Deny Flows: 3000
Syslog Facility : 4
Syslog Severity: 6
Syslog Srvr 1: 10.1.1.1
Syslog Srvr 2: 0.0.0.0
Syslog Srvr 3: 0.0.0.0
VSM: 192.168.1.1
```

CHAPTER 3

Configuring MAC ACLs

This chapter contains the following sections:

- Information About MAC ACLs, on page 25
- Prerequisites for MAC ACLs, on page 25
- Guidelines and Limitations for MAC ACLs, on page 25
- Default Settings for MAC ACLs, on page 25
- Configuring MAC ACLs, on page 26
- Verifying MAC ACL Configurations, on page 33
- Monitoring MAC ACLs, on page 33
- Configuration Examples for MAC ACLs, on page 34

Information About MAC ACLs

MAC access control lists (ACLs) are ACLs that filter traffic using information in the Layer 2 header of each packet.

Prerequisites for MAC ACLs

- You must be familiar with MAC addressing and non-IP protocols to configure MAC ACLs.
- You must be familiar with the ACL concepts presented in this document.

Guidelines and Limitations for MAC ACLs

ACLs are not supported in port channels.

Default Settings for MAC ACLs

Parameters	Default
MAC ACLs	No MAC ACLs exist by default.

Parameters	Default
ACL rules	Implicit rules apply to all ACLs.

Configuring MAC ACLs

Creating a MAC ACL

You can create a MAC ACL and add rules to it. You can also use this procedure to add the ACL to a port profile.

Before you begin

- Log in to the CLI in EXEC mode.
- Have a name to assign to the ACL that you are creating.
- Create a port profile if you want to add the ACL to it.

If you want to also add the ACL to a port profile, you must know the following:

- If you are using an existing port profile, you have already created it and you know its name.
- The interface type (Ethernet or vEthernet) and the name that you want to give the port profile if you are creating a new port profile.
- The direction of packet flow for the access list.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **mac access-list** *name*
3. switch(config-mac-acl)# {**permit** | **deny**} *source destination protocol*
4. (Optional) switch(config-mac-acl)# **statistics per-entry**
5. (Optional) switch(config-mac-acl)# **show mac access-lists** *name*
6. (Optional) switch(config-mac-acl)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac access-list** *name*	Creates the MAC ACL and enters ACL configuration mode.
Step 3	switch(config-mac-acl)# {**permit** | **deny**} *source destination protocol*	Creates a rule in the MAC ACL. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000VE Command Reference*.

	Command or Action	Purpose
Step 4	(Optional) switch(config-mac-acl)# **statistics per-entry**	Specifies that the device maintains global statistics for packets that match the rules in the ACL.
Step 5	(Optional) switch(config-mac-acl)# **show mac access-lists** *name*	Displays the MAC ACL configuration for verification.
Step 6	(Optional) switch(config-mac-acl)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to create a MAC ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config-mac-acl)# statistics per-entry
switch(config-mac-acl)# show mac access-lists acl-mac-01
MAC ACL acl-mac-01
        statistics per-entry
        10 permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config-mac-acl)# copy running-config startup-config
```

Changing a MAC ACL

You can change an existing MAC ACL, for example, to add or remove rules.

Use the **resequence** command to reassign sequence numbers, such as when adding rules between existing sequence numbers.

Before you begin

- Log in to the CLI in EXEC mode.
- In an existing MAC ACL, know that you cannot change existing rules.
- In an existing MAC ACL, know that you can add and remove rules.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **mac access-list** *name*
3. (Optional) switch(config-mac-acl)# [*sequence-number*] {**permit** | **deny**} *source destination protocol*
4. (Optional) switch(config-mac-acl)# **no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*}
5. switch(config-mac-acl)# [**no**] **statistics per-entry**
6. (Optional) switch(config-mac-acl)# **show mac access-lists** *name*
7. switch(config-mac-acl)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac access-list** *name*	Creates the MAC ACL and enters ACL configuration mode.
Step 3	(Optional) switch(config-mac-acl)# [*sequence-number*] {**permit** | **deny**} *source destination protocol*	Creates a rule in the MAC ACL. Using a sequence number allows you to specify a position for the rule in the ACL. Without a sequence number, the rule is added to the end of the rules. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000VE Command Reference*.
Step 4	(Optional) switch(config-mac-acl)# **no** {*sequence-number* | {**permit** | **deny**} *source destination protocol*}	Removes the rule that you specify from the MAC ACL. The **permit** and **deny** keywords support many ways of identifying traffic. For more information, see the *Cisco Nexus 1000VE Command Reference*.
Step 5	switch(config-mac-acl)# [**no**] **statistics per-entry**	Specifies that the device maintains global statistics for packets that match the rules in the ACL. The **no** option stops the device from maintaining global statistics for the ACL.
Step 6	(Optional) switch(config-mac-acl)# **show mac access-lists** *name*	Displays the MAC ACL configuration for verification.
Step 7	switch(config-mac-acl)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to change a MAC ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# show mac access-lists

MAC ACL acl-mac-01
        statistics per-entry
        10 permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config-mac-acl)# no 10
switch(config-mac-acl)# no statistics per-entry
switch(config-mac-acl)# end
switch# show mac access-lists

MAC ACL acl-mac-01
        20 permit f866.f222.e5a6 ffff.ffff.ffff any
switch# copy running-config startup-config
```

Removing a MAC ACL

You can remove a MAC ACL from the switch. Ensure that you know whether the ACL is applied to an interface. The switch allows you to remove ACLs that are currently applied. Removing an ACL does not affect the configuration of interfaces where the ACL is applied. Instead, the switch considers the removed ACL to be empty.

To find the interfaces that a MAC ACL is configured on, use the **show mac access-lists** command with the summary keyword.

Before you begin

- Log in to the CLI in EXEC mode.
- Know whether the ACL is applied to an interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no mac access-list** *name*
3. (Optional) switch(config)# **show mac access-lists** *name* **summary**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no mac access-list** *name*	Removes the specified MAC ACL from the running configuration.
Step 3	(Optional) switch(config)# **show mac access-lists** *name* **summary**	Displays the MAC ACL configuration. If the ACL remains applied to an interface, the command lists the interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to remove a MAC ACL:

```
switch# configure terminal
switch(config)# no mac access-list acl-mac-01
switch(config)# show mac access-lists acl-mac-01 summary
switch(config)# copy running-config startup-config
```

Changing Sequence Numbers in a MAC ACL

You can change sequence numbers assigned to rules in a MAC ACL. Resequencing is useful when you need to insert rules into an ACL and there are not enough available sequence numbers.

Before you begin

Log in to the CLI in EXEC mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **resequence mac access-list** *name starting-sequence-number increment*
3. (Optional) switch(config-mac-acl)# **show mac access-lists** *name*
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **resequence mac access-list** *name starting-sequence-number increment*	Assigns sequence numbers to the rules contained in the ACL, where the first rule receives the number specified by the starting-sequence number that you specify. Each subsequent rule receives a number larger than the preceding rule. The difference in numbers is determined by the increment number that you specify.
Step 3	(Optional) switch(config-mac-acl)# **show mac access-lists** *name*	Displays the MAC ACL configuration for verification.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to change sequence numbers in a MAC ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# show mac access-lists acl-mac-01
MAC ACL acl-mac-01
        10 permit 00c0.4f00.0000 0000.00ff.ffff any
        20 permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config)# resequence mac access-list acl-mac-01 100 10
switch(config)# show mac access-lists acl-mac-01
MAC ACL acl-mac-01
        100 permit 00c0.4f00.0000 0000.00ff.ffff any
        110 permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config)# copy running-config startup-config
```

Applying a MAC ACL as a Port ACL

You can apply a MAC ACL as a port ACL to any of the following interface types:

- Physical Ethernet interfaces
- Virtual Ethernet interfaces

A MAC ACL can also be applied to a port profile that is attached to a physical Ethernet interface or a virtual Ethernet interface.

Note ACLs cannot be applied on a port-channel interface. However, an ACL can be applied on a physical Ethernet interface that is not part of the port channel.

Before you begin

- Log in to the CLI in EXEC mode.
- Know that the ACL that you want to apply exists and is configured to filter traffic in the manner that you need for this application.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** { **ethernet** ¦ **vethernet** } *port*
3. switch(config-if)# **mac port access-group** *access-list* [**in** | **out**]
4. (Optional) switch(config-if)# **show running-config aclmgr**
5. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** { **ethernet** ¦ **vethernet** } *port*	Places you into interface configuration mode for the specified interface.
Step 3	switch(config-if)# **mac port access-group** *access-list* [**in** | **out**]	Applies a MAC ACL to the interface.
Step 4	(Optional) switch(config-if)# **show running-config aclmgr**	Displays the ACL configuration.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to apply a MAC ACL as a port ACL:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface vethernet 1
switch(config-if)# mac port access-group acl-mac-01 in
switch(config-if)# show running-config aclmgr
mac access-list acl-mac-01
  100 permit 00C0.4F00.0000 0000.00FF.FFFF any
  110 permit F866.F222.E5A6 FFFF.FFFF.FFFF any
interface Vethernet1
```

```
  mac port access-group acl-mac-01 in
switch(config-if)# copy running-config startup-config
```

Adding a MAC ACL to a Port Profile

You can add a MAC ACL to a port profile.

Before you begin

- Log in to the CLI in EXEC mode.
- Create the MAC ACL to add to this port profile and know its name.
- If you are using an existing port profile, know its name.
- If you are creating a new port profile, know the interface type (Ethernet or vEthernet) and the name you want to give the profile.
- Know the direction of packet flow for the access list.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*
3. switch(config-port-prof)# **mac port access-group** *name* {**in** | **out**}
4. (Optional) switch(config-port-prof)# **show port-profile name** *profile-name*
5. (Optional) switch(config-port-prof)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** [**type** {**ethernet** | **vethernet**}] *name*	Places you in port profile configuration mode for the named port profile.
Step 3	switch(config-port-prof)# **mac port access-group** *name* {**in** | **out**}	Adds the named ACL to the port profile for either inbound or outbound traffic.
Step 4	(Optional) switch(config-port-prof)# **show port-profile name** *profile-name*	Displays the configuration for verification.
Step 5	(Optional) switch(config-port-prof)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to add a MAC ACL to a port profile:

```
switch# configure termnal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# port-profile vm_eth1
switch(config-port-prof)# mac port access-group acl-mac-01 out
switch(config-port-prof)# show port-profile name vm_eth1
port-profile vm_eth1
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  mac port access-group acl-mac-01 out
  no shutdown
 evaluated config attributes:
  mac port access-group acl-mac-01 out
  no shutdown
 assigned interfaces:
 port-group: vm_eth1
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vn-service: no
 port-profile role: none
 port-binding: static

switch(config-port-prof)# copy running-config startup-config
```

Verifying MAC ACL Configurations

Use one of the following commands to verify the configuration:

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration.
show running-config aclmgr	Displays the ACL configuration, including MAC ACLs and the interfaces that they are applied to.
show running-config interface	Displays the configuration of the interface to which you applied the ACL.
show mac access-lists summary	Displays a summary of all configured MAC ACLs or a named MAC ACLs.

Monitoring MAC ACLs

Use the following commands for MAC ACL monitoring:

Command	Purpose
show mac access-lists	Displays the MAC ACL configuration. If the MAC ACL includes the **statistics per-entry** command, the **show mac access-lists** command output includes the number of packets that have matched each rule.

Command	Purpose
clear mac access-list counters	Clears statistics for all MAC ACLs or for a specific MAC ACL.

Configuration Examples for MAC ACLs

Configuration Example for Creating a MAC ACL for any Protocol

This example shows how to create an MAC ACL named acl-mac-01 and apply it as a port ACL on physical ethernet interface which is not a member of port-channel and configuration verification with match counters.

```
switch(config)# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# mac access-list acl-mac-01
switch(config-mac-acl)# 100 permit 00c0.4f00.0000 0000.00ff.ffff any
switch(config-mac-acl)# 110 permit f866.f222.e5a6 ffff.ffff.ffff any
switch(config-mac-acl)# statistics per-entry
switch(config-mac-acl)# end
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 3/5
switch(config-if)# mac port access-group acl-mac-01 out
switch(config-if)# show mac access-lists acl-mac-01 summary

MAC ACL acl-mac-01
        statistics per-entry
        Total ACEs Configured:2
        Configured on interfaces:
                Ethernet3/5 -  egress (Port ACL)
        Active on interfaces:
                Ethernet3/5 -  egress (Port ACL)
switch(config-if)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
        statistics per-entry
        100 permit 00c0.4f00.0000 0000.00ff.ffff any [match=0]
        110 permit f866.f222.e5a6 ffff.ffff.ffff any [match=546]
switch(config-if)# clear mac access-list counters
switch(config-if)# show mac access-lists acl-mac-01

MAC ACL acl-mac-01
        statistics per-entry
        100 permit 00c0.4f00.0000 0000.00ff.ffff any [match=0]
        110 permit f866.f222.e5a6 ffff.ffff.ffff any [match=0]
switch(config-if)#
```

CHAPTER 4

Configuring Cisco TrustSec

This chapter contains the following sections:

- Information About Cisco TrustSec, on page 35
- Licensing Requirements for Cisco TrustSec, on page 49
- Prerequisites for Cisco TrustSec , on page 49
- Guidelines and Limitations for Cisco TrustSec , on page 49
- Default Settings, on page 50
- Configuring Cisco TrustSec, on page 50
- Configuring RBACL Logging, on page 78
- Verifying the Cisco TrustSec Configuration, on page 84
- Secure Login Enhancements, on page 85

Information About Cisco TrustSec

Cisco TrustSec Architecture

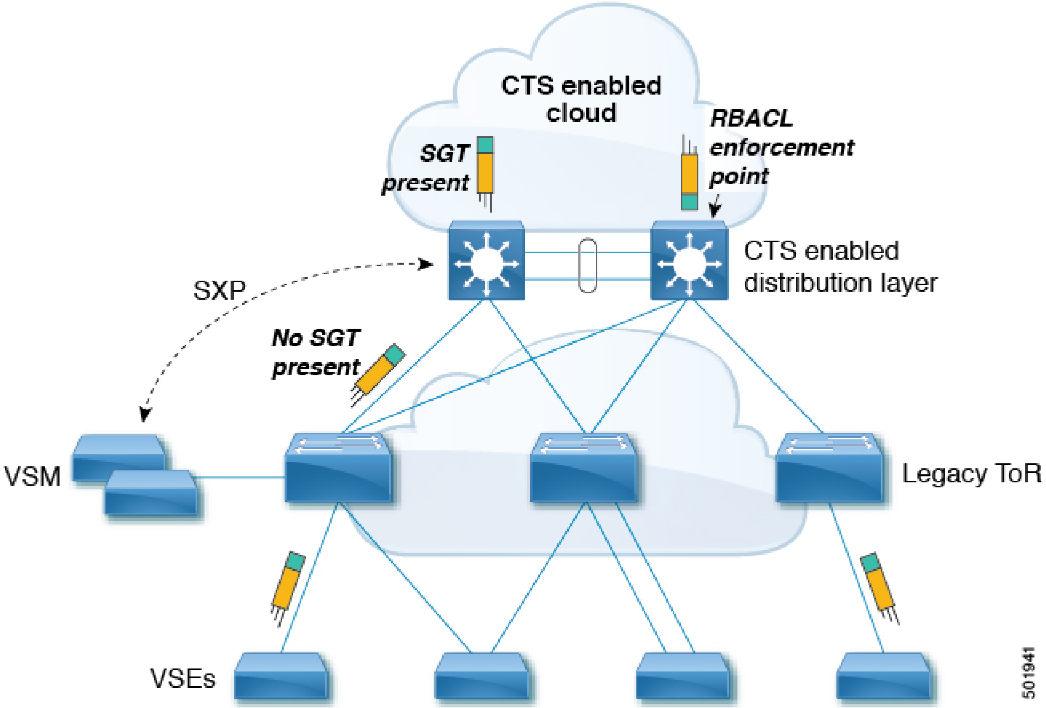
The Cisco TrustSec security architecture enables you to build secure networks by establishing clouds of trusted network devices. Each device in the cloud is authenticated by its neighbors.

Cisco TrustSec uses the device and user identification information that is acquired during authentication to classify or tag packets as they enter the network. These packets are tagged on ingress to the Cisco TrustSec network so that they can be identified for the purpose of applying security and other policy criteria along the data path. The tag, also called the security group tag (SGT), allows the network to enforce the access control policy by enabling the endpoint device to act upon the SGT to filter traffic.

Note Ingress refers to when a packet enters the first Cisco TrustSec-capable device on its path to the destination. Egress refers to when a packet leaves the last Cisco TrustSec-capable device on the path.

This figure shows an example of a Cisco TrustSec cloud.

Figure 1: Cisco TrustSec Network Cloud Example



The Cisco TrustSec architecture consists of the following major components:

- Authentication—Verifies the identity of each device before allowing it to join the Cisco TrustSec network.
- Authorization—Decides the level of access to the Cisco TrustSec network resources that is based on the authenticated identity of the device.
- Access control—Applies access policies on a per-packet basis using the source tags on each packet.
- Secure communication—Provides encryption, integrity, and data-path replay protection for the packets that flow over each link in the Cisco TrustSec network.

Security Group-Based Access Control

SGACLs and SGTs

In security group access lists (SGACLs), you can control the operations that users can perform based on assigned security groups. The grouping of permissions into a role simplifies the management of the security policy. As you add users to the Cisco NX-OS device, you assign one or more security groups and they immediately receive the appropriate permissions. You can modify security groups to introduce new privileges or restrict current permissions.

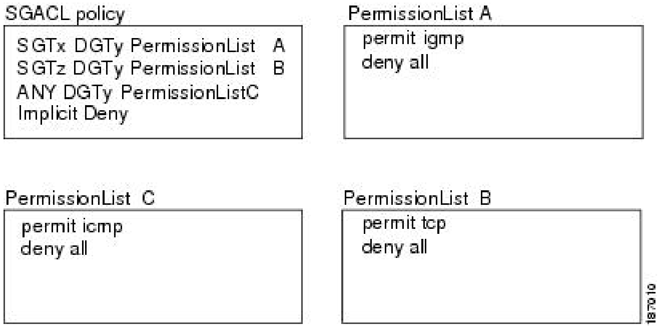
Cisco TrustSec assigns a unique 16-bit tag, called the security group tag (SGT), to a security group. The number of SGTs in the Cisco NX-OS device is limited to the number of authenticated network entities. The SGT is a single label that indicates the privileges of the source within the entire enterprise. Its scope is global within a Cisco TrustSec network.

The management server derives the SGTs based on the security policy configuration. You do not have to configure them manually.

Once authenticated, Cisco TrustSec tags any packet that originates from a device with the SGT that represents the security group to which the device is assigned. The packet carries this SGT throughout the network within the Cisco TrustSec header. Because this tag represents the group of the source, the tag is referred to as the source SGT. At the egress edge of the network, Cisco TrustSec determines the group that is assigned to the packet destination device and applies the access control policy.

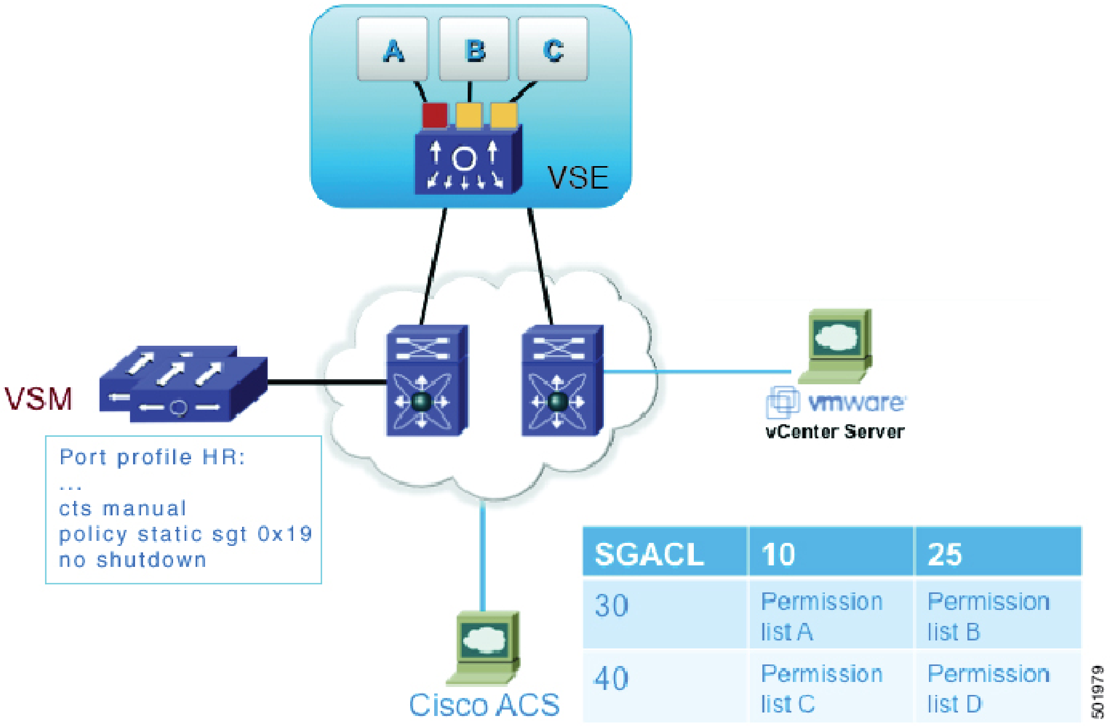
Cisco TrustSec defines access control policies between the security groups. By assigning devices within the network to security groups and applying access control between and within the security groups, Cisco TrustSec achieves access control within the network. The following figure shows an example of an SGACL policy.

Figure 2: SGACL Policy Example



This following figure shows how the SGT assignment and the SGACL enforcement operate in a Cisco TrustSec network.

Figure 3: SGT and SGACL in Cisco TrustSec Network



```
Port profile HR:
...
cts manual
policy static sgt 0x19
no shutdown
```

SGACL	10	25
30	Permission list A	Permission list B
40	Permission list C	Permission list D

The Cisco NX-OS device defines Cisco TrustSec access control policy for a group of devices as opposed to IP addresses in traditional ACLs. With such a decoupling, the network devices are free to move throughout the network and change IP addresses. Entire network topologies can change. As long as the roles and the permissions remain the same, changes to the network do not change the security policy. Cisco TrustSec greatly reduces the size of ACLs and simplifies their maintenance.

In traditional IP networks, the number of access control entries (ACEs) configured is determined as follows:

The number of ACEs = (The number of sources specified) X (The number of destinations specified) X (The number of permissions specified)

Cisco TrustSec uses the following formula:

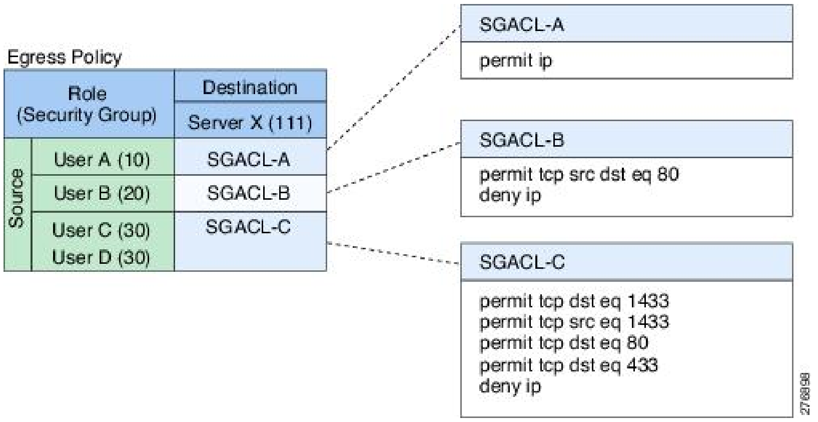
The number of ACEs = The number of permissions specified

SGACL Policies

Using security group access control lists (SGACLs), you can control the operations that users can perform based on the security group assignments of users and destination resources. Policy enforcement within the Cisco TrustSec domain is represented by a permissions matrix, with source security group numbers on one axis and destination security group tags on the other axis. Each cell in the body of the matrix can contain an ordered list of SGACLs which specifies the permissions that should be applied to packets originating from the source security group and destined for the destination security group.

The following figure shows an example of a Cisco TrustSec permissions matrix for a simple domain with three defined user roles and one defined destination resource. Three SGACL policies control access to the destination server based on the role of the user.

Figure 4: SGACL Policy Matrix Example



Egress Policy

	Role (Security Group)	Destination
		Server X (111)
Source	User A (10)	SGACL-A
	User B (20)	SGACL-B
	User C (30) User D (30)	SGACL-C

SGACL-A

```
permit ip
```

SGACL-B

```
permit tcp src dst eq 80
deny ip
```

SGACL-C

```
permit tcp dst eq 1433
permit tcp src eq 1433
permit tcp dst eq 80
permit tcp dst eq 433
deny ip
```

By assigning users and devices within the network to security groups and applying access control between the security groups, Cisco TrustSec achieves role-based topology-independent access control within the network. Because SGACLs define access control policies based on device identities instead of IP addresses as in traditional ACLs, network devices are free to move throughout the network and change IP addresses. As long as the roles and the permissions remain the same, changes to the network topology do not change the security policy. When a user is added to the switch, you simply assign the user to an appropriate security group and the user immediately receives the permissions of that group.

Using role-based permissions greatly reduces the size of ACLs and simplifies their maintenance. With Cisco TrustSec, the number of access control entries (ACEs) configured is determined by the number of permissions specified, resulting in a much smaller number of ACEs than in a traditional IP network.

Determining the Source Security Group

A network device at the ingress of the Cisco TrustSec cloud needs to determine the SGT of the packet that enters the Cisco TrustSec cloud so that it can tag the packet with that SGT when it forwards it into the Cisco TrustSec cloud. The egress network device needs to determine the SGT of the packet so that it can apply the SGACLs.

The network device determines the SGT for a packet in the following order:

1. CTS tag
2. CLI
3. SXP
4. Interface local

Determining the Destination Security Group

The egress network device in a Cisco TrustSec cloud determines the destination group (DGT) for applying the RBACL. This DGT is obtained from the tag that is configured on the egress interface by the interface's port profile.

The network device determines the SGT for a packet in the following order:

1. CLI
2. SXP

3. Interface local

SGACL Enforcement

You configure SGACL enforcement on a port profile. If SGACL enforcement is enabled on the egress interface, the RBACL configured for the (SGT, DGT) pair is applied to the packet. If the packet is dropped, statistics are updated on the ACE. If the SGT is unknown (0), the (*,DGT) policy is applied.

Cisco TrustSec With SXPv3

The Security Group Tag (SGT) Exchange Protocol (SXP) is a control protocol that propagates IP address-SGT binding information across network devices. The Cisco TrustSec supports SXP version 3 (SXPv3) to enable transporting IPv4 subnet to the SGT bindings.

By using the subnet-to-SGT bindings, you can minimize the forward information base (FIB) entries needed for storing the mapping, thereby increasing the scale of TrustSec deployments. In many scenarios, you can use subnet-SGT bindings instead of the L3 interface-SGT.

Note SXPv3 does not support IPv6.

SXPv3 Subnet Expansion

The SXPv3 protocol allows you to configure the expansion limit for a subnet binding. SXP expands a subnet binding to host address bindings when a connection is set up with a peer with a version earlier than Version 3. SXP binding expansion applies only to IPv4 subnet binding.

The characteristics of subnet expansion are as follows:

- When expanding the bindings for overlapping IP addresses with different SGT values, the mapping is obtained from the IP address with the longest prefix length.
- If the subnet expansion reaches the configured limit, a system log is generated for the subnet that cannot be expanded.
- Binding expansion does not expand broadcast IP addresses in a subnet. Also, note that SXP does not summarize host IP addresses to subnet bindings. In the SXP propagation path, if there is a node that does not understand subnet binding, the bindings are expanded and propagated through the rest of the propagation path as the host IP binding, even though there is a node that understands subnet binding.
- The default expansion limit is zero (0) and the maximum allowed expansion limit is 4096. You can set the expansion limit as 0 when you do not have any devices in the network that support a lower version of SXP.

You can use the **cts sxp mapping network-map [***num_bindings***]** command to expand the network limit. The *num_bindings* parameter accept a value from 0 to 4096. The value zero (0) indicates that no expansion is allowed and 4096 is the maximum expansion limit allowed. The default value is zero (0).

Consider an example when the expansion limit is set to 67 and the subnet is /24. Cisco NX-OS expands the first 67 IP addresses for the first subnet SGT known to CTS. Since subnet /24 contains more hosts, it will never be fully expanded, and a syslog is generated.

Note When you set the maximum expansion limit as 4096, Cisco NX-OS supports the mapping of every IP in a /16 subnet. However, you must consider the hardware or software impact of setting the expansion limit to the maximum limit.

Cisco TrustSec Subnet-SGT Mapping

The subnet-SGT mapping binds an SGT to all the host addresses of a specified subnet. After this mapping is completed, Cisco TrustSec imposes SGT on the incoming packets with the source IP address that belongs to the specified subnet. This enables you to enforce the CTS policy on the traffic flowing through data center hosts. You can configure IPv4 subnet-SGT bindings under a VRF instance.

A new attribute, *net-mask*, is added to the **cts role-based sgt map** command to define subnet mapping on the VSM.

In IPv4 networks, SXPv3 and later versions can receive and parse subnet network addresses or prefix strings from SXPv3 peers.

For example, the IPv4 subnet 198.1.1.0/29 is expanded as follows (only three bits for host addresses):

- Host addresses 198.1.1.1 to 198.1.1.7 are tagged and propagated to the SXP peer.
- Network and broadcast addresses 198.1.1.0 and 198.1.1.8 are not tagged and not propagated.

Note Use the **cts sxp mapping network-map** global configuration command to limit the number of subnet binding expansions exported to an SXPv1 peer.

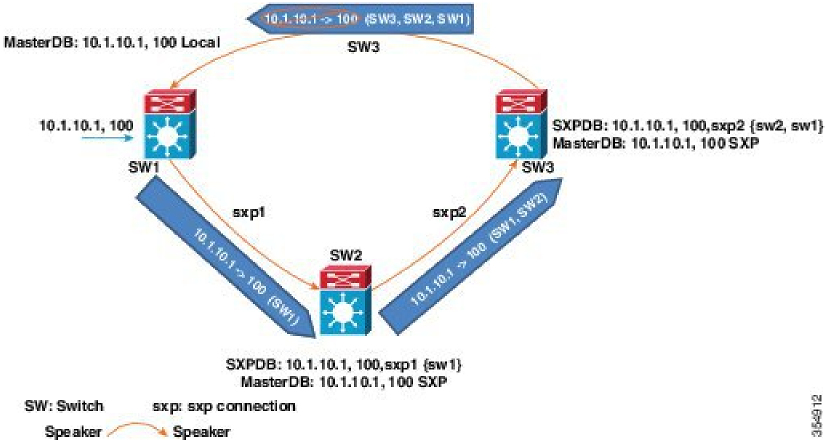
The subnet bindings are static, which means that active hosts are not learned. They can be used locally for SGT imposition and SGACL enforcement. Packets tagged by subnet-SGT mapping can be propagated on Layer 2 or Layer 3 TrustSec links. Additionally, you can use the **cts sxp allow default-route-sgt** command to enable the transport of SGT bindings through the default route, that is, unknown IP address 0.0.0.0.

Overview of Cisco TrustSec with SXPv4

CTS SXP version 4 (SXPv4) enhances the functionality of SXP by adding a loop detection mechanism to prevent stale binding in the network. SXP connections can be enabled such that the binding forwarded by one switch for an SXP connection can be received from another SXP connection, resulting in SXP connection loops. SXP loop topology might, however, result in stale binding in the network. SXPv4' s built-in loop detection and prevention mechanism addresses the stale binding issue whenever there is a loop between SXP nodes.

Loop prevention is achieved by adding SXP propagation path information when propagating (adding/deleting) bindings. Propagation path information keeps track of the network devices (via their node IDs) that the binding travels in an ordered manner. All nodes that participate in the network with looped SXP connections must run SXPv4 to function correctly. Loop detection is a mandatory capability in SXPv4.

Figure 5: SXPv4 Loop Detection


10.1.10.1 -> 100 (SW3, SW2, SW1)
MasterDB: 10.1.10.1, 100 Local
SW3
10.1.10.1, 100
SW1
SXPDB: 10.1.10.1, 100,sxp2 {sw2, sw1}
MasterDB: 10.1.10.1, 100 SXP
SW3
sxp1
sxp2
10.1.10.1 -> 100 (SW1)
SW2
10.1.10.1 -> 100 (SW1, SW2)
SXPDB: 10.1.10.1, 100,sxp1 {sw1}
MasterDB: 10.1.10.1, 100 SXP
SW: Switch
sxp: sxp connection
Speaker
Speaker
354912

In the figure above there are three network devices: SW1, SW2, and SW3. There are also three SXP connections: SXP1, SXP2 and SXP3, together which create an SXP connection loop. A binding (10.1.10.1, 100) is learned at SW1 through local authentication. The binding is exported by SW1 to SW2 together with the path information (that is, SW1, from where the binding is forwarded).

Upon receiving the binding, SW2 exports it to SW3, again prepending the path information (SW2, SW1). Similarly, SW3 forwards the binding to SW1 with path information SW3, SW2, SW1. When SW1 receives the binding, the path information is checked. If its own path attribute is in the binding update received, then a propagation loop is detected. This binding is dropped and not stored in the SXP binding database.

If the binding is removed from SW1, (for example, if a user logs off), a binding deletion event is sent. The deletion event goes through the same path as above. When it reaches SW1, no action will be taken as no such binding exists in the SW1 binding database.

Loop detection is done when a binding is received by an SXP but before it is added to the binding database.

The commonly used SXPv4 terms are:

- **SXP Node ID**: SXP Node ID is a 32 bit identifier that is either self-assigned by the switch or router, or can be configured by the user. It is important for the loop detection/prevention functionality.
- **SXP Default Node ID**: If a SXP Node ID is not configured by the user, when SXP is enabled and before establishing a connection, the switch or router has the capability to self-assign the SXP Node ID identifier. For Nexus 1000VE, the IP address configured for mgmt0 interface is configured as the default Node ID.
- **SXP Peer sequence**: Sequence of node IDs of the devices through which the IP-SGT binding has traversed in order to reach the listener, with the node ID of the immediate speaker at the head of the list. The peer sequence information is necessary for the accurate loop prevention. The listener discards bindings with its own node ID in the sequence information.
- **SXP Keep-alive mechanism**: In-built keep-alive handshake mechanism between speaker and listener in order to allow for timely detection of connectivity loss, deletion of connection resources and staling of the IP-SGT bindings. SXPV4 capable devices use TCP Keep-alive over V1 and V3 connections.
- **SXP Speaker Hold Time**: The minimum acceptable hold-time that the speaker allows for a connection (directly related to the minimum interval at which speaker will send out keep-alive messages).

- **SXP Listener Hold Time Range**: The hold time range the listener requires for a connection (directly related to the minimum and maximum intervals at which listener expects keep-alive messages from the speaker).
- **SXP Connection Negotiated Hold Time**: The negotiated hold time that the speaker and listener agree upon in the open message hand-shakes prior to connection is established.

  Note The listener expects to receive at least one update or keep-alive message within the Listener Hold Time interval on an SXPV4 connection. If the negotiation succeeds, the speaker hold time is lesser than the maximum listener hold time.

- **SXP Capability**: The Nexus 1000VE listeners advertises the following capabilities: IPV4 and Subnet-SGT. Additionally, to support default IP-SGT transport in a mixed network, the default IP-SGT capability is exported to allow the speaker to selectively transport default IP-SGT mapping over SXPV4 connections.
- **IP-SGT (Installed) Database**: The installed IP-SGT database that consists of the final IP-SGT bindings amongst all sources (CLI/SXP, Port-sgt etc) that are selected for local installation and transport over SXP.
- **SXP Contributor Database**: This database contains all the host/subnet SGT bindings learnt from every contributor, along with the subsidiary information useful for loop detection and prevention, Peer Sequence:
  - Time-Stamp/ Counter information
  - Active/Contributor status
  - SGT and Staling Flags.
- **SXP Contributor Logic**: If there are one or more contributors for the same binding learned at a listener, the SXPV4 listener applies the following logic to determine the active/best SXP contributor:
  - **Shortest Path Rule**: Bindings with the shortest peer-sequence length are preferred.
  - **Most Recently Received Rule**: Bindings learnt most-recently are preferred as a tie-breaker.
- **SXP Version Negotiation**: Refer to the SXP Version Negotiation matrix.

SXP Node ID

An SXP node ID is used to identify the individual devices within the network. The node ID is a four-octet integer that can be configured by the user. If it is not configured by the user, Cisco TrustSec assigns the router ID on the default VRF as the node ID, in the same manner that EIGRP generates its router ID, which is the first IP address on Cisco Nexus 1000VE series switches.

The SXP loop detection mechanism drops binding propagation packets based on finding its own node ID in the peer sequence attribute. Changing a node ID in a loop detection-running SXP network could break SXP loop detection functionality and therefore needs to be handled carefully.

The bindings that are associated with the original node ID have to be deleted in all SXP nodes before the new node ID is configured. This can be done by disabling the SXP feature on the network device where you desire to change the node ID. Before you change the node ID, wait until the SXP bindings that are propagated with the particular node ID in the path attribute are deleted.

The node ID configuration is blocked or restricted when SXP is in the enabled state. Router-ID changes in the switch does not affect the SXP node ID, while SXP is enabled. A syslog is generated to indicate that the router ID of the system has changed and this may affect SXP loop detection functionality.

Note Disabling the SXP feature brings down all SXP connections on the device.

Keepalive and Hold-Time Negotiation with SXPv4

SXP uses a TCP-based, keepalive mechanism to determine if a connection is live. SXPv4 adds an optional negotiated keepalive mechanism within the protocol in order to provide more predictable and timely detection of connection loss.

SXP connections are asymmetric with almost all of the protocol messages (except for open/open_resp and error messages) being sent from an SXP speaker to an SXP listener. The SXP listener can keep a potentially large volume of state per connection, which includes all the binding information learned on a connection. Therefore, it is only meaningful to have a keepalive mechanism that allows a listener to detect the loss of connection with a speaker.

The mechanism is based on two timers:

- **Hold timer**: Used by a listener for detection of elapsing time without successive keepalive and/or update messages from a speaker.
- **Keepalive timer**: Used by a speaker to trigger the dispatch of keepalive messages during intervals when no other information is exported via update messages.

The hold-time for the keepalive mechanism may be negotiated during the open/open_resp exchange at connection setup. The following information is important during the negotiation:

- A listener may have desirable range for the hold-time period locally configured or have a default of 90 to 180 seconds. A value of 0xFFFF..0xFFFF indicates that the keepalive mechanism is not used.
- A speaker may have a minimum acceptable hold-time period locally configured or have a default of 120 seconds. This is the shortest period of time a speaker is willing to send keepalive messages for keeping the connection alive. Any shorter hold-time period would require a faster keepalive rate than the rate the speaker is ready to support.
- A value of 0xFFFF implies that the keepalive mechanism is not used.
- The negotiation succeeds when the speaker’s minimum acceptable hold-time falls below or within the desirable hold-time range of the listener. If one end turns off the keepalive mechanism, the other end should also turn it off to make the negotiation successful.
- The negotiation fails when the speaker’s minimum acceptable hold-time is greater than the upper bound of the listener’s hold-time range.
- The selected hold-time period of a successful negotiation is the maximum of the speaker’s minimum acceptable hold-time and the lower bound of the listener’s hold-time range.
- The speaker calculates the keepalive time to one-third of the selected hold-time by default unless a different keepalive time is locally configured.

- Larger Minimum listener hold-time values are recommended on systems with large number of bindings or connections. Also, these values are recommended if there is a requirement to hold the bindings on the listener during network maintenance events.

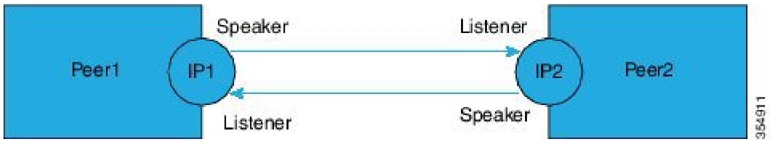
Bidirectional SXP Support Overview

The Bidirectional SXP Support feature enhances the functionality of Cisco TrustSec with SXP version 4 by adding support for Security Group Tag (SGT) Exchange Protocol (SXP) bindings that can be propagated in both directions between a speaker and a listener over a single connection.

With the support for bidirectional Security Group Tag (SGT) Exchange Protocol (SXP) configuration, a peer can act as both a speaker and a listener and propagate SXP bindings in both directions using a single connection.

The bidirectional SXP configuration is managed with one pair of IP addresses, thereby reducing operational complexity. On either end, only the listener initiates the SXP connection and the speaker accepts the incoming connection.

Figure 6: Bidirectional SXP Connection



In addition, Bi-directional SXP uses the underlying loop-detection benefits of SXPV4 to avoid replay of updates back and forth across the same connection.

Note The peers at each end of the connection must be configured as a bidirectional connection using the **both** keyword. It is an incorrect configuration to have one end configured as a bidirectional connection using the **both** keyword and the other end configured as a speaker or listener (unidirectional connection). The system will not be able to detect the mismatch in configuration leading to unpredictable SXP connectivity.

Guidelines and Limitations for SXPv4

Cisco TrustSec SXPv4 has the following guidelines and limitations:

- The Bidirectional SXP Support feature enhances the functionality of Cisco TrustSec with SXP version 4 by adding support for Security Group Tag (SGT) Exchange Protocol (SXP) bindings that can be propagated in both directions between a speaker and a listener over a single connection.
- IPV6 bindings are not learned or transported by the Cisco Nexus 1000VE series switches over SXPV4 connections. However, the SXPv4 peering with speakers transporting IPv6 bindings are still supported.
- Cisco Nexus 1000VE series switches only expands Subnet-SGT bindings over SXPv3 connections.
- After upgrading a switch, the switch advertizes the default SXPv4 version. The appropriate connection versions are re-negotiated with the peers.
- Ensure that there are no overlapping node IDs configured in the network or the node IDs that are configured in the network do not overlap with IP addresses used elsewhere in the network.
- Ensure that there are no overlapping IP addresses to avoid unintentional reuse of default node IDs in the network.

- Prior to modifications to IP addresses in the switch or a router, ensure that the old and the new IP addresses have not been used as default node IDs locally or remotely in the network.
- Ensure that the speaker and listener hold-time values per connection or global or default for each speaker-listener pair are compatible.
- Note that using the hold-time value as 65535 on either speaker or listener disables the in-built keep-alive mechanism and avoids the staling of bindings upon connectivity loss on SXPv4 devices. Administrative connection resets are required to clear these bindings.
- When migrating existing uni-directional connections to bi-directional connections, ensure that the global hold times are compatible and the bindings learnt in both directions are within the supported scale limits. Also, ensure that the global or default hold-time values on speaker and listener are compatible, since you cannot configure hold-time values for these connections on a per-connection basis.

SXP Version Negotiation

The SXP session is established between speaker devices and listener devices. By default, the CTS device advertises the highest supported SXP version. The negotiation is made based on the highest common version supported by the speaker and listener devices. A standalone CTS-supported device can establish an SXP session with different versions, with its peer devices, depending on the SXP versions of the peer devices.

The following table provides information about version negotiation for interoperability in different scenarios.

Table 1: SXP Version Negotiation Cases

Case Number	Speaker	Listener	SXP Session Status
	SXPv1	SXPv1	SXPv1 session is established.
	SXPv1	SXPv2	SXPv1 session is established.
	SXPv1	SXPv3	SXPv1 session is established.
	SXPv1	SXPv4	SXPv1 session is established.
	SXPv1	SXPv4	SXPv1 session is established.
	SXPv2 (Not N1KV)	SXPv1	SXPv1 session is established.
	SXPv2	SXPv2	Not possible because a Cisco Nexus 1000VE switch does not support SXPv2.

Case Number	Speaker	Listener	SXP Session Status
	SXPv2 (Not N1KV)	SXPv3	When the Cisco Nexus 1000VE (SXPv3) Listener receives an OPEN RSP from an SXPv2 speaker: **1.** The Listener generates a system log (syslog), records the Speaker's version, and terminates the session. **2.** The connection is re-established and the Speaker's version is checked: • If the Speaker version is SXPv2, Listener sends OPEN with SXPv1. • If the Speaker version is not SXPv2, Listener sends OPEN with SXPv3. **3.** On receiving an OPEN with SXPv3 response, the Speaker (SXPv2) falls back to SXPv1 and establishes the connection.
	SXPv2 (Not N1KV)	SXPv4	When the Cisco Nexus 1000VE (SXPv4) Listener receives an OPEN RSP from an SXPv2 speaker: **1.** The Listener generates a system log (syslog), records the Speaker's version, and terminates the session. **2.** The connection is re-established and the Speaker's version is checked: • If the Speaker version is SXPv2, Listener sends OPEN with SXPv1. • If the Speaker version is not SXPv2, Listener sends OPEN with SXPv4. **3.** On receiving an OPEN with SXPv4 response, the Speaker (SXPv2) falls back to SXPv1 and establishes the connection.
	SXPv3	SXPv1	SXPv1 session is established.

Case Number	Speaker	Listener	SXP Session Status
	SXPv3	SXPv2	When the Cisco Nexus 1000VE (SXPv3) Listener receives an OPEN RSP from an SXPv2 speaker: **1.** The Listener generates a system log (syslog), records the Speaker's version, and terminates the session. **2.** The connection is re-established and the Speaker's version is checked: • If the Speaker version is SXPv2, Listener sends OPEN with SXPv1. • If the Speaker version is not SXPv2, Listener sends OPEN with SXPv3. **3.** On receiving an OPEN with SXPv1 response, the Speaker (SXPv2) falls back to SXPv1 and establishes the connection.
	SXPv3	SXPv3	SXPv3 session is established.
	SXPv3	SXPv4	SXPv3 session is established.
	SXPv4	SXPv3	SXPv3 session is established.
	SXPv4	SXPv4	SXPv4 session is established.

Authorization and Policy Acquisition

After authentication ends, the supplicant and AT obtain the security policy from the authentication server. The supplicant and AT enforce the policy against each other. Both the supplicant and AT provide the peer device ID that each receives after authentication. If the peer device ID is not available, Cisco TrustSec can use a manually configured peer device ID.

The authentication server returns the following policy attributes:

Cisco TrustSec Trust
Indicates whether the neighbor device is to be trusted for the purpose of putting the SGT in the packets.

Peer SGT
Indicates the security group that the peer belongs to. If the peer is not trusted, all packets received from the peer are tagged with the SGT configured on the ingress interface. If enforcement is enabled on this interface, the SGACLs that are associated with the peer SGT are downloaded. If the device does not know if the SGACLs are associated with the peer’s SGT, the device might send a follow-up request to fetch the SGACLs.

Authorization expiry time
Indicates the number of seconds before the policy expires. The Cisco-proprietary attribute-value (AV) pairs indicate the expiration time of an authorization or policy response to a Cisco TrustSec device. A Cisco TrustSec device should refresh its policy and authorization before it times out.

Tip Each Cisco TrustSec device should support some minimal default access policy in case it is not able to contact the authentication server to get an appropriate policy for the peer.

Licensing Requirements for Cisco TrustSec

The following table shows the licensing requirements for this feature:

Feature	**License Requirement**
Cisco TrustSec	This feature requires an Advanced Services License. See the *Cisco Nexus 1000V License Configuration Guide* for more information on the licensing requirements for Cisco Nexus 1000V.

Prerequisites for Cisco TrustSec

- You must install the Advanced Services license.
- You must enable the 802.1X feature.
- You must enable the Cisco TrustSec feature.
- You must enable the Cisco TrustSec SXP.

Guidelines and Limitations for Cisco TrustSec

- ISE policies do not take precedence over the policies configured locally on the VSM. If you want ISE policies to take precedence, you must remove the locally-configured policy.
- Cisco TrustSec supports only IPv4 addressing.
- To assign an SGT to a VM, you must manually configure SGT in the port profile.
- A maximum of 6000 IP-SGT mappings can be learned system-wide in the DVS. This total is for entries learned through DHCP snooping and device tracking of individual VMs by ARP as well as IP traffic inspection.
- A maximum of 10 IP-SGT bindings can be learned from a single virtual Ethernet interface.
- The IP-SGT mappings can be communicated to up to 64 SXP peer devices.
- Cisco TrustSec does not support 802.1x or data encryption.
- Cisco TrustSec does not support SXPv2 specifications.
- The number of rules per policy is limited to the number of ACL policies that are supported by Cisco Nexus 1000VE.

- If you override SGT mapping to a different SGT, removing the override does not revert the mapping to original SGT configuration.
- CTS propage-sgt configuration does not function as expected.
- The CTS interface delete-hold timer is not applicable when the port-profile is shut down. The IPSGT entry exists till the expiry of the keep-alive default timer which runs on VSE, and then the IPSGT entry is removed from the VSM which takes more than 80 secs.

Default Settings

Table 2: Default Cisco TrustSec Settings

Parameters	Default
Cisco TrustSec	Disabled
SXP	Disabled
SXP default password	None
SXP reconcile period	120 seconds
SXP retry period	60 seconds
Device tracking	Enabled
Interface delete hold timer	60 seconds

Configuring Cisco TrustSec

Enabling the Cisco TrustSec Feature

You must enable the 802.1X feature and the Cisco TrustSec feature on the Cisco Nexus 1000V before you can configure Cisco TrustSec.

Note You cannot disable the 802.1X feature after you enable the Cisco TrustSec feature.

Before you begin

- Log in to the CLI in EXEC mode.
- Ensure that you have installed the Advanced Services license.

SUMMARY STEPS

1. switch# **configure terminal**
2. **feature dot1x**
3. switch(config)# **[no] feature cts**
4. (Optional) switch(config)# **show cts**
5. (Optional) switch(config)# **show feature**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	**feature dot1x** **Example:** `switch(config)# feature dot1x`	Enables the 802.1X feature.
Step 3	switch(config)# **[no] feature cts**	Enables (or disables when you use the **no** form) the Cisco TrustSec feature.
Step 4	(Optional) switch(config)# **show cts**	Displays the Cisco TrustSec configuration.
Step 5	(Optional) switch(config)# **show feature**	Displays the enabled status for features.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable the Cisco TrustSec feature:

```
switch# configure terminal
switch(config)# feature cts
switch(config)# show cts
CTS Global Configuration
==============================
CTS support : enabled
CTS device identity : not configured
SGT : 0
CTS caching support : disabled

Number of CTS interfaces in
DOT1X mode : 0
Manual mode : 0
switch(config)#

switch(config)# show feature
Feature Name Instance State
-------------------- -------- --------
cts 1 enabled
dhcp-snooping 1 enabled
http-server 1 enabled
lacp 1 disabled
netflow 1 disabled
```

```
network-segmentation 1 disabled
port-profile-roles 1 disabled
private-vlan 1 disabled
segmentation 1 disabled
sshServer 1 enabled
tacacs 1 disabled
telnetServer 1 enabled
vtracker 1 disabled
switch(config)#
```

Configuring Cisco TrustSec Device Credentials

You must configure unique Cisco TrustSec credentials on each Cisco TrustSec-enabled Cisco NX-OS device in your network. Cisco TrustSec uses the password in the credentials for device authentication.

Note You must also configure the Cisco TrustSec credentials for the Cisco NX-OS device on the Cisco Secure ISE. See the documentation at the following URL:

http://www.cisco.com/c/en/us/support/security/identity-services-engine/tsd-products-support-series-home.html

Before you begin

Ensure that you enabled Cisco TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **cts device-id** *name* **password** *password*
3. **exit**
4. (Optional) **show cts**
5. (Optional) **show cts environment**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**cts device-id** *name* **password** *password*	Configures a unique device ID and password. The *name* argument has a maximum length of 32 characters and is case sensitive.
Step 3	**exit**	Exits global configuration mode.
Step 4	(Optional) **show cts**	Displays the Cisco TrustSec configuration.
Step 5	(Optional) **show cts environment**	Displays the Cisco TrustSec environment data.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure Cisco TrustSec device credentials:

```
switch# configure terminal
switch(config)# cts device-id MyDevice1 password Cisc0321
switch(config)# exit
switch# copy running-config startup-config
```

Enabling Cisco TrustSec SXP

You can enable the Cisco TrustSec SXP on the Cisco Nexus 1000VE.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **cts sxp enable**
3. (Optional) switch(config)# **show cts sxp**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **cts sxp enable**	Enables (or disables when you use the **no** form) the Cisco TrustSec SXP feature. The default is disabled.
Step 3	(Optional) switch(config)# **show cts sxp**	Displays the Cisco TrustSec SXP configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable the Cisco TrustSec SXP:

```
switch# configure terminal
switch(config)# cts sxp enable
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP default password configured
SXP retry timeout:30
SXP reconcile timeout:120
Minimum SXP Version: 1
Maximum SXP Version:3
Network Map expansion limit:2000
Unsupported SXP version(s):2
```

This example shows how to expand the network limit for SXPv3 subnet expansion:

```
switch# configure terminal
switch(config)# cts sxp enable
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP retry timeout:60
SXP reconcile timeout:120
Minimum SXP Version: 1
Maximum SXP Version:3
Network Map expansion limit:0
Unsupported SXP version(s):2
vsm-sxpv3(config)#
vsm-sxpv3(config)# cts sxp mapping network-map 255
vsm-sxpv3(config)# sh cts sxp
CTS SXP Configuration:
SXP enabled
SXP retry timeout:60
SXP reconcile timeout:120
Minimum SXP Version: 1
Maximum SXP Version:3
Network Map expansion limit:255
Unsupported SXP version(s):2
vsm-sxpv3(config)#
```

Configuring Cisco TrustSec Device Tracking

You can configure device tracking to enable VM IP address learning by inspecting the Address Resolution Protocol (ARP) and IP traffic on virtual Ethernet ports.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **cts device tracking**
3. (Optional) switch(config)# **show cts device tracking**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **cts device tracking**	Enables device tracking on Cisco TrustSec. **Note** The Cisco Nexus 1000VE supports tracking of IP addresses from the ARP/IP traffic inspection on the vses and from DHCP snooping. Cisco TrustSec device tracking tracks IP addresses using the ARP/IP traffic inspection on the vses. To enable Cisco TrustSec device tracking to track IP addresses from DHCP snooping, you must also enable the DHCP snooping feature. By default, device tracking is enabled.
Step 3	(Optional) switch(config)# **show cts device tracking**	Displays the Cisco TrustSec device tracking configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure Cisco TrustSec device tracking:

```
switch# configure terminal
switch(config)# cts device tracking
enabled
switch(config)#
```

Configuring a Default SXP Password

By default, SXP uses no password when setting up connections. You can configure a default SXP password for the Cisco NX-OS device.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **cts sxp default password** [**word** | **7**] *password*

3. (Optional) switch(config)# **show cts sxp**
4. (Optional) switch(config)# **show running-config cts**
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **cts sxp default password** [**word** | **7**] *password*	Configures the SXP default password using the following options: • **word**—Specifies an unencrypted default password. • **7**—Specifies an encrypted default password. By default, no SXP password is used.
Step 3	(Optional) switch(config)# **show cts sxp**	Displays the SXP configuration.
Step 4	(Optional) switch(config)# **show running-config cts**	Displays the running configuration for Cisco TrustSec.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the default SXP password:

```
switch# configure terminal
switch(config)# cts sxp default password 7 CiscoPassword
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP default password configured
SXP retry timeout:30
SXP reconcile timeout:120
Minimum SXP Version: 1
Maximum SXP Version:3
Network Map expansion limit:2000
Unsupported SXP version(s):2
```

Configuring a Default SXP Source IPv4 Address

The Cisco NX-OS software uses the default source IPv4 address in all new TCP connections where a source IPv4 address is not specified. The default source IPv4 address must be set to the IPv4 address of the mgmt0 interface. No other source IPv4 address works when configuring an SXP peer connection.

Note There is no effect on existing TCP connections when you configure the default SXP source IPv4 address.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **cts sxp default password** *password*
3. switch(config)# **cts sxp default source-ip** *mgmt0-interface*
4. (Optional) switch(config)# **show cts sxp**
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **cts sxp default password** *password*	Configures the SXP default password.
Step 3	switch(config)# **cts sxp default source-ip** *mgmt0-interface*	Configures the mgmt0 interface as the SXP default source IPv4 address.
Step 4	(Optional) switch(config)# **show cts sxp**	Displays the SXP configuration.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the default SXP source IPv4 address:

```
switch# configure terminal
switch# cts sxp default password xyzexy
switch(config)# cts sxp default source-ip  10.78.1.73
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP default password configured
SXP retry timeout:30
SXP reconcile timeout:120
Minimum SXP Version: 1
Maximum SXP Version:3
Network Map expansion limit:2000
Unsupported SXP version(s):2
switch(config)#
```

Configuring Cisco TrustSec SXP Peer Connections

You must configure the SXP peer connection on both the speaker and the listener devices. When you are using password protection, make sure to use the same password on both the devices.

Note The SXP source IPv4 address must be configured with the mgmt0 IPv4 address for all SXP connections.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **cts sxp connection peer** *peer-ip-address* **source** *source-ip-address*] **password** {[**default**] | [**none** [**required**] *password*} [**mode**{**listener** | **speaker**}] **vrf management**
3. (Optional) switch(config)# **show cts sxp connection**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **cts sxp connection peer** *peer-ip-address* **source** *source-ip-address*] **password** {[**default**] | [**none** [**required**] *password*} [**mode**{**listener** | **speaker**}] **vrf management**	Configures the SXP address connection. • **Source**—Specifies the IPv4 address of the source. The default source is the IPv4 address that you configured using the **cts sxp default source-ip** command. • **Password**—Specifies the password that SXP should use for the connection using the following options: • **Default**—Uses the default SXP password that you configured using the **cts sxp default password** command. • **None**—Does not use a password. • **Required**—Uses the password specified in the command. • **Mode**—Specifies the role of the remote peer device using the following options:

	Command or Action	Purpose
		- **listener**—The Cisco Nexus 1000V acts as the speaker in the connection and the peer is configured as the listener. - **speaker**—The Cisco Nexus 1000V acts as the listener in the connection and the peer is configured as the speaker. • The **vrf management** keywords specify that the Virtual Routing and Forwarding (VRF) to the peer is the management (mgmt0) interface.
Step 3	(Optional) switch(config)# **show cts sxp connection**	Displays the SXP connections and their status.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure Cisco TrustSec peer connections:

```
switch# configure terminal
switch(config)# cts sxp connection peer 1.2.3.4 password none mode listener vrf management
switch(config)# show cts sxp connection

PEER_IP_ADDR VRF PEER_SXP_MODE SELF_SXP_MODE CONNECTION STATE
10.197.130.184 management listener speaker connected
10.197.130.185 management speaker listener connected
```

Configuring SXPv4

Configuring the Node ID of a Network Device

Before you begin

Enable the Cisco TrustSec feature.

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure the node ID of a network device:

switch(config)# **cts sxp node-id** {*sxp-node-id* | **interface** *interface-type* | *ipv4-address*}

Note Use the **no** form of this command to delete a node ID.

Step 3 Exit global configuration modes:

switch(config)# **exit**

Step 4 (Optional) Display the node ID of a network device by using one of the following commands:

switch# **show cts sxp sgt-map**

switch# **show run | include node-id**

switch# **show cts sxp sgt-map detail**

Example: Configuring the Node ID of a Network Device

The following running configuration shows how to configure the node ID of a network device. Replace the placeholders with relevant values for your setup.

```
#Node Id in Hexadecimal format
configure terminal
 cts sxp node-id <0x1-0xffffffff>
 exit

#Node Id in IPv4 address format
configure terminal
 cts sxp node-id <172.16.1.3>
 exit
```

The following example shows how to configure node ID as an interface.

```
switch(config)# cts sxp node-id interface ethernet 1/1
```

Note that the specified interface should have a valid IP configuration. Otherwise, you cannot configure the node ID.

The following example shows how to display the node ID.

```
switch(config)# show cts sxp sgt-map
SXP Node ID(configured):0x00006789

switch(config)# show run | include node-id
cts sxp node-id interface Eth1/1
```

Configuring the Hold-Time for the SXPv4 Protocol on a Network Device

Before you begin

Enable the Cisco TrustSec feature.

SUMMARY STEPS

1. Enter global configuration mode:
2. Configure a minimum and maximum acceptable hold-time period in seconds for the listener device:
3. Configure a minimum acceptable hold-time period in seconds for the speaker device:
4. Exit global configuration modes:
5. (Optional) Display the hold-time configuration value:

DETAILED STEPS

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure a minimum and maximum acceptable hold-time period in seconds for the listener device:

switch(config)# **cts sxp listener hold-time** *minimum-period maximum-period*

The valid range is from 1-65534 seconds. The default hold-time range for a listener is 90-180 seconds.

Note The maximum-period value must be greater than the minimum-period value.

Step 3 Configure a minimum acceptable hold-time period in seconds for the speaker device:

switch(config)# **cts sxp speaker hold-time** *minimum-period*

The valid range is 1-65534. The default hold-time for a speaker is 120 seconds.

Step 4 Exit global configuration modes:

switch(config)# **exit**

Step 5 (Optional) Display the hold-time configuration value:

switch# **show run | grep speaker**

switch# **show run | grep listener**

Example: Configuring the Hold-Time for the SXPv4 Protocol on a Network Device

The following running configuration shows how to configure the hold-time for the SXPv4 protocol on a listener device. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp listener hold-time <100> <200>
 exit
```

The following running configuration shows how to configure the hold-time for the SXPv4 protocol on a speaker device. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp speaker hold-time <100>
 exit
```

The following example shows how to display the hold-time configuration values.

```
switch(config)# show run | grep speaker
cts sxp speaker hold-time 456

switch(config)# show run | grep listener
cts sxp listener hold-time 20 30
```

Configuring the Hold-Time for the SXPv4 Protocol for Each Connection

The peer connection must be configured on both devices. One device is the speaker and the other is the listener. When using password protection, make sure to use the same password on both ends.

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure a minimum and maximum acceptable hold-time period in seconds for the listener device:

switch(config)# **cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **required** *password*} **mode [[both | local {listener | speaker} | peer {listener | speaker} | listener | speaker] hold-time** *minimum-period maximum-period*] [**vrf** *vrf-name*]]

Configures the CTS-SXP peer address connection.

Note A **hold-time** *maximum-period* value is required only when you use the following keywords: **peer speaker** and **local listener**. In other instances, only a **hold-time** *minimum-period* value is required.

The **source** keyword specifies the IPv4 address of the source device. If no address is specified, the connection uses the default source address, if configured, or the address of the port.

The **password** keyword specifies the password that CTS-SXP uses for the connection using the following options:

- **default**—Use the default CTS-SXP password you configured using the **cts sxp default password** command.
- **none**—A password is not used.

The **mode** keyword specifies the role of the remote peer device:

- **both** — The specified mode refers that the device is both the speaker and the listener in the bidirectional SXP connection.
- **local**—The specified mode refers to the local device.
- **peer**—The specified mode refers to the peer device.
- **listener**— Specifies that the peer device is the listener.
- **speaker**— Specifies that the peer device is the speaker.

The **hold-time** keyword allows you to specify the length of the hold-time period for the speaker or listener device. The valid range is from 0-65534 seconds. The value 0 is the global or default hold-time. You can disable the keep-alive mechanism by specifying the maximum hold-time value as 65535. If the **hold-time** option is not specified, the global hold-time value is used. However, if the global hold-time configuration is missing, the default hold-time is used.

Note A **hold-time** *maximum-period* value is required only when you use the following keywords: **peer speaker** and **local listener**. In other instances, only a **hold-time** *minimum-period* value is required.

The optional **vrf** keyword specifies the VRF to the peer. The default is the default VRF.

You cannot use the management (mgmt 0) interface for SXP.

Note The maximum-period value must be greater than or equal to the minimum-period value.

Step 3 Configure a minimum acceptable hold-time period in seconds for the speaker device:

switch(config)# **cts sxp speaker hold-time** *minimum-period*

The valid range is 1-65534. The default hold-time for a speaker is 120 seconds.

Step 4 Exit global configuration mode:

switch(config)# **exit**

Step 5 (Optional) Displays CTS-SXP status and connections:

switch# **show cts sxp {connections | sgt-map}** [**detail**| **vrf** *vrf-name*]

Example: Configuring the Hold-Time for the SXPv4 Protocol for Each Connection

Example: Disabling Keep-Alive Mechanism at Listener and Speaker Devices

The following running configuration shows how to configure the hold-time for the SXPv4 protocol for each connection. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp connection peer <10.20.2.2> password default mode local speaker hold-time <500>
 exit
```

The following example shows how to display the hold-time for the SXPv4 protocol for a connection.

```
switch(config)# show run cts | include connection
cts sxp connection peer 1.2.3.4 source 3.4.5.6 password none mode speaker hold-time 113 314
 vrf default

switch-listener(config)# show cts sxp sgt-map detail
SXP Node ID(generated):0x14141409
IP-SGT Mappings as follows:
IPv4,SGT : <1.34.56.45/32 , 119>
Vrf      :1
Peer IP  :5.1.1.1
Status   : Active
Seq Num  : 3
Peer Seq :0b0b0b0a
IPv4,SGT : <2.3.11.0/28 , 123>
Vrf      :1
Peer IP  :5.1.1.1
Status   : Active
Seq Num  : 3
Peer Seq :0b0b0b0a,0e0e0e01
Total number of IP-SGT Mappings: 2

switch # show cts sxp connection detail

----------------------------------------------
Peer IP           :3.1.1.2
VRF               :default
PEER MODE         :speaker
Connection State  :connected
Version           :4
Node ID           :0x0e0e0e01
Capability        :UNKNOWN
Conn Hold Time    :120 seconds
```

The following example shows how to display the hold-time configuration values.

```
switch(config)# show run | grep speaker
cts sxp speaker hold-time 456

switch(config)# show run | grep listener
cts sxp listener hold-time 20 30
```

The following example shows how to disable keep-alive mechanism at listener and speaker devices by configuring maximum values for hold-time.

```
switch# configure terminal
switch(config)# cts sxp connection peer 1.2.3.4 source 3.4.5.6 password none mode speaker
hold-time 65535 65535 vrf default
switch(config)# exit

switch# configure terminal
switch(config)# cts sxp connection peer 4.5.6.7 source 6.7.8.9 password none mode listener
 hold-time 65535 vrf default
switch(config)# exit
```

Configuring Bidirectional SXP Support

Before you begin

Enable the Cisco TrustSec feature.

Step 1 Enter global configuration mode:

switch# **configure terminal**

Step 2 Configure the Cisco TrustSec SXP peer address connection for a bidirectional SXP configuration:

switch(config)# **cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **required** *password*} **mode both** [**vrf** *vrf-name*]

Note The **both** keyword configures the bidirectional SXP configuration.

Step 3 Exit global configuration mode:

switch(config)# **exit**

Step 4 (Optional) Displays CTS-SXP status and connections:

switch# **show cts sxp {connections | sgt-map}** [**detail**| **vrf** *vrf-name*]

Example: Configuring Bidirectional SXP Support

The following running configuration shows how to configure bidirectional SXP support. Replace the placeholders with relevant values for your setup.

```
configure terminal
 cts sxp connection peer <3.3.3.2> source <3.3.3.1> password <none> mode both vrf <vrf-name>
Warning: The peer should also be configured as both when this peer is configured as both.
```

The following example shows how to display bidirectional SXP configuration details.

```
switch(config)# show run | include connection
cts sxp connection peer 3.3.3.2 source 3.3.3.1 password none mode both vrf management
```

The following example shows the SXP learnt SGT bindings:

```
switch(config)# show cts sxp sgt-map detail
SXP Node ID(generated):0x00000000
IP-SGT Mappings as follows:
Total number of IP-SGT Mappings: 0
```

Verifying Cisco TrustSec with SXPv4

The following table provides information about how to verify SXPv4 configuration details.

Commands	Purpose
show cts sxp sgt-map vrf *vrf-name*	Displays information about SXP connection.
show cts sxp connection	Displays detailed information about SXP connections.
show cts sxp connection detail	Displays SXP connection for the specified VRF.
show cts sxp connection vrf *vrf-name*	Displays IP address to SGT mapping.
show cts sxp sgt-map	Displays SXP learnt SGT bindings in detail.
show cts sxp sgt-map detail	Displays the SGT mapping for the specified VRF.

Configuring Static IP-SGT Bindings

You can define a static binding between an IP host address to a security group tag (SGT). The static IP-SGT bindings are configured in the context of a management VRF.

Note Any Cisco TrustSec configuration must be done only in the management VRF.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **cts role-based sgt-map** *ip-address*| *ip-address/IPv4_Length_Prefixsgt_value*

3. (Optional) switch(config)# **vrf context**
4. (Optional) switch(config)# **show cts role-based sgt-map**
5. (Optional) switch(config)# **show cts ipsgt entries**
6. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **cts role-based sgt-map** *ip-address*| *ip-address/IPv4_Length_Prefixsgt_value*	Configures the static binding between an IP host address to a security group tag (SGT). • *ip-address*—IP address of the host. • *ip-address/IPv4_Length_Prefix*—IP address and subnet prefix of the host. • *sgt*—SGT corresponding to the IP address. The range is from 1 to 65519.
Step 3	(Optional) switch(config)# **vrf context**	Specifies the IP-SGT bindings in a VRF context. The default is the default VRF.
Step 4	(Optional) switch(config)# **show cts role-based sgt-map**	Displays the mapping of the IP address to SGT for Cisco TrustSec.
Step 5	(Optional) switch(config)# **show cts ipsgt entries**	Displays all the IP-SGT and Subnet-SGT bindings.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure static IP-SGT bindings:

```
switch# configure terminal
switch(config)# cts role-based sgt-map 1.1.1.1 100
switch(config)# vrf context management
switch(config-vrf)# cts role-based sgt-map 2.2.2.3 200
switch(config-vrf)# exit
switch(config)# show cts role-based sgt-map

IP ADDRESS SGT VRF/VLAN SGT CONFIGURATION
193.191.0.174 2 vlan:971 Device Tracking
193.191.0.176 2 vlan:971 Device Tracking
193.191.0.180 2 vlan:971 Device Tracking
193.191.0.178 2 vlan:971 Device Tracking
25.0.0.4 4 vlan:972 Device Tracking
25.0.0.3 4 vlan:972 Device Tracking
1.1.1.241 411 management CLI Configured
1.1.1.242 421 management CLI Configured
1.1.1.243 431 management CLI Configured
1.1.1.244 441 management CLI Configured
```

```
1.1.1.245 451 management CLI Configured
1.1.2.49  491 management CLI Configured
1.1.2.50  501 management CLI Configured
1.1.2.51  511 management CLI Configured
1.1.12.46 461 management CLI Configured
1.1.12.47 471 management CLI Configured
1.12.1.48 481 management CLI Configured
2.2.2.2   3   management CLI Configured
25.0.0.3 4 management SXP peer:10.197.130.185
25.0.0.4 4 management SXP peer:10.197.130.185
25.0.0.5 5 management SXP peer:10.197.130.185

switch(config)# show cts ipsgt entries vrf management

Interface SGT IP ADDRESS VRF/VLAN Learnt
-------------- ------ ------------ ---------- ---------

- 3 2.2.2.2 management CLI Configured
- 4 25.0.0.3 management SXP peer: 10.197.130.185
- 4 25.0.0.4 management SXP peer: 10.197.130.185
- 5 25.0.0.5 management SXP peer: 10.197.130.185
```

Note IP-SGT binding can be configured using the CLI or from SXP. Any SGT mapping that is configured from the CLI displays “CLI Configured” under the SGT_CONFIGURATION column.

This example shows how to configure static subnet IP-SGT bindings:

```
switch# configure terminal
switch(config)# cts role-based sgt-map 1.1.1.1 100
switch(config)# vrf context management
switch(config-vrf)#  cts role-based sgt-map 200.200.200.0/24 2000
switch(config-vrf)# exit
switch(config)# show cts role-based sgt-map

IP ADDRESS SGT VRF/VLAN SGT CONFIGURATION
...........................
...........................
200.200.200.0/24    2000           management     CLI Configured
```

Changing the SXP Retry Period

The SXP retry period determines how often the Cisco NX-OS software retries an SXP connection. When an SXP connection is not successfully set up, the Cisco NX-OS software makes a new attempt to set up the connection after the SXP retry period timer expires. The default value is 60 seconds (1 minute). Setting the SXP retry period to 0 seconds disables the timer and retries are not attempted.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.
- You must enable the Cisco TrustSec feature.

- You must install the Advanced Services license.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **cts sxp retry-period** *seconds*
3. (Optional) switch(config)# **show cts sxp**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **cts sxp retry-period** *seconds*	Specifies the SXP retry timer period. The default value is 60 seconds (1 minute). The range is from 0 to 64000.
Step 3	(Optional) switch(config)# **show cts sxp**	Displays the SXP configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the SXP retry period:

```
switch# configure terminal
switch(config)# cts sxp retry-period 60
switch(config)# show cts sxp
CTS SXP Configuration:
SXP enabled
SXP default password configured
SXP retry timeout:30
SXP reconcile timeout:120
Minimum SXP Version: 1
Maximum SXP Version:3
Network Map expansion limit:2000
Unsupported SXP version(s):2
switch(config)#
```

Changing the Interface Delete Hold Timer

The interface delete hold timer period determines how long the interface holds on to the IP-SGT mapping once the interface goes to a nonparticipating state. After the timer expires, the IP-SGT mappings are deleted from the interface and the peers.

Before you begin

- Log in to the CLI in EXEC mode.
- You must enable the Cisco TrustSec SXP.

- You must enable the Cisco TrustSec feature.
- You must install the Advanced Services license.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **cts interface delete-hold** *seconds*
3. (Optional) switch(config)# **show cts interface delete-hold timer**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **cts interface delete-hold** *seconds*	Specifies the delete hold timer period for an interface. The default value is 60 seconds (1 minute). The range is from 0 to 64000. If the timer is set to 0, the IP-SGT mappings are deleted instantly. The **no** form of this command does not start the timer when the interface goes to a nonparticipating state and the IP-SGT entries are then always held on the interface.
Step 3	(Optional) switch(config)# **show cts interface delete-hold timer**	Displays the interface delete hold timer period.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the interface delete hold timer:

```
switch# configure terminal
switch(config)# cts interface delete-hold 60
switch(config)# show cts interface delete-hold timer
60
switch(config)#
```

Configuring AAA on the Cisco TrustSec Cisco NX-OS Devices

This section describes how to configure AAA on the Cisco NX-OS device in your Cisco TrustSec network cloud.

Before you begin

- Obtain the IPv4 address or hostname for the Cisco Secure ACS.

- Ensure that you enabled Cisco TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **radius-server host** {*ipv4-address* | *hostname*} **key** [**0** | **7**] **key** *key-value* **pac authentication accounting**
3. (Optional) **show radius-server**
4. **aaa group server radius** *group-name*
5. **server** {*ipv4-address* | *hostname*}
6. **use-vrf** *vrf-name*
7. **exit**
8. **aaa authorization cts default group** *group-name*
9. **exit**
10. (Optional) **show radius-server groups** [*group-name*]
11. (Optional) **show aaa authorization**
12. (Optional) **show cts pacs**
13. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**radius-server host** {*ipv4-address* | *hostname*} **key** [**0** | **7**] **key** *key-value* **pac authentication accounting**	Configures a RADIUS server host with a key and PAC. The *hostname* argument is alphanumeric, case sensitive, and has a maximum length of 256 characters. The *key* argument is alphanumeric, case sensitive, and has a maximum length of 63 characters. The **0** option indicates that the key is in clear text. The **7** option indicates that the key is encrypted. The default is clear text.
Step 3	(Optional) **show radius-server**	Displays the RADIUS server configuration.
Step 4	**aaa group server radius** *group-name*	Specifies the RADIUS server group and enters RADIUS server group configuration mode.
Step 5	**server** {*ipv4-address* | *hostname*}	Specifies the RADIUS server host address.
Step 6	**use-vrf** *vrf-name*	Specifies the management VRF instance for the AAA server group. **Note** If you use the management VRF instance, no further configuration is necessary for the devices in the network cloud. If you use a different VRF instance, you must configure the devices with that VRF instance.
Step 7	**exit**	Exits RADIUS server group configuration mode.

	Command or Action	Purpose
Step 8	**aaa authorization cts default group** *group-name*	Specifies the RADIUS server groups to use for Cisco TrustSec authorization.
Step 9	**exit**	Exits global configuration mode.
Step 10	(Optional) **show radius-server groups** [*group-name*]	Displays the RADIUS server group configuration.
Step 11	(Optional) **show aaa authorization**	Displays the AAA authorization configuration.
Step 12	(Optional) **show cts pacs**	Displays the Cisco TrustSec PAC information.
Step 13	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure AAA on the Cisco TrustSec Cisco NX-OS devices:

```
switch# configure terminal
switch(config)# radius-server host 10.10.1.1 key L1a0K2s9 pac authentication accounting
switch(config)# aaa group server radius Rad1
switch(config-radius)# server 10.10.1.1
switch(config-radius)# use-vrf management
switch(config-radius)# exit
switch(config)# aaa authentication cts default group Rad1
switch(config)# exit
switch# copy running-config startup-config
```

Configuring Cisco TrustSec Authentication in Manual Mode

You can manually configure Cisco TrustSec on a port profile if your Cisco NX-OS device does not have access to a Cisco Secure ACS. You must manually configure the port profiles on both ends of the connection.

Before you begin

Ensure that you enabled Cisco TrustSec.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-profile** *name*
3. switch(config-port-prof)# **cts manual**
4. (Optional) switch(config-port-prof-cts-manual)# **policy dynamic identity** *peer-name*
5. (Optional) switch(config-port-prof-cts-manual)# **policy static sgt** *tag* [**trusted**]
6. switch(config-port-prof-cts-manual)# **exit**
7. switch(config-port-prof)# **exit**
8. (Optional) switch(config)# **show cts interface all**
9. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** *name*	Enters port profile configuration mode for the named port profile. If the port profile does not already exist, it is created. **Note** To configure Cisco TrustSec SGTs on an interface, enter the interface configuration mode and specify the interface.
Step 3	switch(config-port-prof)# **cts manual**	Enters Cisco TrustSec manual configuration mode. **Note** You cannot enable Cisco TrustSec on interfaces in half-duplex mode.
Step 4	(Optional) switch(config-port-prof-cts-manual)# **policy dynamic identity** *peer-name*	Configures a dynamic authorization policy download. The *peer-name* argument is the Cisco TrustSec device ID for the peer device. The peer name is case sensitive. **Note** Ensure that you have configured the Cisco TrustSec credentials and AAA for Cisco TrustSec. **Note** The **policy dynamic** and **policy static** commands are mutually exclusive. Only one can be applied at a time. To change from one to the other, you must use the **no** form of the command to remove the configuration before configuring the other command.
Step 5	(Optional) switch(config-port-prof-cts-manual)# **policy static sgt** *tag* [**trusted**]	Configures a static authorization policy. The *tag* argument is a hexadecimal value in the format **0x***hhhh*. The range is from 0x2 to 0xffef. The **trusted** keyword indicates that traffic coming on the interface with this SGT should not have its tag overridden. **Note** The **policy dynamic** and **policy static** commands are mutually exclusive. Only one can be applied at a time. To change from one to the other, you must use the **no** form of the command to remove the configuration before configuring the other command.
Step 6	switch(config-port-prof-cts-manual)# **exit**	Exits the current configuration mode.
Step 7	switch(config-port-prof)# **exit**	Exits the current configuration mode.
Step 8	(Optional) switch(config)# **show cts interface all**	Displays the Cisco TrustSec configuration.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure Cisco TrustSec authentication in CTS manual mode:

```
switch# configure terminal
switch(config)# port-profile pp1
switch(config-port-prof)# cts manual
switch(config-port-prof-cts-manual)# policy dynamic identity MyDevice2

switch(config-port-prof-cts-manual)# exit
switch(config-port-prof)# exit
switch(config)# copy running-config startup-config
```

Configuring SGACL Policies

Manually Configuring SGACL Policies

You can manually configure SGACL policies on your Cisco NX-OS device if a Cisco Secure ICE is not available to download the SGACL policy configuration.

Before you begin

Ensure that you have enabled Cisco TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. **cts role-based access-list** *list-name*
3. (Optional) {**deny** | **permit**} **all**
4. (Optional) {**deny** | **permit**} **icmp**
5. (Optional) {**deny** | **permit**} **igmp**
6. (Optional) {**deny** | **permit**} **ip**
7. (Optional) {**deny** | **permit**} **tcp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]
8. {**deny** | **permit**} **udp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]
9. **exit**
10. **cts role-based sgt** {*sgt-value* | **any** | **unknown**} **dgt** {*dgt-value* | **any** | **unknown**} **access-list** *list-name*
11. (Optional) **show cts role-based access-list**
12. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.
Step 2	**cts role-based access-list** *list-name*	Specifies an SGACL and enters role-based access list configuration mode. The *list-name* argument value is

	Command or Action	Purpose
		alphanumeric, case sensitive, and has a maximum length of 32 characters.
Step 3	(Optional) {**deny** | **permit**} **all**	Denies or permits all traffic.
Step 4	(Optional) {**deny** | **permit**} **icmp**	Denies or permits Internet Control Message Protocol (ICMP) traffic.
Step 5	(Optional) {**deny** | **permit**} **igmp**	Denies or permits Internet Group Management Protocol (IGMP) traffic.
Step 6	(Optional) {**deny** | **permit**} **ip**	Denies or permits IP traffic.
Step 7	(Optional) {**deny** | **permit**} **tcp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]	Denies or permits TCP traffic. The default permits all TCP traffic. The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535.
Step 8	{**deny** | **permit**} **udp** [{**dst** | **src**} {{**eq** | **gt** | **lt** | **neq**} *port-number* | **range** *port-number1 port-number2*}]	Denies or permits UDP traffic. The default permits all UDP traffic. The range for the *port-number*, *port-number1*, and *port-number2* arguments is from 0 to 65535.
Step 9	**exit**	Exits role-based access-list configuration mode.
Step 10	**cts role-based sgt** {*sgt-value* | **any** | **unknown**} **dgt** {*dgt-value* | **any** | **unknown**} **access-list** *list-name*	Maps the SGT values to the SGACL. The *sgt-value* and *dgt-value* argument values range from 0 to 65519. **Note** You must create the SGACL before you can map SGTs to it.
Step 11	(Optional) **show cts role-based access-list**	Displays the Cisco TrustSec SGACL configuration.
Step 12	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to configure an SGACL policy:

```
switch# configure terminal
switch(config)# cts role-based access-list MySGACL
switch(config-rbacl)# deny all log
switch(config-rbacl)# permit icmp
switch(config-rbacl)# deny igmp
switch(config-rbacl)# permit ip
switch(config-rbacl)# deny tcp dst eq 100
switch(config-rbacl)# permit udp src eq 1312
switch(config-rbacl)# exit
switch(config)# cts role-based sgt 3 dgt 10 access-list MySGACL
witch(config)# copy running-config startup-config
```

Enabling SGACL Policy Enforcement

If you use SGACLs, you must enable SGACL policy enforcement on the port profiles or interfaces that have Cisco TrustSec enabled.

Before you begin

- Ensure that you enabled Cisco TrustSec.

SUMMARY STEPS

1. **configure terminal**
2. switch(config)# **port-profile** *name*
3. switch(config-port-prof)# **cts manual**
4. switch(config-port-prof-cts-manual)# **role-based enforcement**
5. switch(config-port-prof-cts-manual)# **exit**
6. switch(config-port-prof)# **exit**
7. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** *name*	Enters port profile configuration mode for the named port profile. If the port profile does not already exist, it is created. **Note** To configure Cisco TrustSec SGTs on an interface, enter the interface configuration mode and specify the interface.
Step 3	switch(config-port-prof)# **cts manual**	Enters CTS manual configuration mode.
Step 4	switch(config-port-prof-cts-manual)# **role-based enforcement**	Enables Cisco TrustSec SGACL policy enforcement on the port profile.
Step 5	switch(config-port-prof-cts-manual)# **exit**	Saves the configuration and exits the current configuration mode.
Step 6	switch(config-port-prof)# **exit**	Saves the configuration and exits the current configuration mode.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enable role-based enforcement on a port profile:

```
switch# configure terminal
switch(config-port-prof)# cts manual
switch(config-port-prof-cts-manual)# role-based enforcement
switch(config-port-prof-cts-manual)# exit
switch(config-port-prof)# exit
switch(config)# copy running-config startup-config
```

Displaying the Downloaded SGACL Policies

After you configure the Cisco TrustSec device credentials and AAA, you can verify the Cisco TrustSec SGACL policies downloaded from the Cisco ISE. The Cisco NX-OS software downloads the SGACL policies when it learns of a new SGT through authentication and authorization on an interface.

Before you begin

Ensure that you enabled Cisco TrustSec.

SUMMARY STEPS

1. **show cts role-based access-list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show cts role-based access-list** **Example:** `switch# show cts role-based access-list`	Displays Cisco TrustSec SGACLs, both downloaded from the Cisco ICE and manually configured on the Cisco NX-OS device.

Refreshing the Downloaded SGACL Policies

You can refresh the SGACL policies downloaded to the Cisco NX-OS device by the Cisco ISE.

Before you begin

Ensure that you enabled Cisco TrustSec.

SUMMARY STEPS

1. **cts refresh role-based policy**
2. (Optional) **show cts role-based policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**cts refresh role-based policy** **Example:** `switch# cts refresh policy`	Refreshes the Cisco TrustSec SGACL policies from the Cisco ISE.
Step 2	(Optional) **show cts role-based policy** **Example:** `switch# show cts role-based policy`	Displays the Cisco TrustSec SGACL policies.

Clearing Cisco TrustSec SGACL Policies

You can clear the Cisco TrustSec SGACL policies.

Note The way policies are cleared depends on whether the SGT is static or dynamic. For a static SGT, the SGT is reset to 0 after the flap occurs. For a dynamic SGT, the SGT is downloaded again from the RADIUS server after the flap occurs.

Before you begin

Ensure that you enabled Cisco TrustSec.

SUMMARY STEPS

1. (Optional) **show cts role-based policy**
2. **clear cts policy** {**all** | **sgt** *sgt-value* | **role-based counters**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) **show cts role-based policy** **Example:** `switch# clear cts policy all`	Displays the Cisco TrustSec RBACL policy configuration.

Step 2	**clear cts policy** {**all** | **sgt** *sgt-value* | **role-based counters**}
	Example:
	`switch# clear cts policy all`
		Clears the policies for Cisco TrustSec connection information.

Enabling Statistics for RBACL

You can request a count of the number of packets that match role-based access control list (RBACL) policies. These statistics are collected per ACE.

Note RBACL statistics are lost only when the Cisco NX-OS device reloads or you deliberately clear the statistics.

Before you begin

Ensure that you have enabled Cisco TrustSec.

If you plan to enable RBACL statistics, ensure that you have enabled RBACL policy enforcement on the port profile or interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **cts role-based counters enable**

3. (Optional) switch(config)# **copy running-config startup-config**
4. switch(config)# **exit**
5. (Optional) switch# **show cts role-based counters**
6. (Optional) switch# **clear cts role-based counters**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **cts role-based counters enable**	Enables or disables RBACL statistics. The default is disabled.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Copies the running configuration to the startup configuration.
Step 4	switch(config)# **exit**	Exits global configuration mode.
Step 5	(Optional) switch# **show cts role-based counters**	Displays the configuration status of RBACL statistics and lists statistics for all RBACL policies.
Step 6	(Optional) switch# **clear cts role-based counters**	Clears the RBACL statistics so that all counters are reset to 0.

Example

This example shows how to enable statistics for RBACL:

```
switch# configure terminal
switch(config)# cts role-based counters enable
switch(config)# copy running-config startup-config
```

Configuring RBACL Logging

RBACL Logging

You can use role-based access control list (RBACL) logging to monitor flows that affect specific RBACLs. The RBACLs can be configured with the optional log keyword in each of the access control entries (ACEs). When you configure an option, statistics for each flow that match the RBACL permit or deny conditions that you enter are logged in the software. RBACL logging supports both IPv4 and IPv6 addresses.

This example shows how to apply the log option:

```
switch(config)# cts role-based access-list [name]
switch(config-rbacl)#  permit tcp dst gt 1111 log
```

You can enable logging per rule(s) within the RBACL. An implicit deny rule is the default action for RBACLs. To log any packets that match the implicit deny rule, you must create an explicit deny rule and add the **log** keyword.

Statistics and logging are provided for each flow. A flow has the following fields:

- Virtual Supervisor Module (VSM) ID
- Virtual Supervisor Engine (VSE) ID
- Security Group Tag (SGT)
- Destination Group Tag (DGT)
- Source IP address
- Source port
- Destination IP address
- Destination port
- Source Interface
- Protocol
- Hit Count

Scalability is provided through the following functionality:

- Each Cisco Nexus 1000VE switch can support up to 64 VSEs.
- Each VSE can support up to 5000 permit and 5000 deny flows. The maximum number of permit/deny flows is a configurable option.
- The flow reporting interval can be from 5 to 86,400 seconds (1 day).
- The configuration flow syslog level can be from 0 to 7.
- Up to three syslog servers are supported.

RBACL Flows

An RBACL flow as it pertains to RBACL logging has the following characteristics:

- It represents a stream of IPv4/IPv6 packets with the same packet headers (SrcIP, DstIP, Protocol, SrcPort, DstPort) for which an identical RBACL action is enforced. Each flow entry tracks the count of packets that match the flow.
- It is created only if logging is enabled on the corresponding ingress/egress RBACL policy. Ingress and egress flows are tracked separately.
- Each VSE tracks a maximum of 10,000 ACL flows; a flow space is shared between permit/deny flows, and each has a configurable maximum of 5000.
- Each flow entry contains the following:
    - Packet tuple
    - RBACL action
    - Direction
    - Packet count

- The RBACL flow lifecycle is as follows:
    - A flow is created when the first packet of a unidirectional stream matches a Layer 3 RBACL policy. A new flow notification is sent to the syslog server.
    - For all subsequent packets with a tuple that matches the flow tuple, the per-flow packet counter is incremented.
    - Each flow is tracked periodically based on the configured reporting interval. Within each periodic report, all the active flows and the corresponding packet count seen since the last periodic report are reported to the syslog server.
    - If no packets match a flow for one full periodic interval, the flow entry is purged. This process is the only flow-aging scheme.
    - A flow is not stateful. There is no connection tracking for TCP flows.
- The flow reporting process occurs in the following manner:
    - For each flow created, a new flow notification message is sent to the syslog server.
    - A periodic report for each active flow comes next. A flow is active if packets that match the flow are seen since the last periodic report.
    - The flow information is exported to the syslog server and contains the following: packet tuple, RBACL action, direction, VSE UUID, VSM ID, packet count.
    - The periodic time can be as low as 5 seconds with the default setting of 5 minutes. A new user space RBACL-logging thread handles the periodic poll and report functionality.
    - Syslog messages that identify the flow space usage are sent at 75 percent, 90 percent, and 100 percent of the threshold maximum to the syslog server once during each interval.

Syslog Messages

Syslog message characteristics are as follows:

- Syslog messages that contain flow information are exported from each Virtual Service Engine (VSE).
- The syslog client functionality is RFC-5424 compliant and communicates to servers over a UDP port (514).
- The host must be configured with a vmknic interface that can reach the remote syslog server.

Configuring RBACL Logging

By default, RBACL logging is enabled on all Virtual Service Engines (VSEs). In addition, the following rules apply to RBACL logging configuration:

- Any rule can be enabled for logging by adding the log keyword.
- Only packets that have a rule with the log keyword enabled are logged.

Disabling RBACL Logging

You can disable RBACL logging on a VSE by entering the following command:

Command	Purpose
[**no**] **logging ip access-list cache module** vse	Disables RBACL logging on the specified VSE.

Configuring a Time Interval for Accumulating Packet Counters

You can configure the time interval for accumulating packet counters before they are reported to the syslog servers. You enter the time range in seconds from 5 to 86,400 seconds (1 day). The default is 300 seconds (5 minutes).

You can configure the amount of time to accumulate packet counters by entering one of the following commands:

Command	Purpose
logging ip access-list cache interval *secs*	Sets the time interval in seconds to accumulate packet counters before they are reported to the syslog servers, where *secs* is the number of seconds.
[**no**] **logging ip access-list cache interval** *secs*	Reverts the configuration to the default time interval configuration 300 seconds (5 minutes), where *secs* is the number of seconds.

This example shows the time interval syslog message format that is sent periodically when the time interval expires:

```
Jun 29 16:58:54 172.23.180.5 1 2018-06-29T07:11:27.108 172.23.180.168 n1k-acllog -
ACLLOG-PERMIT-FLOW-INTERVAL VSM ID: 172.23.180.168, VSE ID:
422feff2-360b-0906-e9c1-895960e5b762
SGT :25 DGT :25 Source IP: 0.0.0.0, Destination IP: 255.255.255.255
Source Port: 68, Destination Port: 67
Source Interface: Veth8, Protocol: "UDP"(17), Hit-count = 91
```

Configuring Flows

You can configure the number of deny and permit flows per VSE. The range is from 0 to 5000 flows; the default is 3000. A syslog message is sent when the flow is near the maximum threshold. The first message is sent when the number of flows has reached 75 percent of the maximum threshold and the next message is sent when the number of flows has reached 90 percent of the maximum threshold. The last message is sent when the number of flows reaches the maximum threshold of 100 percent.

Configuring Permit Flows

You can configure permit flows by entering one of the following commands:

Command	Purpose
logging ip access-list cache max-permit-flows *num*	Sets the number of permit flows, where *num* is the number of flows.

Command	Purpose
[**no**] **logging ip access-list cache max-permit-flows**	Reverts the configuration to the default permit flow value of 3000.

These examples show permit flow syslog messages:

- New flow notification message:

```
Oct 6 17:05:10 192.0.2.199 1 1988-01-19T07:17:43.810 192.0.2.168 n1k-acllog -
ACLLOG-PERMIT-FLOW-CREATE VSM ID: 192.0.2.168, vse ID:
42205f8e-0959-fbe2-6403-bf6d9f75c384
SGT :25 DGT :25 Source IP: 192.0.2.3, Destination IP: 192.0.2.2
Source Port: 40116, Destination Port: 2048
Source Interface: Veth15, Protocol: "TCP"(6), Hit-count = 19
```

- Periodic flow reporting message:

```
Oct 6 17:06:38 192.0.2.5 1 1988-01-19T07:17:53.809 192.0.2.168 n1k-acllog -
ACLLOG-PERMIT-FLOW-INTERVAL VSM ID: 192.0.2.168, vse ID:
422feff2-360b-0906-e9c1-895960e5b762
SGT :25 DGT :25 Source IP: 192.0.2.2, Destination IP: 192.0.2.3
Source Port: 2048, Destination Port: 40063
Source Interface: Veth6, Protocol: "TCP"(6), Hit-count = 2100
```

- Threshold crossing alarm messages:

```
-  Oct 6 04:17:22 sfish-231-157.cisco.com 1 2011-08-28T11:14:24 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 75 percent

limit (3969)
-  Oct 6 04:17:26 sfish-231-157.cisco.com 1 2011-08-28T11:14:26 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 90 percent

limit (4969)
-  Oct 6 04:17:27 sfish-231-157.cisco.com 1 2011-08-28T11:14:31 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 100 percent

limit (5000)
```

Configuring Deny Flows

You can configure deny flows by entering one of the following commands:

Command	Purpose
logging ip access-list cache max-deny-flows *num*	Sets the number of deny flows, where *num* is the number of flows.
[**no**] **logging ip access-list cache max-deny-flows**	Reverts the configuration to the default deny flow value 3000.

These examples show deny flow syslog messages:

- New flow notification message:

```
Oct 6 17:05:10 192.0.2.199 1 1988-01-19T07:17:43.810 192.0.2.168 n1k-acllog -
ACLLOG-DENY-FLOW-CREATE VSM ID: 192.0.2.168, VSE ID: 42205f8e-0959-fbe2-6403-bf6d9f75c384
SGT :25 DGT :25 Source IP: 192.0.2.3, Destination IP: 192.0.2.2
Source Port: 40116, Destination Port: 2048
Source Interface: Veth15, Protocol: "TCP"(6), Hit-count = 19
```

- Periodic flow reporting message:

```
Oct 6 17:06:38 192.0.2.5 1 1988-01-19T07:17:53.809 192.0.2.168 n1k-acllog -
ACLLOG-DENY-FLOW-INTERVAL VSM ID: 192.0.2.168, VSE ID:
422feff2-360b-0906-e9c1-895960e5b762
SGT :25 DGT :25 Source IP: 192.0.2.2, Destination IP: 192.0.2.3
Source Port: 2048, Destination Port: 40063
Source Interface: Veth6, Protocol: "TCP"(6), Hit-count = 2100
```

- Threshold crossing alarm messages:

```
-  Oct 6 04:17:27 sfish-231-157.cisco.com 1 2011-08-28T11:14:31 - n1k-acllog -
ACLLOG-MAX-DENY-FLOW-REACHED The number of ACL log deny-flows has reached 75 percent
limit
(4330)
-  Oct 6 04:18:27 sfish-231-157.cisco.com 1 2011-08-28T11:15:31 - n1k-acllog -
ACLLOG-MAX-DENY-FLOW-REACHED The number of ACL log deny-flows has reached 90 percent
limit
(4630)
-  Oct 6 04:20:17 sfish-231-157.cisco.com 1 2011-08-28T11:17:20 - n1k-acllog -
ACLLOG-MAX-PERMIT-FLOW-REACHED The number of ACL log permit-flows has reached 100 percent

limit (5000)
```

Configuring Syslog Server Severity Levels

You can set the severity level of a syslog message and up to three remote servers to which you want the message to be sent using the following commands:

Command	Purpose
[**no**] **acllog match-log-level** *level*	Sets the severity level at which syslog messages are sent, where *level* is the severity code from 0 to 7. The no acllog match-log-level level command reverts the RBACL log level to the default severity level 6.
[**no**] **logging ip access-list cache max-deny-flows** *number*	Sets the maximum number of deny flows to *number* per module. The **no logging ip access-list cache max-deny-flows** *number* sets the maximum number of deny-flows to the default value of 3000.
[**no**] **logging ip access-list cache max-permit-flows** *number*	Set the max-permit-flows to a specified number per module. The **no logging ip access-list cache max-permit-flows** *number* sets the maximum number of permit-flows to the default value of 3000.
logging server *A.B.C.D 0-7*	Specifies the syslog server on which you want to set a severity level, where *A.B.C.D* is the syslog server IP address and 0 to 7 are the severity levels you can choose.

Note For ACL logging to work, the RBACL Logging level should be less than or equal to the Syslog level.

The severity level range is from 0 to 7 (default is 6):

Severity Code	Severity Level	Description
0	Emergency	System is unusable
1	Alert	Action must be taken immediately
2	Critical	Critical conditions
3	Error	Error conditions
4	Warning	Warning conditions
5	Notice	Normal but significant condition
6 (Default)	Informational	Informational messages
7	Debug	Debug-level messages

Verifying the Cisco TrustSec Configuration

Use the following commands to verify the configuration:

Command	Purpose
show aaa authentication	Displays the AAA authentication configuration on the Cisco Nexus 1000V.
show aaa authorization	Displays the AAA authorization configuration on the Cisco Nexus 1000V.
show cts	Displays the global Cisco TrustSec configuration on the Cisco Nexus 1000V.
show cts sxp	Displays the Cisco TrustSec SXP configuration.
show cts device tracking	Displays the Cisco TrustSec device tracking configuration.
show cts sxp connection	Displays Cisco TrustSec SXP connections.
show cts role-based sgt-map	Displays all the mapping between IP address/Subnet and SGT for Cisco TrustSec.
show cts ipsgt entries	Displays all the IP address/Subnet to SGT mappings.
show cts interface delete-hold timer	Displays the Cisco TrustSec interface delete hold timer period.
show cts environment-data	Displays Cisco TrustSec environmental data.
show cts interface	Displays the Cisco TrustSec configuration for all interfaces.

Command	Purpose
show cts interface ethernet *slot*/*port*	Displays the Cisco TrustSec configuration for the specified Ethernet interface.
show cts interface vethernet *number*	Displays the Cisco TrustSec configuration for the specified virtual Ethernet interface.
show cts role-based access-list	Displays Cisco TrustSec SGACL information.
show cts role-based counters	Displays the configuration status of RBACL statistics and lists statistics for all RBACL policies.
show cts role-based policy	Displays Cisco TrustSec SGACL policy information.
show running-configuration cts	Displays the running configuration information for Cisco TrustSec.

Secure Login Enhancements

Starting with Cisco Nexus 1000VE for VMware vSphere Release 5.2(1)SV5(1.2), you can configure login parameters to enhance secure login to Cisco Nexus 1000VE switches.

Configuring Login Parameters

Use this task to configure your Cisco Nexus 1000VE device for login parameters that help detect suspected DoS attacks and slow down dictionary attacks.

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

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

Step 1 **configure terminal**

Example:

```
Switch# configure terminal
```

Enters global configuration mode.

Step 2 **[no] login block-for** *seconds* **attempts** *tries* **within** *seconds*

Example:

```
Switch(config)# login block-for 100 attempts 2 within 100
```

Configures your Cisco NX-OS device for login parameters that help you detect DoS attack.

Note This command must be issued before any other login command can be used.

Step 3 **[no] login quiet-mode access-class** {*acl-name* | *acl-number*}

Example:

```
Switch(config)# login quiet-mode access-class myacl
```

(Optional) Although this command is optional, it is recommended that it be configured to specify an ACL that is to be applied to the device when the device switches to quiet mode. When the device is in quiet mode, all login requests are denied and the only available connection is through the console.

Step 4 **exit**

Example:

```
Switch(config)# exit
```

Exits to privileged EXEC mode.

Step 5 **show login failures**

Example:

```
Switch# show login failure
```

Displays login parameters.

- **failures** - Displays information related to failed login attempts.

Configuration Examples for Login Parameters

Setting Login Parameters Example

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

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

Showing Login Parameters Example

The following sample output from the **show login** command verifies that secure login parameters have been specified:

```
Switch# show login

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

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

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

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

```
Switch# show login failures

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

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

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

The following example shows how to clear the failed login attempts using the clear command:

```
Switch# clear login failures

This command is provided to clear statistics about failure details

Usage:

Nexus 1000ve# sh login failures

Information about last 20 login failure's with the device.
-------------------------------------------------------------------------------------------
Username                                 Line    SourceIPAddr    Appname         TimeStamp
-------------------------------------------------------------------------------------------
admin                                    ssh     10.78.184.85    login           Mon Mar 18
 07:38:16 2019
admin                                    ssh     10.78.184.85    login           Mon Mar 18
 07:38:18 2019
-------------------------------------------------------------------------------------------
Nexus 1000ve#

Nexus 1000ve# clear login failures
Nexus 1000ve#
Nexus 1000ve# sh login failures
```

Guidelines and Limitations

Follow these usage guidelines and limitations while configuring Secure Login Enhancement:

- When the Quite mode is activated and login access is blocked for SSH and Telnet with ACLs, existing login sessions are also stopped. This behavior is consistent with the regular ACL behavior as applied to any interface handling traffic.

- Ensure that ACLs have last entries as **permit ip any any** in order to allow any other permitted protocol traffic to pass through the management interface, other than those handled by ACL entries. Default policy otherwise is to deny such additional IP traffic.
- PNSC access to VSM could get blocked due to ACL. To avoid this issue, configure secure login on VSM such that https access between VSM and PNSC is bidirectional and open the port 443.
- Secure login feature does not work together with ACLs directly configured with management interface (mgmt0) for VSM. Both are mutually exclusive configurations.