

Cisco TrustSec

- Feature History for Cisco TrustSec, on page 1
- Cisco TrustSec, on page 2
- Identities and credentials, on page 4
- Guidelines for Cisco TrustSec, on page 6
- Configure Cisco TrustSec credentials, on page 6
- PAC download, on page 8
- PUSH ID (HTTPS) traceability, on page 10
- Cisco TrustSec Environment Data , on page 14
- Security Groups and Security Group Tags, on page 16

Feature History for Cisco TrustSec

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 26.2.1	PUSH ID (HTTPS): PUSH ID (HTTPS) feature support has been introduced.	Cisco Catalyst 9300 Series Switches
Cisco IOS XE 26.2.1ea	Cisco TrustSec: Cisco TrustSec feature support has been introduced.	Cisco C9550 Series Smart Switches
Cisco IOS XE 26.1.1	CTS policy server list enablement over RADIUS for IPv6: Introduced the support for CTS policy server list enablement over RADIUS for IPv6.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches
Cisco IOS XE 26.1.1	PAC download over IPv6: Introduced the support for PAC download over IPv6.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	Cisco TrustSec: Cisco TrustSec builds secure networks by establishing domains of trusted network devices. Each device in the domain is authenticated by its peers, and communication between devices is secured with encryption, message integrity checks, and data-path replay protection.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Cisco TrustSec

Cisco TrustSec builds secure networks by establishing domains of trusted network devices. Each device in the domain is authenticated by its peers, and communication between devices is secured with encryption, message integrity checks, and data-path replay protection.

Key components of Cisco TrustSec

Cisco TrustSec architecture incorporates these three key components:

- Authenticated networking infrastructure

  After the first device (called the seed device) authenticates with the authentication server to begin the Cisco TrustSec domain, each new device added to the domain is authenticated by its peer devices already within the domain. The peers act as intermediaries for the domain's authentication server. Each newly-authenticated device is categorized by the authentication server and assigned a security group number based on its identity, role, and security posture.

- Security group-based access control:

  Access policies within the Cisco TrustSec domain are topology-independent, based on the roles (as indicated by security group number) of source and destination devices rather than on network addresses. Individual packets are tagged with the security group number of the source.

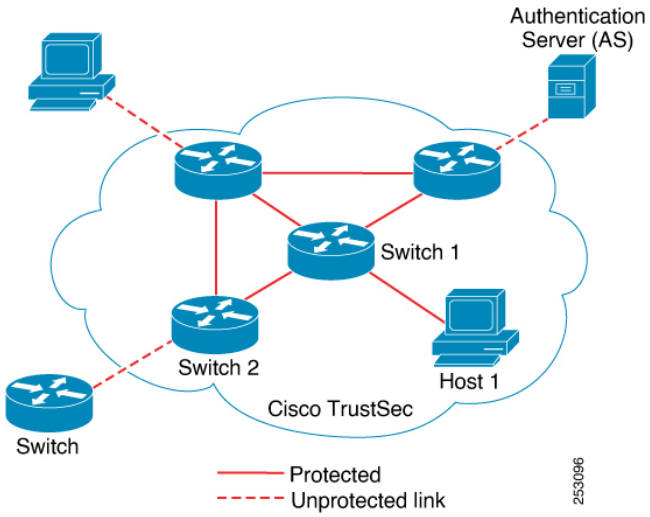
- Secure communication:

  With encryption-capable hardware, communication on each link between devices in the domain can be secured with a combination of encryption, message integrity checks, and data-path replay protection mechanisms.

Figure shows a Cisco TrustSec domain with

- An endpoint device and several networking devices within the Cisco TrustSec domain.
- An endpoint device and one networking device outside the Cisco TrustSec domain. This is because they are either not Cisco TrustSec-capable devices or they have been refused access.
- The authentication server outside of the Cisco TrustSec domain. The authentication server can either be a Cisco Identities Service Engine (Cisco ISE), or a Cisco Secure Access Control System (Cisco ACS).

Figure 1: Cisco TrustSec Domain



Roles in the Cisco TrustSec authentication process

In the Cisco TrustSec authentication process, there are these roles:

- Supplicant

  An unauthenticated device that is connected to a peer within the Cisco TrustSec domain and is attempting to gain access to the domain.

- Authentication Server

  The system responsible for verifying the supplicant's identity and assigning policies that control the supplicant's access to services within the Cisco TrustSec domain.

- Authenticator

  An authenticated device that is already a member of the Cisco TrustSec domain and is authorized to authenticate new supplicant peers on behalf of the authentication server.

Authentication process sequence

When a connection is established between a supplicant and an authenticator, the process typically follows these steps:

1. Authentication (802.1X):

   The supplicant's credentials are verified by the authentication server, with the authenticator serving as an intermediary. Mutual authentication takes place between the supplicant and the authenticator.

2. Authorization:

   Based on the supplicant's identity, the authentication server issues authorization policies, such as security group assignments and access control lists (ACLs), to both connected peers. The server also shares each peer's identity with the other, enabling the appropriate policies to be applied to the link.

Once these steps are successfully completed, the authenticator transitions the link from an unauthorized (blocking) state to an authorized state, allowing the supplicant to join the Cisco TrustSec domain.

Identities and credentials

The following sections describes these foundational elements and how they interact in a TrustSec deployment.

- Device identities,
- Device credentials,
- User credentials, and
- Protected Access Credentials (PACs)

Device Identities

Cisco TrustSec uses assigned device names (device IDs) instead of IP or MAC addresses to uniquely identify switches. These device IDs are used for authorization policy lookups and password lookups during authentication.

Device Credentials

Cisco TrustSec supports password-based credentials and uses MSCHAPv2 for mutual authentication.

Device credentials are used during EAP-FAST phase 0 (PAC provisioning) exchanges. Subsequent link bring-ups use EAP-FAST phase 1 and 2, leveraging the provisioned PAC.

User Credentials

Cisco TrustSec does not require a specific type of user credential for endpoint devices. Any authentication method supported by the authentication server (for example, MSCHAPv2, GTC, RSA OTP for Cisco ACS 5.1) can be used.

Protected Access Credential

A PAC is a unique shared credential for mutual client and server authentication, eliminating the need for Public Key Infrastructure (PKI) and digital certificates.

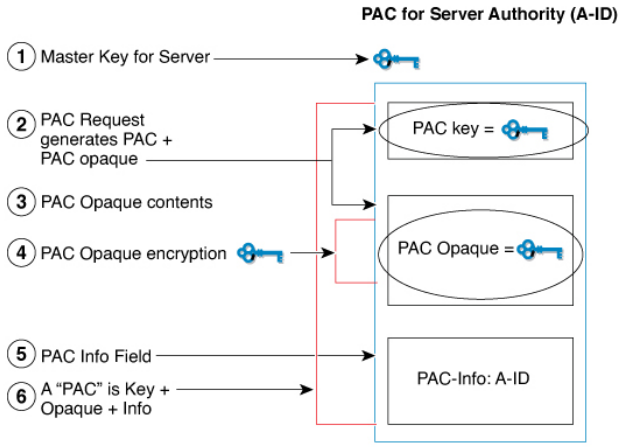
PAC Creation Steps

1. The server's Authority Identity (A-ID) maintains a local master key.
2. When a client, initiator identity (I-ID), requests a PAC, the server generates a unique PAC key and PAC-Opaque field.
3. The PAC-Opaque field contains the PAC key, I-ID, and key lifetime, encrypted with the master key.
4. A PAC-Info field containing the A-ID is created.
5. The PAC is automatically distributed or imported to the client.

Note The server does not maintain the PAC or PAC key, enabling a stateless EAP-FAST server.

The figure below describes the PAC's construction. A PAC consists of the PAC-Opaque, PAC Key, and PAC-Info fields. The PAC-Info field contains the A-ID.

Figure 2: PAC's Process Flow



PAC Provisioning

- In Secure RADIUS, the PAC key is provisioned into each device during authentication to derive the shared secret.
- Clients send an additional RADIUS attribute containing the PAC-Opaque field, which the server interprets to recover information and validate identity.
- EAP-FAST Phase 0 is used for automatic PAC provisioning.

Note Cisco Catalyst 9000 Series Switches have the capability of storing multiple PACs coming from different RADIUS servers to enable high availability. This way, even if the first trustsec in the list does not respond, the switch can request the trustsec info to the second one in the list and skip the PAC provisioning phase.

PAC-less Authentication

PAC-less mode simplifies the deployment of Cisco TrustSec policies by eliminating the need for a Protected Access Credential (PAC), which is typically required for secure communication between devices and the Identity Services Engine (ISE). This approach is particularly advantageous in environments with multiple ISE nodes, as devices can seamlessly connect to a secondary ISE node if the primary becomes unavailable, without the need to re-establish the PAC, thus reducing service interruptions.

Benefits

By removing the dependency on PACs, AAA PAC-less authentication streamlines the authentication process. This enhances scalability, improves the user experience, and enables support for modern authentication approaches consistent with Zero Trust security principles.

PAC-less Authentication Workflow

1. In PAC-less mode, devices begin authentication by sending a RADIUS request that includes the Cisco TrustSec username, password, and an EAP attribute message.
2. The ISE replies with a RADIUS access-challenge to initiate an EAP-FAST session.
3. Once the EAP-FAST session is established, the ISE provides the PAC opaque and PAC information. The PAC opaque contains the PAC key and user identity, encrypted using the EAP-FAST server master key. The PAC information includes the server identity and Time-to-Live timers.
4. The PAC opaque is then embedded in the Message-Authenticator field of subsequent Cisco TrustSec RADIUS requests to the ISE. This enables the secure download of environment data and Security Group Access Control List (SGACL) policies.

Guidelines for Cisco TrustSec

General Guidelines

- Cisco TrustSec is not supported in FIPS mode.
- Cisco TrustSec cannot be configured on a pure bridging domain with the IPSG enabled.
- Cisco Trustsec does not support Security Association Protocol (SAP).

Guidelines for Identities and Credentials

Protected access credential (PAC) provisioning fails and remains in hung state, when an invalid device ID is specified. Even after clearing the PAC, and configuring the correct device ID and password, PAC still fails.

As a workaround, in the Cisco Identity Services Engine (ISE), uncheck the Suppress Anomalous Clients option in the **Administration** > **System** > **Settings** > **Protocols** > **Radius** menu for PAC to work.

Configure Cisco TrustSec credentials

This task allows you to configure your device with device ID, password and authentication method.

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

Enter your password, if prompted.

Step 2 **cts credentials id** *cts-id* **password** *password*

Example:

```
Device# cts credentials id ctsid password abcd
```

Configures the Cisco TrustSec device ID for this device to use when authenticating with other Cisco TrustSec devices with EAP-FAST because Cisco TrustSec requires each device in the network to identity itself uniquely.

- The *cts-id* argument has a maximum length of 32 characters and is case sensitive.
- The *password* argument is the password or this device to use when authenticating with other Cisco TrustSec devices with EAP-FAST.

Step 3 **configure terminal**

Example:

```
Device(config)# configure terminal
```

Enters global configuration mode.

Step 4 **aaa new-model**

Example:

```
Device(config)# aaa new-model
```

Enables new RADIUS and AAA access control commands and functions and disables old commands.

Step 5 **aaa authentication dot1x default group radius**

Example:

```
Device(config)# aaa authentication dot1x default group radius
```

Specifies that RADIUS servers are used for authentication on interfaces running IEEE 802.1X.

Step 6 **cts authorization list network** *list-name*

Example:

```
Device(config)# cts authorization list network cts-mlist
```

Specifies a list of AAA servers for the Cisco TrustSec seed device to use.

Step 7 **aaa authorization network** *list-name* **group radius**

Example:

```
Device(config)# aaa authorization network cts-mlist group radius
```

Specifies the Cisco TrustSec authorization list name for allnetwork-related service requests from RADIUS servers.

Step 8 **exit**

Example:

```
Device(config)# exit
```

Exits global configuration mode.

Step 9 **show cts server-list**

Example:

```
Device# show cts server-list
```

Displays the RADIUS the server configurations for Cisco TrustSec seed devices.

Step 10 **show cts credentials**

Example:

```
Device# show cts credentials
```

Displays the Cisco TrustSec device ID. The stored password is never displayed.

PAC download

Cisco TrustSec Protected Access Credentials (PAC) download is a security process that

- establishes a secure identity for a network device with the ISE
- utilizes the EAP-FAST protocol over a RADIUS transport layer, and
- serves as a prerequisite for fetching TrustSec environment data and SGACL policies.

PAC download over IPv6

PAC download over IPv6 is a network transport capability that

- establishes a RADIUS session with a policy server using literal IPv6 addresses
- utilizes the EAP-FAST protocol to acquire PAC over an IPv6 transport, and
- automates credential re-acquisition when the server address type is modified.

This capability enables the TrustSec policy plane to function in IPv6-only or dual-stack environments.

Policy server version for PAC download over IPv6

PAC download over IPv6 is supported from Cisco IOS XE 26.1.1 release and requires policy server (ISE) version 3.5, or later.

Benefits of PAC download over IPv6

Key benefits of PAC download over IPv6 include:

- Address flexibility: supports literal IPv6 addresses for policy server identification, allowing the device to resolve server locations through DNS
- Transport security: maintains the security of the EAP-FAST session while transitioning the underlying transport from IPv4 to IPv6
- Dynamic updates: ensures that the device remains synchronized with the policy server by automatically re-triggering PAC requests when a user switches between address types in the configuration.

ISE version for PAC download over IPv6

PAC download over IPv6 is supported from Cisco IOS XE 26.1.1 release and requires ISE version 3.5, or later.

Restrictions for PAC download

Follow these restrictions while configuring policy server for PAC download:

- The device does not support link local IP addresses and Full Qualified Domain Name (FQDN) for the RADIUS server configuration.
- The support for IPv4 and IPv6 address types for PAC download is not mutually exclusive. If you configure both IPv4 and IPv6 address types on the device, only one of the configurations, that is, the latest configuration will take effect.
- The device does not support mixed mode address types—scenarios in which device source address type is IPv4 and ISE is of IPv6 type, and the other way around.

Configure policy server for PAC download

Use this procedure to establish an IPv6 connection to a policy server to acquire the PAC required for Cisco TrustSec (CTS) operations. Similarly, use IPv4 address for PAC download over IPv4.

Perform this task when setting up a new TrustSec-enabled device or when migrating an existing policy server connection to an IPv6 transport.

Before you begin

- Verify that the RADIUS server (ISE) is reachable through the IPv6 or IPv4 address.
- Ensure that the shared secret (PAC key) is available.

Procedure

Step 1 Configure the RADIUS server by specifying the IPv4 or IPv6 address, and the authentication and accounting ports used by the policy server.

Example:

```
Switch(config)#radius server test-server
Switch(config-radius-server)#address ipv6 2001::10 auth-port 1812 acct-port 1813
```

The device does not support using Fully Qualified Domain Name (FQDN) for the RADIUS server.

Step 2 Define the PAC key for the RADIUS server.

Example:

```
Switch(config-radius-server)#pac key testkey123
```

This step enables the device to authenticate and initiate the EAP-FAST session for PAC acquisition.

The key must match the shared secret configured on the ISE server.

Step 3 Monitor the PAC provisioning status.

a) Check the CTS PAC details.

Example:

```
Switch#show cts pac
AID: 60DABBF8AD435A7B63900EE07E68D2FD
PAC-Info:
  PAC-type = Cisco Trustsec
  AID: 60DABBF8AD435A7B63900EE07E68D2FD
  I-ID: ha_senth_cat9k
  A-ID-Info: Identity Services Engine
  Credential Lifetime: 16:17:20 UTC Mon Feb 24 2025
PAC-Opaque: 000200B800030001000...705E
Refresh timer is set for 12w4d
Switch#
```

b) Check the CTS provisioning queue details.

Example:

```
Switch#show cts provisioning queue
Server: 2001:420:54FF:4::400:11, Type: Radius, Provisioned: YES
AID: 3ec5d7e02bde9fd04c453918a5e2d3b4
```

The device establishes a secure session with the policy server and successfully acquires the PAC over IPv4 or IPv6 transport, enabling subsequent TrustSec policy downloads.

PUSH ID (HTTPS) traceability

PUSH ID (HTTPS) traceability is a Cisco TrustSec capability that improves transaction-level tracking of Security Group ACL (SGACL) policy provisioning between Cisco Identity Services Engine (ISE) and Network Access Devices (NADs). It applies to HTTPS-based TrustSec communication that uses REST API version 3. It does not apply when SGACL policies are provisioned through Remote Authentication Dial-In User Service (RADIUS).

Key identifiers for PUSH ID (HTTPS) traceability

PUSH ID (HTTPS) traceability uses these identifiers:

- Policy Version ID: Signature representing the complete TrustSec policy configuration. It changes whenever a policy or related object changes.
- Push ID: Identifies an ISE-initiated Change of Authorization (CoA) transaction.
- Pull ID: Identifies a device-initiated policy or environment-data request.
- Propagation ID: Tracks a system-wide update initiated through the user interface or API. It remains within ISE and is not sent to NADs.
- Report Type: Determines whether the device reports changed objects, the full configuration, or object-version information.

- GenID: Identifies the version of an individual policy object and supports efficient reporting and synchronization.

Benefits of PUSH ID (HTTPS) traceability

PUSH ID (HTTPS) traceability provides these benefits:

- Correlates CoA messages, policy-download requests, responses, and device reports.
- Provides transaction-level visibility into SGACL provisioning.
- Helps administrators identify missed, incomplete, or partially successful updates.
- Improves validation of policy consistency between ISE and individual NADs.
- Reduces the risk of undetected policy drift.
- Supports faster troubleshooting in large TrustSec deployments.

Limitations of PUSH ID (HTTPS) traceability

- The feature is unavailable when RADIUS is used for TrustSec policy download.
- Selecting RADIUS automatically disables PUSH ID traceability and releases its tracking resources.
- Initial supported releases do not provide periodic device reporting.
- Device reports do not include certificate information in the referenced release.
- REST API version 3 requires compatible ISE and NAD software.
- When version 3 is unsupported, communication falls back to version 2 and then version 1; full PUSH ID behavior might therefore be unavailable.
- Push ID tracking is supported over both IPv4 and IPv6 HTTPS connections.
- The source does not identify a separate device CLI command for enabling the feature.

How PUSH ID (HTTPS) works

PUSH ID (HTTPS) traceability tracks two kinds of transactions: an Identity Services Engine (ISE) initiated update and a device-initiated refresh.

Summary

These actors take part in PUSH ID (HTTPS) traceability:

- Cisco Identity Services Engine (ISE): Initiates Change of Authorization (CoA) updates, assigns identifiers, and validates device reports.
- Network Access Device (NAD): Stores identifiers, exchanges SGACL and environment data, and reports transaction results to ISE.

ISE and the NAD exchange Push ID, Pull ID, Policy Version ID, and GenID values so that each SGACL policy provisioning transaction can be tracked end to end.

Workflow

These stages describe how PUSH ID (HTTPS) works for an ISE-initiated update.

1. ISE generates a Push ID and initiates a Change of Authorization (CoA) operation, then sends the Push ID and the current Policy Version ID to the target NAD.
2. The NAD stores the identifiers and returns a CoA-Accept associated with the transaction. The NAD includes the Push ID in subsequent HTTPS requests made as part of that CoA session.
3. ISE and the NAD exchange the required SGACL and environment data.
4. After processing the update, the NAD reports the Push ID, Policy Version ID, and the resulting Security Group Tag (SGT) and SGACL GenIDs to ISE. ISE validates the report and returns an HTTP 200 OK. After receiving the successful response, the NAD marks the transaction complete and clears its session record.



Note If part of a transaction fails, GenID reporting can indicate which objects succeeded or failed, helping isolate partial policy updates.

5. For a device-initiated refresh, the NAD creates a new Pull ID. This occurs for independent events such as policy refreshes, enforcement enablement, or Security Group Tag (SGT) additions. The Pull ID is carried through the corresponding requests, responses, and final report.

Enable PUSH ID (HTTPS) traceability

Devices attempt to use REST API version 3 by default. If ISE returns HTTP 404, the NAD falls back to version 2, and if necessary, falls back again to version 1. You can check the negotiated API support through HTTP header information and related syslog messages.

Before you begin

- Cisco ISE 4.0 or later.
- A compatible NAD release, such as 26.2.1 or later.
- HTTPS-based TrustSec policy provisioning.
- REST API version 3 support on ISE and the NAD.

Procedure

Step 1 Enable manageability and traceability through the available option on the ISE NAD configuration page.

No separate NAD CLI enablement command is documented.

Caution
Choosing RADIUS as the download method automatically disables PUSH ID (HTTPS) traceability.

Step 2 Verify that REST API version 3 and PUSH ID capability negotiated successfully.

Example:

```
show cts transaction-id
```

Example output:

```
------------------ show cts transaction-id ------------------
==================== Transaction Summary ====================
PushID Feature Status : Enabled
Global policy ver id : 56ab-13ade-1a06bed3be9
RESTAPI version : RESTAPI/3.0
Transaction type : Environment-Data
Push-id : 0
Pull-id : 1788505302
Report-type : DELTA-WITH-GENID
Report Sent : No
```

When REST API version 3 and PUSH ID capability are successfully negotiated, transaction tracking operates automatically.

Best practices for PUSH ID (HTTPS)

Follow these best practices to get full traceability benefit from PUSH ID (HTTPS).

- Use Cisco ISE 4.0 or later and a NAD software release that supports PUSH ID from 26.2.1 and later.
- Use REST API version 3 over HTTPS for full traceability support.
- Confirm that the Policy Version ID and Push ID shown on the NAD match the corresponding ISE transaction records.
- Check REST API negotiation messages when traceability information is missing.
- Maintain Push ID, Pull ID, Policy Version ID, and report-type state across active and standby nodes in high-availability deployments.
- Validate synchronization after a stateful switchover, because device-generated Pull IDs are refreshed after failover.
- Use GenID-based reports to identify changed objects and avoid unnecessarily processing the complete configuration.

Troubleshoot PUSH ID (HTTPS) traceability issues

Procedure

Step 1 Check for missing Push ID (HTTPS) information.

- Confirm that HTTPS, rather than RADIUS, is configured for TrustSec policy provisioning.
- Verify that ISE and the NAD support REST API version 3.
- Check syslog messages for API-version negotiation and fallback.

- Inspect HTTP response headers for advertised API support.
- Confirm that manageability and traceability are enabled for the NAD in ISE.

Step 2 Isolate an incomplete PUSH ID (HTTPS) transaction.

- Compare the Push ID or Pull ID on the device with the corresponding ISE transaction.
- Compare the Policy Version ID maintained by ISE and the NAD.
- Review the device's final report for failed or missing object GenIDs.
- Confirm that ISE returned HTTP 200 OK for the report.
- Check whether a failure affected only one object within an otherwise successful transaction.

Step 3 Validate PUSH ID (HTTPS) state after a stateful switchover.

- Confirm that the active node resumed with the last synchronized Push ID.
- Expect new device-generated Pull IDs after stateful switchover.
- Verify that Policy Version ID and reporting state remain aligned between active and standby in the NAD.

Cisco TrustSec Environment Data

Environment data comprises of operational data that supplement Cisco TrustSec functions. The environment data request from a device to Cisco ISE consists of the following data:

- Device name: Specifies the name of the device.
- Device capability: Specifies additional data.

The environment data response from Cisco ISE to a device consists of the following data:

- Device security group tag (SGT): Derived from Cisco ISE based on the device name.
- Server list: Displays the list of Cisco TrustSec servers specified in Cisco ISE.
- SG-Name Table: Displays the mapping between SGT and the device name. SGT is displayed in numerals and the device name in text format.
- Refresh time: Indicates the time when the environment data will be refreshed.

Note

- As part of Cisco TrustSec environment data refresh, the last received servers are deleted and newly received servers are added to the server list. As a result of the refresh, the server list statistics restarts from zero, and the server status is set to *Inactive* and the IP address state is set to *Reachable*. The device then updates the server statistics and status based on the subsequent policy request and response.
- Starting with Cisco IOS XE Bengaluru 17.4.1, you can configure automated tester to be VRF aware. You can use the **vrf** keyword with the **automate-tester** command to enable automate-tester for a non-default VRF.

  For VRF aware automate-tester to work, you have to configure **global config ipv4/ipv6 source interface** *interface-name* **vrf** *vrf-name* command.

CTS policy server list enablement over RADIUS for IPv6

From Cisco IOS XE Release 26.1.1 and later, ISE pushes the IPv6 addresses of PSNs in addition to the IPv4 addresses to the devices as part of the environment data. When the device downloads the SGACL policies, it uses these PSN IPv6 addresses to reach the ISE.

Prior to this release, the server-list IPs from ISE received as part of environment data was only of IPv4 type.

Server selection and failover logic to download the SGACL policy

When the device receives a server list containing both IPv4 and IPv6 addresses from ISE, it follows specific selection and failover rules:

- The device uses a round-robin failover method to pick an active server.
- If the primary server in the list becomes unreachable, the device automatically attempts to connect to the next available server in the sequence.
- The device typically prioritizes the server address type that matches its configured source interface address type for SGACL policy download.

Restriction for CTS policy server list enablement over RADIUS

The device does not support mixed mode address types—scenarios in which device source address type is IPv4 and ISE is of IPv6 type, and the other way around.

Verify CTS policy server list enablement for IPv6

Use this procedure to verify if the IPv6 server lists are received from ISE as part of downloading the environment data to the device.

Procedure

Verify the details of environment data downloaded to the device from the ISE server.

Example:

```
Switch# show cts environment-data
CTS Environment Data
====================
Current state = COMPLETE
Last status = Successful
Service Info Table:
Local Device SGT:
  SGT tag = 2-48:TrustSec_Devices
Server List Info:
Installed list: CTSServerList1-0003, 3 server(s):
 *Server: 2001:420:54FF:4::400:11, port 1812, A-ID 1695AF86D38B22DC7C9500408E2DD35D
          Status = DEAD
          auto-test = TRUE, keywrap-enable = FALSE, idle-time = 60 mins, deadtime = 20 secs

 *Server: 2001:420:54FF:4::400:12, port 1812, A-ID 1695AF86D38B22DC7C9500408E2DD35D
          Status = ALIVE
          auto-test = TRUE, keywrap-enable = FALSE, idle-time = 60 mins, deadtime = 20 secs

 *Server: 2001:420:54FF:4::400:13, port 1812, A-ID 1695AF86D38B22DC7C9500408E2DD35D
          Status = DEAD
          auto-test = TRUE, keywrap-enable = FALSE, idle-time = 60 mins, deadtime = 20 secs
Security Group Name Table:
    0-48:Unknown
    2-48:TrustSec_Devices
    3-62:Network_Services
    4-56:Employees
    5-48:Contractors
    6-49:Guests
    7-48:Production_Users
    8-49:Developers
    9-49:Auditors
    10-48:Point_of_Sale_Systems
    11-48:Production_Servers
    12-0177:Development_Servers
    13-48:Test_Servers
    14-56:PCI_Servers
    15-48:BYOD
    16-01:sgt_101
Environment Data Lifetime = 86400 secs
Last update time = 05:26:38 UTC Tue Jul 1 2025
Env-data expires in   0:23:52:36 (dd:hr:mm:sec)
Env-data refreshes in 0:23:52:36 (dd:hr:mm:sec)
Cache data applied           = NONE
State Machine is running
Retry_timer (60 secs) is not running
Switch#
```

Security Groups and Security Group Tags

These sections provide information about security groups and security group tags.

Security Groups

A security group is a collection of users, endpoint devices, and resources that share common access control policies. Administrators define these groups in Cisco ISE or Cisco Secure ACS. When new users or devices join the Cisco TrustSec domain, the authentication server automatically assigns them to the appropriate security groups. Each group receives a unique 16-bit security group number, which is globally recognized within the Cisco TrustSec domain. The total number of security groups is determined by the number of authenticated network entities, and administrators do not need to manually configure these group numbers.

Security Group Tags and Packet Tagging

After a device is authenticated, Cisco TrustSec tags every packet originating from that device with a Security Group Tag (SGT). This tag includes the device's assigned security group number and travels with the packet across the network within the Cisco TrustSec header. The SGT acts as a single label that defines the source's access privileges throughout the enterprise network.

Source and Destination Group Tags

The SGT represents the security group of the packet's source, so it is often called the source SGT. The device receiving the packet is also assigned to a security group, known as the destination security group. For simplicity, this may be referred to as the destination group tag (DGT), although the Cisco TrustSec packet itself does not include the destination device's security group number.

Determining the Source Security Group Tag

When a packet enters a Cisco TrustSec domain, the network device at the ingress point must identify the source SGT so it can tag the packet appropriately before forwarding it into the TrustSec environment. Similarly, the egress device needs to determine the packet's SGT to enforce Security Group Access Control Lists (SGACLs).

There are several ways a network device can determine the SGT for a packet:

1. Policy Acquisition from the Authentication Server

   After the Cisco TrustSec authentication process, the network device obtains policy information from the authentication server. This information includes whether the peer device is trusted. If the device is not trusted, the authentication server provides an SGT, which the network device applies to all packets received from that peer.

2. SGT Included in the Packet

   If the packet is received from a trusted peer device, it will already carry the SGT. This situation applies to devices that are not the initial entry point in the Cisco TrustSec domain for the given packet.

3. Identity Port Mapping (IPM)

   With Identity Port Mapping, the network administrator can manually assign an identity to the link connected to the peer device. The network device then requests policy details, including the SGT and trust status, from the authentication server.

4. Source IP Address Lookup

In certain scenarios, policies can be configured to assign an SGT to a packet based on its source IP address. The SGT Exchange Protocol (SXP) can also be used to automatically populate the mapping table between IP addresses and SGTs.

Determining the Destination Security Group Tag

The egress network device in a Cisco TrustSec domain is responsible for identifying the destination security group (DGT) in order to enforce SGACLs. To determine the destination security group for a packet, the network device uses the same methods as those used for determining the source security group, except that it cannot extract the group number from a packet tag—since the destination group number is not included in the packet's tag.

In certain scenarios, ingress devices or other intermediate network devices may have access to destination group information. When this occurs, SGACLs can be enforced at these devices rather than solely at the egress point.