Cisco TrustSec with SXPv4

Cisco TrustSec (CTS) builds secure networks by establishing domains of trusted network devices. Each device in the domain is authenticated by its peers. Communication on the links between devices in the domain is secured with a combination of encryption, message integrity check, and data-path replay protection mechanisms.

The Scalable Group Tag (SGT) eXchange Protocol (SXP) is one of several protocols that supports CTS. CTS SXP version 4 (SXPv4) enhances the functionality of SXP by adding a loop detection mechanism to prevent stale binding in the network. SXPv4 is an alternative SGT transport mechanism to inline tagging. It enables the propagation of security group bindings between network devices that do not support carrying the SGT in the CMD field of Ethernet frames (inline tagging).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Information About Cisco TrustSec with SXPv4

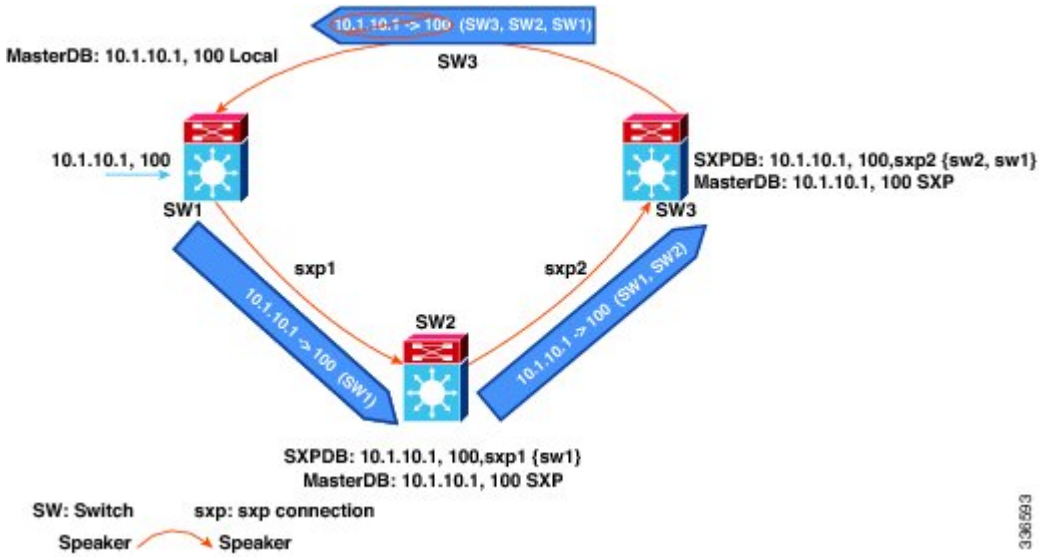
Overview of Cisco TrustSec with SXPv4

Cisco TrustSec (CTS) Scalable Group Tag (SGT) Exchange Protocol (SXP) (CTS-SXP) is a control plane protocol which propagates IP address to Security Group Tag (SGT) binding information across network devices. SGT is maintained by tagging packets (inline tagging) on ingress to the CTS-SXP network so that they can be properly identified for the purpose of applying security and other policy criteria along the data

path. 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.

SXP versions prior to version 4 required careful attention to SXP traffic flow. For example, in Figure 1, SXP traffic flows in one direction (access layer to data center) and from the data center to the distribution layer. This unidirectional traffic pattern is done on purpose because if SXP traffic were to flow in the opposite direction, an SXP loop could be created. SXP version 4 prevents a loop from occurring.

Figure 1: SXPv4 Loop Detection



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

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

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

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

SXP Node ID

An SXP node ID is used to identify the individual devices within the network. The node ID is a four-octet integer that can be configured by the user. If it is not configured by the user, SXP picks a node ID itself using the highest IPv4 address in the default VRF domain, in the same manner that EIGRP generates its node ID. The node ID has to be unique in the network that SXP connections traverse to enable SXP loop detection.

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.

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

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.

Note A syslog is generated when you change the node ID.

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:

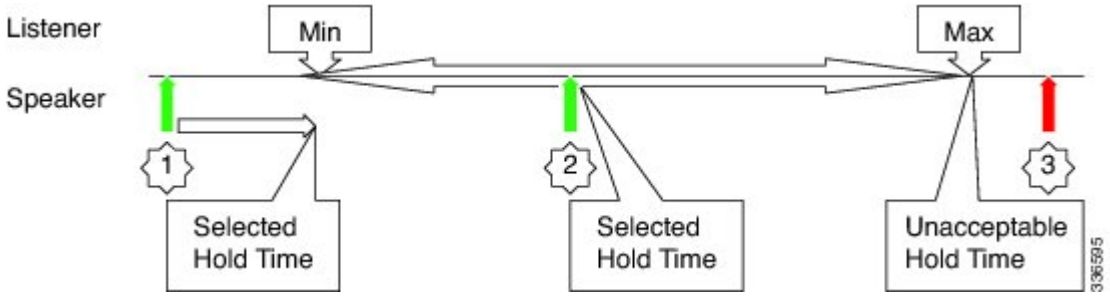
- **Hold timer**: Used by an SXP listener to detect when a connection is no longer live, that is, no KEEPALIVE or UPDATE message is received.
- **Keepalive timer**: Used by an SXP speaker to trigger the dispatch of keepalive messages during intervals when no other information is exported via update messages.

The hold-time for the keepalive mechanism may be negotiated during the open/open_resp exchange at connection setup. The following issues are 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.

Figure 2: Hold-time Negotiation Process



The figure above illustrates the hold-time negotiation process. More detail on the listener's and speaker''s roles is given below.

Connection Initiated by Listener

- A listener may include a hold-time attribute in the open message with minimum and maximum values set to its configured range of the hold-time period. A hold-time attribute with just a minimum value set to 0xFFFF0 would indicate to the speaker that the keepalive mechanism is not used.
- When a speaker receives an open message, it will react as follows:
  - If the hold-time attribute is not present or if it contains a minimum value that is set to 0xFFFF0, the speaker will set its keepalive time to 0xFFFF0 to indicate that the keepalive mechanism is disabled.
  - If the received hold-time attribute contains a valid range, the speaker must include a hold-time attribute in its open_resp message with a minimum value set as follows:
    - 0xFFFF0 if the speaker does not support the keepalive mechanism or if the mechanism is supported but disabled due to a local configuration, which sets the keepalive time to 0xFFFF0.
    - If the speaker's minimum acceptable hold-time value is greater than the upper bound of the offered range, the speaker must send an open error message with the subcode set to "Unacceptable hold-time" and terminate the connection. Otherwise the speaker will set the selected hold-time to the maximum of its minimum acceptable hold-time value and the lower bound of the offered hold-time range.
    - The speaker will calculate a new value for its keepalive time as one-third of that selected hold-time.
    - The speaker will set the minimum hold-time value of the hold-time attribute to the selected hold-time.
- When the listener receives the open_resp message from the speaker, it will look for hold-time attribute:
  - If the hold-time attribute is present and contains a minimum hold-time value of 0xFFFF0, the speaker will set its hold-time value to 0xFFFF0 to indicate that the keepalive mechanism is not used.
  - If the minimum hold-time value is within the range offered by the listener, the listener will set its hold-time period to the selected value it has received in the open_resp message.

- If the minimum hold-time value is outside the offered range, the listener will send an open error message with subcode set to “Unacceptable hold-time” and terminate the connection.

Connection Initiated by Speaker

- A speaker may include a hold-time attribute in the open message with minimum value set to its minimum acceptable hold-time period. A hold-time attribute with just a minimum value of 0xFFFF0 would indicate to the listener that the keepalive mechanism is not used.
- When a listener receives an open message, it will react as follows:
  - If the hold-time attribute is not present or if it contains a minimum value that is set to 0xFFFF0, the listener will set its hold-time to 0xFFFF0 to indicate that keepalive mechanism is disabled.
  - If the received hold-time attribute contains a valid value, the speaker must include hold-time attribute in its open_resp message with a minimum value set as follows:
    - 0xFFFF0 if the listener does not support the keepalive mechanism or if the mechanism is supported but disabled due to a local configuration, which sets the keepalive time to 0xFFFF0.
    - If the received hold-time value is greater than the upper bound of the listener’s configured hold-time range, the speaker must send an open error message with subcode set to “Unacceptable hold-time” and terminate the connection.
    - If the received hold-time value falls within the listener’s configured hold-time range, the listener will make it the selected hold-time.
    - If the received hold-time value is less than the lower bound of the listener’s configured hold-time range, the listener will set the selected hold-time to the lower bound of its hold-time range.
    - The listener will set the minimum hold-time value of the hold-time attribute to the selected hold-time.
- When the speaker receives the open_resp message from the listener, it will look for the hold-time attribute:
  - If the hold-time attribute is present and contains a minimum hold-time value of 0xFFFF0. The speaker will set its hold-time value to 0xFFFF0 to indicate that the keepalive mechanism is not used.
  - If the received hold-time value is greater or equal to the speaker's minimum acceptable hold-time, the speaker will calculate a new value for its keepalive time as one-third of the received hold-time.
  - If the received hold-time value is lower than the minimum acceptable, the speaker must send an open error message with subcode set to “Unacceptable hold-time” and terminate the connection.

How to Configure Cisco TrustSec with SXPv4

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

Hold-time can be configured globally on a network device, which applies to all SXP connections configured on the device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp listener hold-time** *minimum-period maximum-period*
4. **cts sxp speaker hold-time** *minimum-period*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp listener hold-time** *minimum-period maximum-period* **Example:** `Device(config)# cts sxp listener hold-time 750 1500`	Configures a minimum and maximum acceptable hold-time period in seconds for the listener device. The valid range is from 1 to 65534. The default hold-time range for a listener is 90 to 180 seconds. **Note** The *maximum-period* value must be greater than or equal to the *minimum-period* value.
Step 4	**cts sxp speaker hold-time** *minimum-period* **Example:** `Device(config)# cts sxp speaker hold-time 950`	Configures a minimum acceptable hold-time period in seconds for the speaker device. The valid range is 1 to 65534. The default hold-time for a speaker is 120 seconds.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **none**} **mode** {**local** | **peer**} [[**listener** | **speaker**] [**hold-time** *minimum-period maximum-period*] [**vrf** *vrf-name*]]
4. **exit**
5. **show cts sxp** {**connections** | **sgt-map**} [**brief** | **vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp connection peer** *ipv4-address* {**source** | **password**} {**default** | **none**} **mode** {**local** | **peer**} [[**listener** | **speaker**] [**hold-time** *minimum-period maximum-period*] [**vrf** *vrf-name*]] **Example:** `Device(config)# cts sxp connection peer 10.20.2.2 password default mode local speaker`	Configures the CTS-SXP peer address connection. 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: • **local**—The specified mode refers to the local device. • **peer**—The specified mode refers to the peer device. • **listener**—Specifies that the device is the listener in the connection. • **speaker**—Specifies that the device is the speaker in the connection. This is the default. The **hold-time** keyword allows you to specify the length of the hold-time period for the speaker or listener device. **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.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode.
Step 5	**show cts sxp** {**connections** | **sgt-map**} [**brief** | **vrf** *vrf-name*] **Example:** `Device# show cts sxp connections`	(Optional) Displays CTS-SXP status and connections.

Configuring the Node ID of a Network Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts sxp node-id** {*sxp-node-id* | **interface** *interface-type* | *ipv4-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**cts sxp node-id** {*sxp-node-id* | **interface** *interface-type* | *ipv4-address*} **Example:** `Device(config)# cts sxp node-id 172.16.1.3`	Configures the node ID of a network device.

Configuration Examples for Cisco TrustSec with SXPv4

Example: Configuring Cisco TrustSec with SXPv4

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

```
Device(config)# cts sxp speaker hold-time 950
```

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

```
Device(config)# cts sxp connection peer 10.20.2.2 password default mode local speaker
hold-time 500
```

Configuring the Node ID of a Network Device

```
Device(config)# cts sxp node-id 172.16.1.3
```

Verifying Cisco TrustSec with SXPv4

Display the SXP connections on a device

```
Device# show cts sxp connection

 SXP              : Enabled
 Highest Version Supported: 4
 Default Password : Set
 Default Source IP: Not Set
Connection retry open period: 120 secs
Reconcile period: 120 secs
Retry open timer is not running
----------------------------------------------
Peer IP          : 2.2.2.1
Source IP        : 2.2.2.2
Conn status      : On
Conn version     : 4
Conn capability  : IPv4-IPv6-Subnet
Conn hold time   : 0 seconds
Local mode       : SXP Listener
Connection inst# : 1
TCP conn fd      : 1
TCP conn password: default SXP password
Duration since last state change: 32:00:41:31 (dd:hr:mm:sec)


Total num of SXP Connections = 1
```

Displaying the current CST-SGT map database

In SXPv4, an SXP node ID is shown:

```
Device# show cts sxp sgt-map

SXP Node ID(generated):0x02020202(2.2.2.2)
IP-SGT Mappings as follows:
IPv4,SGT: <2.2.2.0/29 , 29>
source  : SXP;
Peer IP : 2.2.2.1;
Ins Num : 1;
Status  : Active;
Seq Num : 3
Peer Seq: 0B0B0B02,
IPv4,SGT: <12.12.133.1 , 12>
source  : SXP;
Peer IP : 2.2.2.1;
Ins Num : 1;
Status  : Active;
Seq Num : 5
Peer Seq: 0B0B0B02,
Total number of IP-SGT Mappings: 2
```

Displaying the Platform Specific CTS Information

CTS does not maintain separate send and receive counters for IPv4 and IPv6 traffic. Hence, the below show command displays the combined statistics for IPv4 and IPv6.

```
Device# show platform hardware qfp active feature cts datapath stats

Tagged Packets rcv: 1055       xmt: 1048       Def tag: 0
         Unknown SGT: 109677    Unknown DGT: 0
 Invalid tags (drop): 34        Bad format (drop): 0
 No xmt buffer: 0
 IPSec SGT tagged packets received: 0
 IPSec Invalid SGT tagged packets received: 0
 GRE SGT tagged packets received: 0
 GRE Invalid SGT tagged packets received: 0
 GRE invalid next protocol 0
 LISP SGT tagged packets received: 0
 LISP Invalid SGT tagged packets received: 0
 VXLAN SGT tagged packets received: 0
 VXLAN Invalid SGT tagged packets: 0
```

Additional References for Cisco TrustSec with SXPv4

Related Documents

Related Topic	Document Title
Security commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*

MIBs

MIB	MIBs Link
CISCO-TRUSTSEC-SXP-MIB	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for Cisco TrustSec with SXPv4

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Cisco TrustSec with SXPv4

Feature Name	Releases	Feature Information
Cisco TrustSec with SXPv4	Cisco IOS XE Release 3.9S	CTS SXP version 4 (SXPv4) enhances the functionality of SXP by adding a loop detection and prevention mechanism to prevent stale binding in the network. In addition, Cisco TrustSec with SXPv4 supports SGT inline tagging, which allows propagation of SGT embedded in clear-text (unencrypted) Ethernet packets. In Cisco IOS XE Release 3.9S, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced: **cts sxp listener hold-time**, **cts sxp node-id**, **cts sxp speaker hold-time**.