Cisco TrustSec VRF-Aware SGT

VRF-Aware SXP

The Security Group Tag (SGT) Exchange Protocol (SXP) implementation of Virtual Routing and Forwarding (VRF) binds an SXP connection with a specific VRF. It is assumed that the network topology is correctly configured for Layer 2 or Layer 3 VPNs, with all VRFs configured before enabling Cisco TrustSec.

SXP VRF support can be summarized as follows:

- Only one SXP connection can be bound to one VRF.
- Different VRFs may have overlapping SXP peer or source IP addresses.
- IP–SGT mappings learned (added or deleted) in one VRF can be updated only in the same VRF domain. The SXP connection cannot update a mapping bound to a different VRF. If no SXP connection exits for a VRF, IP–SGT mappings for that VRF won’t be updated by SXP.
- Multiple address families per VRF is supported. Therefore, one SXP connection in a VRF domain can forward both IPV4 and IPV6 IP-SGT mappings.
- SXP has no limitation on the number of connections and number of IP–SGT mappings per VRF.

IPv6 Support for VRF Aware SGT and SGACL

Beginning with the Cisco IOS XE Bengaluru 17.6.x release, IPv6 is supported for VRF Aware Security Group Tag (SGT) and SG Access Control List (SGACL). The feature extends for IPv6 the same functionality as for IPv4.

IPV6 support for SGT and SGACL feature enables the following features:

- SGT binding
  - Static binding between IPV6 addresses to SGT
  - VLAN-to-SGT bindings
  - Dynamic learning of mapping between IPv6 addresses and SGTs
- Enforcement
  - SGACL enforcement for IPV6 traffic based on UDP or TCP ports
  - SGACL enforcement for IPv6 traffic based on upper layer protocol type

Note

- SGT binding is not supported for link-local address.
- SGACL is not enforced on multicast traffic.

IPV6 SGT and SGACL scale numbers are the same for both IPv4 and IPv6, and most CLI commands are unchanged.

For more information about IPv6 support, see the following sections:

- How IPv4 and IPv6 Share SGT and SGACL Tables, on page 2
- SGT and SGACL Scale Numbers, on page 2

Also see the *Cisco TrustSec Configuration Guide, Cisco IOS XE 17* on Cisco.com.

How IPv4 and IPv6 Share SGT and SGACL Tables

IPv4 and IPv6 share SGT and SGACL tables in FPGA. The following list shows how that sharing is managed:

- If you enable either IPv4 *or* IPv6, it will use the entire table based on configurations.
- If you enable IPv4 *and* IPv6, the tables are shared based on which feature makes the first request.
- The appropriate syslog is generated if the SGT and SGACL tables are exceeded.
- The appropriate syslog is generated if you configure unsupported policies.

SGT and SGACL Scale Numbers

The following table shows scale numbers for IPv4 and IPv6.

Entry Type	Scale Number	Description
Host-SGT	1024	Host-to-SGT bind
Subnet-SGT	64	Network-to-SGT bind
SGT x DGT matrix	21 x 21	SGT-DGT mapping

Entry Type	Scale Number	Description
SGACL policy list size	15	Maximum ACE for each SGACL
Logging counters [31:0] Number of SGT and DGT pairs	32	Maximum pairs

Note By default, logging is enabled for only 32 SGT and DGT pairs. However, you can specify the pairs for which to enable logging. You can disable logging for any pairs among 32 and enable logging for different pairs.

- To see the SGT and DGT pairs for which logging is enabled, use the command `show platform hardware cts cell-logging`.
- To disable logging for specific SGT and DGT pairs, use the command `no platform cts logging`.
- To enable logging on specific SGT and DGT pairs, use the command `platform cts logging`.

The following text shows the options for the `no platform cts logging` command:

```
Device> enable
Device#configure terminal
Device(config)#no platform cts logging ?
all Disable logging for all the cells
default default logging list
from Source Group Tag (SGT) for enabling logging
```

How to Configure Cisco TrustSec VRF-Aware SGT

This section describes how to configure Cisco TrustSec VRF-Aware SGT.

Configuring VRF-to-SGT Mapping

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cts role-based sgt-map vrf** *vrf-name* {**ip4_netaddress** | *ipv6_netaddress* | **host** {*ip4_address* | *ip6_address*}}] **sgt** *sgt_number*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**cts role-based sgt-map vrf** *vrf-name* {**ip4_netaddress** | *ipv6_netaddress* | **host** {*ip4_address* | *ip6_address*}}] **sgt** *sgt_number*
	Example:
	`Device(config)# cts role-based sgt-map vrf red 10.0.0.3 sgt 23`
	Example:
	`Device(config)# cts role-based sgt-map vrf VRF_1 2405:201:c::f115 sgt 1201`
		Applies the SGT to packets in the specified VRF.
		The IP-SGT binding is entered into the IP-SGT table associated with the specified VRF and the IP protocol version implied by the type of IP address.

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for Cisco TrustSec VRF-Aware SGT

The following sections show configuration examples of Cisco TrustSec VRF-Aware SGT:

Example: Configuring VRF-to-SGT Mapping

IPv4 example:

```
Device> enable
Device# configure terminal
Device(config)# cts role-based sgt-map vrf VRF_1 22.1.1.1 sgt 1204
Device(config)# end
```

IPv6 example:

```
Device> enable
Device# configure terminal
Device(config)# cts role-based sgt-map vrf VRF_1 2405:201:c::f115 sgt 1201
Device(config)# end
```

Example: Role-based Access List Commands

```
Switch(config)# ipv6 access-list role-based acl-name
Switch(config-rb-acl)#?
Role-based Access List configuration commands:
```

```
<1-2147483647> Sequence Number
default    Set a command to its defaults
deny       Specify packets to reject
exit       Exit from access-list configuration mode
no         Negate a command or set its defaults
permit     Specify packets to forward
remark     Access list entry comment
Switch(config-rb-acl)#
```

ACE Port Ranges

Starting with the Cisco IOS XE Bengaluru 17.6.x release, the TrustSec FPGA module supports the port-range option in policy elements to address some scaling issues.

As part of TrustSec, the FPGA module maintains IP-to-SGT bindings and SGACL policies. Cisco IE3400 switches support 21 x 21 SGT or DGT pairs and 15 policies at each cell, matching on IP protocol field, the L4 source port, and L4 destination port.

However, given the match criteria, you may not be able to scale user access permission. So the TrustSec FPGA module now supports the port range option in each policy element by keeping supported policies to 15 for each cell.

The enhancement enables you to combine multiple rules as shown in the following list:

- Match on IP protocol field
- Match on L4 source start port and end port
- Match on L4 destination start port and end port

Example: Role-based Access List Commands for ACE Port Ranges

You can use the following commands to configureACE port ranges for source and destination ports:

```
Switch(config)# ip access-list role-based rbacl
Switch(config-rb-acl)#10 deny tcp dst range ftp-data telnet
Switch(config-rb-acl)#20 permit tcp dst lt 10
Switch(config-rb-acl)#30 deny tcp dst gt 50
```

Feature History for Cisco TrustSec VRF-Aware SGT

This table provides release and related 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	Feature Description
Cisco IOS XE Release 17.6.1	Extension of IPv6 support for SGT and SGACL	Enables host-to-SGT mapping and binding and subnet-to-SGT bindings.
	Extension of IPv6 support for SGACL enforcement	Enforces SGACL for IPV6 traffic based on UDP, TCP ports, and upper layer protocol type
	TrustSec FPGA module support for port-range option	Option is supported in policy elements to address scaling issues.
Cisco IOS XE Release 17.5.1	Cisco TrustSec VRF-Aware SGT	The Cisco TrustSec VRF-Aware SGT feature binds a SGT SXP connection with a specific VRF instance.