Configuring Tunnel Interfaces on Cisco IOS XR Software

This module describes the configuration of Tunnel-IPSec interfaces on the Cisco XR 12000 Series Routers.

Tunnel interfaces are virtual interfaces that provide encapsulation of arbitrary packets within another transport protocol. The Tunnel-IPSec interface provides secure communications over otherwise unprotected public routes.

A virtual interface represents a logical packet switching entity within the router. Virtual Interfaces have a global scope and do not have an associated location. The Cisco IOS XR Software uses the *rack/slot/module/port* notation for identifying physical interfaces, but uses a globally unique numerical ID after the interface name to identify virtual interfaces. Examples of this numerical ID are Loopback 0, Loopback 1, and Null99999. The ID is unique for each virtual interface type so you may simultaneously have a Loopback 0 and a Null 0.

Virtual interfaces have their control plane presence on the active route processor (RP). The configuration and control plane are mirrored onto the standby RP and, in the event of a switchover, the virtual interfaces will move to the standby, which then becomes the newly active RP.

Note Subinterfaces can be physical or virtual, depending on their parent interface.

Virtual tunnels are *configured* on any RP or distributed RP (DRP), but they are created and operate only from the RP.

Note Tunnels do not have a one-to-one modular services card association.

Feature History for Configuring Tunnel Interfaces on Cisco IOS XR Software

Release	Modification
Release 3.2	This feature was introduced on the Cisco XR 12000 Series Router.

Contents

Prerequisites for Configuring Tunnel Interfaces

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Information About Configuring Tunnel Interfaces

To implement tunnel interfaces, you must understand the following concepts:

Tunnel Interfaces Overview

Tunneling provides a way to encapsulate arbitrary packets inside of a transport protocol. This feature is implemented as a virtual interface to provide a simple interface for configuration. The tunnel interfaces are not tied to specific “passenger” or “transport” protocols, but, rather, they represent an architecture that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme. Because supported tunnels are point-to-point links, you must configure a separate tunnel for each link.

There are three necessary steps in configuring a tunnel interface:

1. Specify the tunnel interface—**interface tunnel-ipsec** *identifier*.
2. Configure the tunnel source—**tunnel source** {*ip-address* | *interface-id*}.
3. Configure the tunnel destination—**tunnel destination** {*ip-address* | *tunnel-id*}.

Virtual Interface Naming Convention

Virtual interface names never use the physical interface naming notation *rack/slot/module/port* for identifying an interface’s rack, slot, module, and port, because they are not tied to any physical interface or subinterface.

Virtual interfaces use a globally unique numerical identifier (per virtual interface type).

Examples of naming notation for virtual interfaces:

```
Interface                      IP-Address      Status Protocol
Loopback0                      10.9.0.0         Up           Up
Loopback10                     10.7.0.0         Up           Up
Tunnel-TE5000                  172.18.189.38    Down         Down
Null10                         10.8.0.0          Up           Up
```

Tunnel-IPSec Overview

IPSec (IP security) is a framework of open standards for ensuring secure private communications over the Internet. It can be used to support Virtual Private Network (VPN), firewalls, and other applications that must transfer data across a public or insecure network. The router IPSec protocol suite provides a set of standards that are used to provide privacy, integrity, and authentication service at the IP layer. The IPSec protocol suite also includes cryptographic techniques to support the key management requirements of the network-layer security.

When IPSec is used, there is no need to use Secure Shell (SSH) or Secure Socket Layer (SSL). Their use causes the same data to be encrypted or decrypted twice, which creates unnecessary overhead. The IPSec daemon is running on both the RPs and the DRPs. IPSec is an optional feature on the router. IPSec is a good choice for a user who has multiple applications that require secure transport. On the client side, customers can use “Cisco VPN 3000 Client” or any other third-party IPSec client software to build IPSec VPN.

Note IPSec tunnel exists in the control plane, so you do not have to bring up or bring down the tunnel. Entry into the IPSec tunnel is only for locally sourced traffic from the RP or DRP, and is dictated by the access control lists (ACL) configured as a part of the profile that is applied to the Tunnel-IPSec.

Tunnel-IPSec Naming Convention

A profile is entered from interface configuration submode for interface tunnel-ipsec. For example:

```
interface tunnel-ipsec 30
     profile <profile name>
```

Crypto Profile Sets

Crypto profile sets must be configured and applied to tunnel interfaces (or to the crypto IPSec transport). For details on using the crypto IPSec transport, refer to the link provided in the “Additional References” section on page 577. For IPSec to succeed between two IPSec peers, the crypto profile entries of both peers must contain compatible configuration statements.

Two peers that try to establish a security association must each have at least one crypto profile entry that is compatible with one of the other peer's crypto profile entries. For two crypto profile entries to be compatible, they must at least meet the following criteria:

- They must contain compatible crypto access lists. In the case where the responding peer is using dynamic crypto profiles, the entries in the local crypto access list must be “permitted” by the peer's crypto access list.
- They must each identify the other peer (unless the responding peer is using dynamic crypto profiles).
- They must have at least one transform set in common.

Note Crypto profiles cannot be shared; that is, the same profile cannot be attached to multiple interfaces.

How to Configure Tunnel Interfaces

This section contains the following procedures:

- Configuring Tunnel-IPSec Interfaces, page 574 (Required)

Configuring Tunnel-IPSec Interfaces

This task explains how to configure Tunnel-IPSec interfaces.

Prerequisites

To use the **profile** command, you must be in a user group associated with a task group that includes the proper task IDs for crypto commands. To use the **tunnel destination** command, you must be in a user group associated with a task group that includes the proper task IDs for interface commands.

For detailed information about user groups and task IDs, see the *Configuring AAA Services on Cisco IOS XR Software* module of *Cisco IOS XR System Security Configuration Guide*.

The following tasks are required for creating Tunnel-IPSec interfaces:

- Setting Global Lifetimes for IPSec Security Associations
- Configuring Checkpointing
- Configuring Crypto Profiles

For detailed information on configuring the prerequisite checkpointing and crypto profiles, and setting the global lifetimes for IPSec security associations, refer to the *Implementing IPSec Network Security on Cisco IOS XR Software* module in *Cisco IOS XR System Security Configuration Guide*.

After configuring crypto profiles, you must apply a crypto profile to each tunnel interface through which IPSec traffic will flow. Applying the crypto profile set to a tunnel interface instructs the router to evaluate all the interface's traffic against the crypto profile set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by crypto.

SUMMARY STEPS

1. **configure**
2. **interface tunnel-ipsec** *identifier*
3. **profile** *profile-name*
4. **tunnel source** {*ip-address* | *interface-id*}
5. **tunnel destination** {*ip-address* | *tunnel-id*}
6. **end**
   or
   commit
7. **show ip route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface tunnel-ipsec** *identifier* **Example:** `RP/0/RP0/CPU0:router(config)# interface tunnel-ipsec 30`	Identifies the IPSec interface to which the crypto profile will be attached and enters interface configuration mode.
Step 3	**profile** *profile-name* **Example:** `RP/0/RP0/CPU0:router(config-if)# profile user1`	Assigns the crypto profile name to be applied to the tunnel for IPSec processing. • The same crypto profile cannot be shared in different IPSec modes.
Step 4	**tunnel source** (*ip-address* | *interface-id*) **Example:** `RP/0/RP0/CPU0:router(config-if)# tunnel source Ethernet0/1/1/2`	Specifies the tunnel source IP address or interface ID. • This command is required for both static and dynamic profiles.
Step 5	**tunnel destination** {*ip-address* | *tunnel-id*} **Example:** `RP/0/RP0/CPU0:router(config-if)# tunnel destination 192.168.164.19`	(Optional) Specifies the tunnel destination IP address. • This command is not required if the profile is dynamic.

	Command or Action	Purpose

Step 6	**end**
	or
	commit
	
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
	or
	`RP/0/RP0/CPU0:router(config-if)# commit`
		Saves configuration changes.
		• When you issue the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		– Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
		– Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
		– Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.
		• Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Step 7	**show ip route** **Example:** `RP/0/RP0/CPU0:router# show ip route`	Displays forwarding information for the tunnel. • The command **show ip route** displays what was advertised and shows the routes for static and autoroute.

Configuration Examples for Tunnel Interfaces

This section contains the following example:

Tunnel-IPSec: Example, page 576

Tunnel-IPSec: Example

This example shows the process of creating and applying a profile to an IPSec tunnel. The necessary preliminary steps are also shown. You must first define a transform set and then create a profile before configuring the IPSec tunnel.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ipsec transform-set tset1
RP/0/RP0/CPU0:router(config-transform-set tset1)# transform esp-sha-hmac
RP/0/RP0/CPU0:router(config-transform-set tset1)# end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: yes

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# crypto ipsec profile user1
RP/0/RP0/CPU0:router(config-user1)# match sampleac1 transform-set tset1
RP/0/RP0/CPU0:router(config-user1)# set pfs group5
RP/0/RP0/CPU0:router(config-user1)# set type dynamic
RP/0/RP0/CPU0:router(config-user1)# exit
```

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface tunnel-ipsec 30
RP/0/RP0/CPU0:router(config-if)# profile user1
RP/0/RP0/CPU0:router(config-if)# tunnel source MgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:router(config-if)# tunnel destination 192.168.164.19
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: yes
```

Where to Go Next

You now must apply a crypto profile to each transport. Applying the crypto profile set to a transport instructs the router to evaluate all the interface's traffic against the crypto profile set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by crypto.

For information on applying a crypto profile to each transport, see the *Implementing IPSec Network Security on Cisco IOS XR Software* module of *Cisco IOS XR System Security Configuration Guide.*

Additional References

These sections provide references related to tunnel interface configuration.

Related Documents

Related Topic	Document Title
Cisco IOS XR master command reference	*Cisco IOS XR Master Commands List*
Cisco IOS XR interface configuration commands	*Cisco IOS XR Interface and Hardware Component Command Reference*
Information about IPSec and crypto profiles	*Cisco IOS XR System Security Configuration Guide*
Information about MPLS Traffic Engineering, including configuring a tunnel interface for MPLS-TE	*Cisco IOS XR Multiprotocol Label Switching Configuration Guide*
Information about user groups and task IDs	*Cisco IOS XR Interface and Hardware Component Command Reference*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
There are no applicable MIBs for this module.	To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/support