Crypto Conditional Debug Support

Last Updated: October 20, 2011

The Crypto Conditional Debug Support feature introduces three new command-line interfaces (CLIs) that allow users to debug an IP Security (IPSec) tunnel on the basis of predefined crypto conditions such as the peer IP address, connection-ID of a crypto engine, and security parameter index (SPI). By limiting debug messages to specific IPSec operations and reducing the amount of debug output, users can better troubleshoot a router with a large number of tunnels.

Feature History for Crypto Conditional Debug Support

Feature History	
Release	Modification
12.3(2)T	This feature was introduced.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see 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 at the end of this document.

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.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Crypto Conditional Debug Support

To use the new crypto CLIs, you must be using a crypto image such as the k8 or k9 subsystem.

Restrictions for Crypto Conditional Debug Support

- This feature does not support debug message filtering for hardware crypto engines.
- Although conditional debugging is useful for troubleshooting peer-specific or functionality related Internet Key Exchange (IKE) and IPSec problems, conditional debugging may not be able to define and check large numbers of debug conditions.
- Because extra space is needed to store the debug condition values, additional processing overhead is added to the CPU and memory usage is increased. Thus, enabling crypto conditional debugging on a router with heavy traffic should be used with caution.

Information About Crypto Conditional Debug Support

Supported Condition Types

The new crypto conditional debug CLIs-- debug crypto condition , debug crypto condition unmatched , and show crypto debug-condition --allow you to specify conditions (filter values) in which to generate and display debug messages related only to the specified conditions. The table below lists the supported condition types.

Table 1 Supported Condition Types for Crypto Debug CLI

Condition Type (Keyword)	Description
connid [1]	An integer between 1-32766. Relevant debug messages will be shown if the current IPSec operation uses this value as the connection ID to interface with the crypto engine.
flowid 1	An integer between 1-32766. Relevant debug messages will be shown if the current IPSec operation uses this value as the flow-ID to interface with the crypto engine.

1 If an IPSec connid, flowid, or SPI is used as a debug condition, the debug messages for a related IPSec flow are generated. An IPSec flow has two connids, flowids, and SPIs--one inbound and one outbound. Both two connids, flowids, and SPIs can be used as the debug condition that triggers debug messages for the IPSec flow.

Condition Type (Keyword)	Description
FVRF	The name string of a virtual private network (VPN) routing and forwarding (VRF) instance. Relevant debug messages will be shown if the current IPSec operation uses this VRF instance as its front-door VRF (FVRF).
IVRF	The name string of a VRF instance. Relevant debug messages will be shown if the current IPSec operation uses this VRF instance as its inside VRF (IVRF).
peer group	A Unity group-name string. Relevant debug messages will be shown if the peer is using this group name as its identity.
peer hostname	A fully qualified domain name (FQDN) string. Relevant debug messages will be shown if the peer is using this string as its identity; for example, if the peer is enabling IKE Xauth with this FQDN string.
peer ipaddress	A single IP address. Relevant debug messages will be shown if the current IPSec operation is related to the IP address of this peer.
peer subnet	A subnet and a subnet mask that specify a range of peer IP addresses. Relevant debug messages will be shown if the IP address of the current IPSec peer falls into the specified subnet range.
peer username	A username string. Relevant debug messages will be shown if the peer is using this username as its identity; for example, if the peer is enabling IKE Extended Authentication (Xauth) with this username.
SPI 1	A 32-bit unsigned integer. Relevant debug messages will be shown if the current IPSec operation uses this value as the SPI.

How to Enable Crypto Conditional Debug Support

Enabling Crypto Conditional Debug Messages

Performance Considerations

- Before enabling crypto conditional debugging, you must decide what debug condition types (also known as debug filters) and values will be used. The volume of debug messages is dependent on the number of conditions you define.

Note Specifying numerous debug conditions may consume CPU cycles and negatively affect router performance.

- Your router will perform conditional debugging only after at least one of the global crypto debug commands--**debug crypto isakmp**, **debug crypto ipsec**, and **debug crypto engine**--has been enabled. This requirement helps to ensure that the performance of the router will not be impacted when conditional debugging is not being used.

Disable Crypto Debug Conditions

If you choose to disable crypto conditional debugging, you must first disable any crypto global debug CLIs you have issued ; thereafter, you can disable conditional debugging.

Note The **reset** keyword can be used to disable all configured conditions at one time.

SUMMARY STEPS

1. **enable**
2. **debug crypto condition** [**connid***integer***engine-id***integer*] [**flowid***integer* **engine-id***integer*] [**fvrf** *string*] [**ivrf** *string*] [**peer** [**group** *string*] [**hostname** *string*] [**ipv4** *ipaddress*] [**subnet** *subnet mask*] [**username** *string*]] [**spi** *integer*] [**reset**]
3. **show crypto debug-condition** {[**peer**] [**connid**] [**spi**] [**fvrf**] [**ivrf**] [**unmatched**]}
4. **debug crypto isakmp**
5. **debug crypto ipsec**
6. **debug crypto engine**
7. debug crypto condition **unmatched** [**isakmp** | **ipsec** | **engine**]

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**debug crypto condition** [**connid***integer***engine-id***integer*] [**flowid***integer* **engine-id***integer*] [**fvrf** *string*] [**ivrf** *string*] [**peer** [**group** *string*] [**hostname** *string*] [**ipv4** *ipaddress*] [**subnet** *subnet mask*] [**username** *string*]] [**spi** *integer*] [**reset**] **Example:** `Router# debug crypto condition connid 2000 engine-id 1`	Defines conditional debug filters.
Step 3	**show crypto debug-condition** {[**peer**] [**connid**] [**spi**] [**fvrf**] [**ivrf**] [**unmatched**]} **Example:** `Router# show crypto debug-condition spi`	Displays crypto debug conditions that have already been enabled in the router.

Step 4	**debug crypto isakmp**
	Example:
	`Router#`
	`debug crypto isakmp`
		Enables global IKE debugging.

Step 5	**debug crypto ipsec**
	Example:
	`Router#`
	`debug crypto ipsec`
		Enables global IPSec debugging.

Step 6	**debug crypto engine**
	Example:
	`Router#`
	`debug crypto engine`
		Enables global crypto engine debugging.

Step 7	debug crypto condition **unmatched** [**isakmp** | **ipsec** | **engine**] **Example:** `Router# debug crypto condition unmatched ipsec`	(Optional) Displays debug conditional crypto messages when no context information is available to check against debug conditions. If none of the optional keywords are specified, all crypto-related information will be shown.

Enabling Crypto Error Debug Messages

To enable crypto error debug messages, you must perform the following tasks.

- debug crypto error CLI, page 6

debug crypto error CLI

Enabling the **debug crypto error** command displays only error-related debug messages, thereby, allowing you to easily determine why a crypto operation, such as an IKE negotiation, has failed within your system.

Note When enabling this command, ensure that global crypto debug commands are not enabled; otherwise, the global commands will override any possible error-related debug messages.

SUMMARY STEPS

1. **enable**
2. **debug crypto** {**isakmp** | **ipsec** | **engine**} **error**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto** {**isakmp**	**ipsec**	**engine**} **error** **Example:** `Router# debug crypto ipsec error`	Enables only error debugging messages for a crypto area.

Configuration Examples for the Crypto Conditional Debug CLIs

Enabling Crypto Conditional Debugging Example

The following example shows how to display debug messages when the peer IP address is 10.1.1.1, 10.1.1.2, or 10.1.1.3, and when the connection-ID 2000 of crypto engine 0 is used. This example also shows how to enable global debug crypto CLIs and enable the **show crypto debug-condition** command to verify conditional settings.

```
Router#
debug crypto condition connid 2000 engine-id 1
Router#
debug crypto condition peer ipv4 10.1.1.1
Router#
debug crypto condition peer ipv4 10.1.1.2
```

```
Router#
debug crypto condition peer ipv4 10.1.1.3
Router#
debug crypto condition unmatched
! Verify crypto conditional settings.
Router#
show crypto debug-condition
Crypto conditional debug currently is turned ON
IKE debug context unmatched flag:ON
IPsec debug context unmatched flag:ON
Crypto Engine debug context unmatched flag:ON
IKE peer IP address filters:
10.1.1.1  10.1.1.2   10.1.1.3
Connection-id filters:[connid:engine_id]2000:1,
! Enable global crypto CLIs to start conditional debugging.
Router#
debug crypto isakmp
Router#
debug crypto ipsec
Router#
debug crypto engine
```

Disabling Crypto Conditional Debugging Example

The following example shows how to disable all crypto conditional settings and verify that those settings have been disabled:

```
Router#
debug crypto condition reset
! Verify that all crypto conditional settings have been disabled.
Router#
show crypto debug-condition
Crypto conditional debug currently is turned OFF
IKE debug context unmatched flag:OFF
IPsec debug context unmatched flag:OFF
Crypto Engine debug context unmatched flag:OFF
```

Additional References

The following sections provide references to the Crypto Conditional Debug Support feature.

Related Documents

Related Topic	Document Title
IPSec and IKE configuration tasks	"Internet Key Exchange for IPsec VPNs" section of *Cisco IOS Security Configuration Guide: Secure Connectivity*
IPSec and IKE commands	*Cisco IOS Security Command Reference*

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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/techsupport