Security Configuration Guide: Unified Threat Defense, Cisco IOS XE 17

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on standards documentation, or language that is used by a referenced third-party product.

CONTENTS

CHAPTER 1

CHAPTER 1

Cisco Firepower Threat Defense for ISR

Cisco Firepower Threat Defense is Cisco's premier network security option. It provides a comprehensive suite of security features, such as firewall capabilities, monitoring, alerts, and Intrusion Detection System (IDS)

This module describes how to configure and deploy IDS on Cisco Integrated Services Routers (ISRs).

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.

Restrictions for Cisco Firepower Threat Defense for ISR

- Multicast traffic is not inspected.
- IPv6 traffic cannot be exported.

Information About Cisco Firepower Threat Defense for ISR

Cisco Firepower Threat Defense for ISR Overview

Cisco Firepower Threat Defense is a premier security solution that provides enhanced inspection for packet flows.

The Cisco Firepower Threat Defense solution consists of the following two entities:

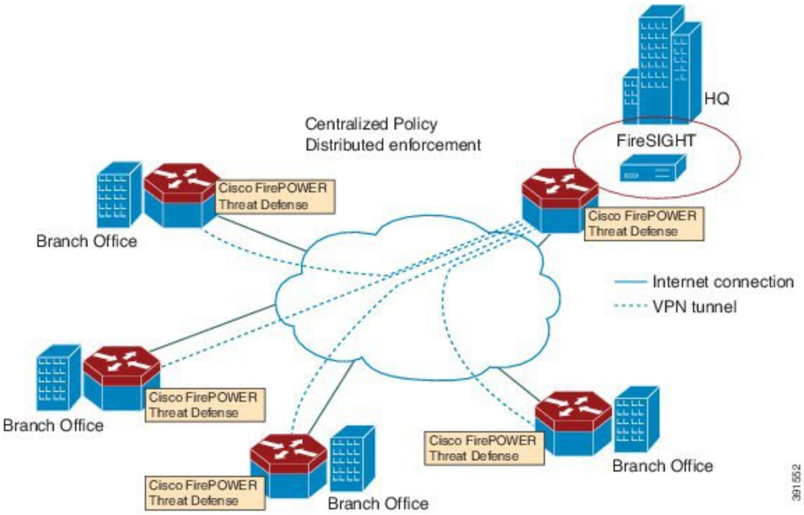
- Cisco FireSIGHT—A centralized policy and reporting entity that can run anywhere in the network. This can be the Cisco FireSIGHT appliance or a virtual installation on a server class machine.
- Virtual Firepower sensor—Security entities that implement policies, and send events and statistics back to the defense center. The Firepower sensor is hosted on Cisco Unified Computing System (UCS) E-Series Blade. Both the FireSIGHT and sensor are distributed as virtual packages.

UCS E-Series Blades are general purpose blade servers that are housed within Cisco Integrated Services Routers (ISR) Generation 2 (G2) and Cisco ISR 4000 Series Integrated Services Routers. These blades can be deployed either as bare-metal on operating systems or as virtual machines on hypervisors. There are two internal interfaces that connect a router to an UCS E-Series Blade. On ISR G2, Slot0 is a Peripheral Component Interconnet Express (PCIe) internal interface, and UCS E-Series Slot1 is a switched interface connected to the backplane Multi Gigabit Fabric (MGF). In Cisco ISR 4000 Series Routers, both internal interfaces are connected to the MGF.

A hypervisor is installed on the UCS E-Series Blade, and Cisco Firepower Threat Defense runs as a virtual machine on it. The Cisco Firepower Threat Defense OVA file is directly installed on the UCS E-Series Blade using the hypervisor operating system. Cisco Firepower Threat Defense runs as an anonymous inline device with no additional communication with the router. Traffic is diverted from the ingress physical interface to the Cisco Firepower Threat Defense that runs on the UCS E-Series Blade.

The following figure shows a Cisco Firepower Threat Defense deployment scenario. In this figure, the traffic lines between sensors and FireSIGHT are control connections. Packets are routed through these connections using router forwarding rules.

Figure 1: Cisco Firepower Threat Defense Deployment Scenario


HQ
Centralized Policy
Distributed enforcement
FireSIGHT
Cisco FirePOWER Threat Defense
Branch Office
Cisco FirePOWER Threat Defense
Internet connection
VPN tunnel
Branch Office
Cisco FirePOWER Threat Defense
Cisco FirePOWER Threat Defense
Branch Office
Cisco FirePOWER Threat Defense
Branch Office
391552

By default, the virtualized Cisco Firepower sensor comes with three interfaces, one for management, and two others for traffic analysis. These interfaces must be mapped to the UCS E-Series interfaces.

UCS-Based Hosting

The Cisco Unified Computing System (UCS) E-Series Blade provides a generic server blade for hosting applications. This blade typically runs VMware ESXi hypervisor and is managed through vSphere like other VMWare deployments.

If the Firepower sensor is hosted on the Cisco UCS E-Series Blade, you must specify the Cisco IOS interfaces connected to Cisco Firepower Threat Defense. Applications running within the UCS E-Series Blade are only loosely coupled with Cisco IOS, and to determine the interfaces that are attached to appliances a mapping of the interfaces must be done. Interfaces to connect to the Cisco UCS E-Series Blade are Bridge Domain Interfaces (BDI).

The following Cisco UCS E-Series Blades are supported for hosting the Firepower sensor:

- UCS-E 120S
- UCS-E 140D
- UCS-E 140S
- UCS-E 160D
- UCS-E 180D

IDS Packet Flow in Cisco Firepower Threat Defense

Cisco Firepower Threat Defense supports Intrusion Detection System (IDS). In IDS mode, traffic is copied to the sensor and is analyzed for threats. IDS mode cannot enforce policies; it can detect and report violations. In IDS mode, traffic is replicated from interfaces and redirected to Cisco Firepower Threat Defense that runs on the Cisco UCS E-Series blade.

IDS copies the traffic and analyzes them for threats. Enable the **utd** command to replicate packets to the Firepower sensor based on one of the following criteria:

- If global inspection is enabled, all packets that flow through a router are replicated to the sensor.
- If per interface inspection is enabled, packets are replicated only if the input or output interface has enabled the **utd** command for inspection.

To view the interfaces that have enabled packet inspection in IDS mode, use the **show platform software utd interfaces** command. The packet replication occurs as one of the first output features.

For general packet processing, features that are applied to a packet form an ordered sequence that is determined by the configuration of the device. In general, these features are grouped as either input or output features, with the routing function marking the boundary between the two. The IDS packet replication occurs as one of the first output features and so if any input feature drops the packet, it will not be replicated to the IDS engine.

Firepower Sensor Interfaces

The Firepower sensor virtual appliance has three network interfaces—two for analyzing the traffic and one for management connectivity to FireSIGHT. The two traffic-bearing interfaces are represented as two virtual interfaces; Bridge Domain Interfaces (BDIs), in the configuration.

Although two interfaces are available for analyzing the traffic, only one traffic-bearing interface is used for Intrusion Detection System (IDS).

The Firepower sensor is connected to the management network and appears as another host on the LAN segment.

Note To monitor VLAN traffic in your virtual environment, set the VLAN ID of the promiscuous port to 4095.

Cisco Firepower Threat Defense Interoperability

Cisco Firepower Threat Defense supports Intrusion Detection System (IDS). In IDS mode, selected traffic is copied to the Firepower sensor for analysis.

Cisco Firepower Threat Defense interoperates with the following features:

- Zone-based firewall—Application layer gateways (ALGs), application inspection and controls (AICs), and policies configured between zones
- Network Address Translation (NAT)

Note Cisco Firepower Threat Defense does not support outside address translation, because there is no mechanism to inform Firepower Threat Defense about outside global addresses. However; you can still enable address translation on outside interfaces. Intrusion Prevention System (IPS) or IDS is invoked after NAT on the ingress interface, and before NAT on the egress interface, always using inside addresses.

- Crypto
- Intelligent WAN (IWAN)
- Kernel-based Virtual Machine Wide-Area Application Services (kWAAS)

Hardware and Software Requirements for Cisco Firepower Threat Defense

The following hardware is required to run the Cisco Firepower Threat Defense solution:

- Cisco Firepower Sensor version 5.4
- Cisco Integrated Services Routers (ISR) 4000 Series Routers
- Cisco Unified Computing System (UCS) E-Series Blade
- Cisco FireSIGHT

The following software is required to run the Cisco Firepower Threat Defense solution:

- UCS-E hypervisor
- ESXi 5.0.0, 5.1.0, or 5.5.0
- Cisco Firepower Sensor version Cisco IOS XE Release 3.14S and later releases
- Cisco FireSIGHT version 5.2, 5.3 or 5.4. FireSIGHT only supports the current version and is backward compatible with only the previous version. In case, your Cisco Firepower Sensor version is 5.4, then you have to use FireSIGHT version 5.4 or 5.3.

Obtaining Cisco Firepower Threat Defense License

Cisco ISR 4000 Series Integrated Services Routers must have the security K9 license and Application Experience (AppX) license to enable the Cisco Firepower Threat Defense.

```
Technology Package License Information:
-----------------------------------------------------------------
Technology    Technology-package           Technology-package
              Current       Type           Next reboot
------------------------------------------------------------------
appx             appxk9           EvalRightToUse   appxk9
uc               uck9             EvalRightToUse   uck9
security         securityk9       EvalRightToUse   securityk9
ipbase           ipbasek9         Permanent        ipbasek9
```

How to Deploy Cisco Firepower Threat Defense for ISR

To deploy Cisco Firepower Threat Defense Intrusion Detection System (IDS), perform the following tasks:

1. Obtain the Firepower sensor package.
2. Install the Firepower sensor package through a hypervisor, such as VMWare VSphere.
3. Configure router interfaces for traffic redirection.
    - Bridge-Domain interface (BDI) configuration for Cisco ISR 4000 Series Routers.
    - VLAN configuration for Cisco ISR Generation 2 routers.
4. Bootstrap the Firepower sensor.
5. Configure a policy in Cisco FireSIGHT.

- The policy is configured through the FireSIGHT GUI.

6. Enable inspection.

Obtaining the Firepower Sensor Package

To deploy the Firepower sensor on an Unified Computing System (UCS) E-Series Blade, download and save the OVA file. OVA is an Open Virtualization Archive that contains a compressed and installable version of a virtual machine. Download the OVA file from https://support.sourcefire.com/sections/1/sub_sections/51#5-2-virtual-appliances.

Installing the Firepower Sensor OVA File

Install the Firepower Sensor OVA on a UCS E-Series Blade, using a hypervisor, such as VMWare VSphere.

Installing Firepower Sensor on a UCS E-Series Blade

This section describes how to install the Firepower Sensor on a Unified Computing System (UCS) E-Series Blade that is installed on Cisco ISR 4000 Series Integrated Services Routers:

1. Install the UCS E-Series card.
2. Verify that the card is running by using the **show platform** command.
3. Configure the Cisco Integrated Management Controller (CIMC) port.

   The CIMC GUI is a web-based management interface for E-Series Servers. You can launch the CIMC GUI to manage the server from any remote host that meets the following minimum requirements:

   - Java 1.6 or later
   - HTTP or HTTPS-enabled
   - Adobe Flash Player 10 or later

   The CIMC runs on the port that is named management. The following example shows how to bootstrap the management port with an IP address:

   ```
   ucse subslot 1/0
     imc access-port dedicated
     imc ip-address 10.66.152.158 255.255.255.0
   !
   ```

   Connect to the CIMC through the browser by using the default login and password, which are admin and password, respectively. Based on the configuration example, the browser address is https://10.66.152.158.

4. Install ESXi.

   Download the ESXi image for your Cisco UCS E-Series Blade from https://my.vmware.com/web/vmware/details?downloadGroup=CISCO-ESXI-5.1.0-GA-25SEP2012&productId=284.

5. Install Firepower Sensor by using VMWare VSphere on the Cisco UCS E-Series blade.
6. Configure traffic redirect. For more information, see the section “Configuring Traffic Redirect on Cisco UCS E-Series Blade”.
7. Configure the VMWare vSwitch. The Virtual Machine Network Interface Card (VMNIC) mapping on ISR 4000 Series Routers is as follows:

   - VMNIC0—Mapped to UCS E-Series interface x/0/0 on the router backplane
   - VMNIC1—Mapped to UCS E-Series interface x/0/1 on the router backplane

- VMNIC2—Mapped to UCS E-Series frontplane GigabitEthernet 2 interface.
- VMNIC3—Mapped to UCS E-Series frontplane GigabitEthernet 3 interface.

Note VMNIC3 is only available on UCS E-Series 140D, 160Dm and 180D.

UCS E-Series 120S and 140S have 3 network adaptors and one management port. UCS E-Series 140D, 160Dm and 180D have 4 network adaptors.

Configuring Traffic Redirect on Cisco UCS E-Series Blade

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ip address**
5. **no negotiation auto**
6. **switchport mode trunk**
7. **no mop enabled**
8. **no mop sysid**
9. **service instance** *service-instance-number ethernet*
10. **encapsulation dot1q** *vlan-id*
11. **rewrite ingress tag pop** {**1** | **2**} **symmetric**
12. **bridge domain** *bridge-ID*
13. **end**

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	**interface** *type number* **Example:** `Router(config)# interface ucse 1/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**no ip address** **Example:**	Removes an IP address or disables IP processing on an interface.

	Command or Action	Purpose
	`Router(config-if)# no ip address`	
Step 5	**no negotiation auto** **Example:** `Router(config-if)# no negotiation auto`	Disables advertisement of speed, duplex mode, and flow control on an interface.
Step 6	**switchport mode trunk** **Example:** `Router(config-if)# switchport mode trunk`	Specifies a trunking VLAN Layer 2 interface.
Step 7	**no mop enabled** **Example:** `Router(config-if)# no mop enabled`	Disables the Maintenance Operation Protocol (MOP) on an interface.
Step 8	**no mop sysid** **Example:** `Router(config-if)# no mop sysid`	Disables the sending of periodic MOP system identification messages from an interface.
Step 9	**service instance** *service-instance-number ethernet* **Example:** `Router(config-if)# service instance 10 ethernet`	Configures an Ethernet service instance on an interface and enters Ethernet service-instance configuration mode.
Step 10	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 10`	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance.
Step 11	**rewrite ingress tag pop** {**1** | **2**} **symmetric** **Example:** `Router(config-if-srv)# rewrite ingress tag pop 1 symmetric`	Specifies the encapsulation adjustment to be performed on a frame ingressing a service instance.
Step 12	**bridge domain** *bridge-ID* **Example:** `Router(config-if-srv)# bridge domain 10`	Binds a service instance or a MAC tunnel to a bridge domain instance.

Step 13	**end**
	Example:
	`Router(config-if)# end`
		Exits Ethernet service-instance configuration mode and returns to privileged EXEC configuration mode.

Bootstrapping the Firepower Sensor

You must configure the Firepower Sensor manually. Perform this task to configure a Firepower sensor to communicate with FireSIGHT. For more information, see https://support.sourcefire.com/sections/10.

A sensor running on a Cisco Unified Computing System (UCS) E-Series Blade is bootstrapped by logging into the console of the Firepower Sensor virtual machine through VSphere.

Note Firepower Sensor must be installed and deployed before bootstrapping it.

SUMMARY STEPS

1. Provide the default username and password to login.
2. **configure network ipv4 manual** *ip-address network-mask default-gateway*
3. **configure network dns servers** *dns-server*
4. **configure network dns searchdomains** *domain-name*
5. **configure manager add** *dc-hostname registration-key*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Provide the default username and password to login.	To configure the sensor, the default username and password are admin and Sourcefire, respectively. • You must change the admin password after you login to the Firepower Sensor the first time.
Step 2	**configure network ipv4 manual** *ip-address network-mask default-gateway* **Example:** `Device# configure network ipv4 manual 10.66.152.137 255.255.255.0 10.66.152.1`	Configures network connectivity.
Step 3	**configure network dns servers** *dns-server* **Example:** `Device# configure network dns servers 192.10.26.10`	Configures domain name system (DNS) servers.
Step 4	**configure network dns searchdomains** *domain-name* **Example:** `Device# configure network dns searchdomains cisco.com`	Configures DNS search domains.
Step 5	**configure manager add** *dc-hostname registration-key* **Example:** `Device# configure manager sourcefire-dc.cisco.com cisco-sf`	Associates the sensor with the FireSIGHT. • The *registration key* is a string selected by the user that is later used to register the sensor with FireSIGHT.

Example

The following is sample output from the **show network** command that displays the configured network settings of the Firepower Sensor:

```
Device# show network
```

```
----------------------------------------------------
IPv4
Configuration             : manual
Address                   : 10.66.152.137
Netmask                   : 255.255.255.0
Gateway                   : 10.66.152.1
MAC Address               : 44:03:A7:43:05:AD
Management port           : 8305
----------------------------------------------------
IPv6
Configuration             : disabled
Management port           : 8305
----------------------------------------------------
```

The following is sample output from the **show dns** command that displays the configured DNS settings:

```
Device# show dns

search cisco.com
nameserver 192.10.26.10
```

The following is sample output from the **show managers** command that displays the configured management settings:

```
Device# show managers

Host                      : sourcefire-dc.cisco.com
Registration Key          : cisco-sf
Registration              : pending
RPC Status                :
```

Enabling IDS Inspection Globally

Based on your requirements, you can configure the Intrusion Detection System (IDS) inspection at a global level or at an interface level.

You cannot enable IDS inspection on dedicated management interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **utd enable**
4. **utd engine advanced**
5. **threat detection**
6. **exit**
7. **utd**
8. **all-interfaces**
9. **engine advanced**
10. **fail close**
11. **rate** *pps-rate*
12. **redirect-interface** *interface interface-number*

13. **end**

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	**utd enable** **Example:** `Router(config)# utd enable`	Enters unified threat defense configuration mode.
Step 4	**utd engine advanced** **Example:** `Router(config)# utd engine advanced`	Configures the unified threat defense (UTD) advanced engine and enters UTD advanced engine configuration. mode.
Step 5	**threat detection** **Example:** `Router(config-utd-eng-adv)# threat detection`	Configures threat detection or Intrusion Prevention System (IPS) as the operating mode for the Snort engine.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 7	**utd** **Example:** `Router(config)# utd`	Enters unified threat defense configuration mode.
Step 8	**all-interfaces** **Example:** `Router(config-utd)# all-interfaces`	Configures UTD on all Layer 3 interfaces of the device
Step 9	**engine advanced** **Example:** `outer(config-utd)# engine advanced`	Configures the unified threat defense (UTD) advanced engine and enters UTD advaned engine configuration.
Step 10	**fail close** **Example:** `Device(config-engine-std)# fail close`	(Optional) Defines the action when there is a UTD engine failure. Default option is fail-open. Fail-close option drops all the IPS/IDS traffic when there is an UTD engine failure. Fail-open option allows all the IPS/IDS traffic when there is an UTD engine failure.

	Command or Action	Purpose
Step 11	**rate** *pps-rate* **Example:** `Device(config-engine-std)# rate 2000000`	(Optional) Specify the pps rate to push to the sensor. The range is from 1000 to 4000000.
Step 12	**redirect-interface** *interface interface-number* **Example:** `Router(config-utd)# redirect-interface BDI 10`	Configures IDS traffic redirect on an interface.

Step 13	**end**
	Example:
	`Router(config-utd)# end`
		Exits unified threat defense configuration mode and returns to privileged EXEC mode.

Enabling IDS Inspection per Interface

Based on your requirements, you can configure the Intrusion Detection System (IDS) inspection at a global level or at an interface level.

You cannot enable IDS inspection on dedicated management interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **utd enable**
5. **exit**
6. Repeat Steps 3 to 5, on all interfaces that require IDS inspection. Do not configure inspection on management interfaces.
7. **utd engine advanced**
8. **threat detection**
9. **utd**
10. **engine advanced**
11. **fail close**
12. **rate** *range*
13. **redirect interface** *type number*
14. **end**

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	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 0/1/1`	Configures an interface and enters interface configuration mode.
Step 4	**utd enable** **Example:** `Router(config-if)# utd enable`	Enables intrusion detection on an interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	Repeat Steps 3 to 5, on all interfaces that require IDS inspection. Do not configure inspection on management interfaces.	-
Step 7	**utd engine advanced** **Example:** `Router(config)# utd engine advanced`	Configures the unified threat defense (UTD) advanced engine and enters UTD advanced engine configuration. mode.
Step 8	**threat detection** **Example:** `Router(config-utd-eng-adv)# threat detection`	Configures threat detection or Intrusion Prevention System (IPS) as the operating mode for the Snort engine.
Step 9	**utd** **Example:** `Router(config)# utd`	Enters unified threat defense configuration mode.
Step 10	**engine advanced** **Example:** `outer(config-utd)# engine advanced`	Configures the unified threat defense (UTD) advanced engine and enters UTD advaned engine configuration.
Step 11	**fail close** **Example:** `Device(config-engine-std)# fail close`	(Optional) Defines the action when there is a UTD engine failure. Default option is fail-open. Fail-close option drops all the IPS/IDS traffic when there is an UTD engine failure. Fail-open option allows all the IPS/IDS traffic when there is an UTD engine failure.
Step 12	**rate** *range* **Example:** `Device(config-engine-std)# rate 1000`	(Optional) Specify the pps rate to push to the sensor. The range is 1000 to 4000000.

	Command or Action	Purpose
Step 13	**redirect interface** *type number* **Example:** `Router(config-utd)# redirect interface BDI 10`	Configures IDS traffic redirect on an interface.

Step 14	**end**
	Example:
	`Router(config-utd)# end`
		Exits unified threat defense configuration mode and returns to privileged EXEC mode.

Configuration Examples for Cisco Firepower Threat Defense on ISR

Example: Configuring Traffic Redirect on Cisco UCS E-Series Blade

This example shows how to configure ingress and egress interfaces for traffic redirect:

```
Router# configure terminal
Router(config)# interface ucse 1/0/0
Router(config-if)# no ip address
Router(config-if)# no negotiation auto
Router(config-if)# switchport mode trunk
Router(config-if)# no mop enabled
Router(config-if)# no mop sysid
Router(config-if)# exit
Router(config)# interface ucse 1/0/1
Router(config-if)# no ip address
Router(config-if)# no negotiation auto
Router(config-if)# switchport mode trunk
Router(config-if)# no mop enabled
Router(config-if)# no mop sysid
Router(config-if)# service instance 10 ethernet
Router(config-if-srv)# encapsulation dot1q 10
Router(config-if-srv)# rewrite ingress tag pop 1 symmetric
Router(config-if-srv)# bridge domain 10
Router(config-if-srv)# exit
Router(config-if)# exit
Router(config)# interface BDI 10
Router(config-if)# no shutdown
Router(config-if)# ip address 10.1.1.1 255.255.255.0
Router(config-if-srv)# end
```

Example: Bootstrapping the Firepower Sensor

The following example shows how to bootstrap the Firepower Threat Defense sensor:

```
Sourcefire3D login: admin
Password: Sourcefire
```

```
Last login: Tue Nov 12 11:15:03 UTC 2013 on tty1

Copyright 2001-2013, Sourcefire, Inc. All rights reserved. Sourcefire is
a registered trademark of Sourcefire, Inc. All other trademarks are
property of their respective owners.

Sourcefire Linux OS v5.2.0 (build 135)
Sourcefire Virtual Device 64bit v5.2.0 (build 838)

> configure password
Enter current password:
Enter new password:
Confirm new password:

> configure network ipv4 manual 10.66.152.137 255.255.255.0 10.66.152.1
Setting IPv4 network configuration.
ADDRCONF(NETDEV_UP): eth0: link is not ready
e1000: eth0: e1000_phy_read_status: Error reading PHY register
e1000: eth0: e1000_watchdog_task: NIC Link is Up
1000 Mbps Full Duplex, Flow Control: None
ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready

Network settings changed.

> configure network dns servers 192.10.26.10

> configure network dns searchdomains cisco.com

configure manager add sourcefire-dc.cisco.com cisco-sf
Manager successfully configured.
```

Example: Enabling IDS Inspection Globally

```
Router# configure terminal
Router(config)# utd enable
Router(config-utd)# utd engine advanced
Router(config-utd-adv)# threat detection
Router(config-utd-adv)# exit
Router(config)# utd
Router(config-utd)# all-interfaces
Router(config-utd)# engine advanced
Router(config-utd)# fail close
Router(config-utd)# rate 1000
Router(config-utd)# redirect-interface BDI 10
Router(config-utd)# end
```

Example: Enabling IDS Inspection per Interface

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/1/1
Device(config-if)# utd enable
Router(config-utd)# utd engine advanced
Router(config-utd-adv)# threat detection
Router(config-utd-adv)# exit
Router(config)# utd
Router(config-utd)# engine advanced
Router(config-utd)# fail close
Router(config-utd)# rate 1000
Router(config-utd)# redirect-interface BDI 10
Router(config-utd)# end
```

Verifying and Monitoring IDS Inspection

Use the following commands to verify and monitor your Intrusion Detection System (IDS) deployment:

SUMMARY STEPS

1. **enable**
2. **debug platform condition feature utd controlplane**
3. **debug platform condition feature utd dataplane submode**
4. **show platform hardware qfp active utd** {**config** | **status** [**all**] [**clear**] [**drop**] [**general**]}

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **debug platform condition feature utd controlplane**

Enables the debugging of the IDS configuration and status information.

Example:

```
Router# debug platform condition feature utd controlplane

network RF:
  network-rf idb-sync-history events debugging is on
IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop

Feature      Type           Submode       Level
------------|-------------|------------------------------
UTD           controlplane                info

IOSXE Packet Tracing Configs:

Packet Infra debugs:

Ip Address                                               Port
------------------------------------------------------|----------
```

Step 3 **debug platform condition feature utd dataplane submode**

Enables the debugging of IDS packet flow information.

Example:

```
Router# debug platform  condition feature utd dataplane submode

network RF:
  network-rf idb-sync-history events debugging is on
IOSXE Conditional Debug Configs:

Conditional Debug Global State: Stop

Feature      Type           Submode                    Level
------------|-------------|---------------------|----------
UTD           controlplane                             info
UTD           dataplane     fia proxy punt             info

IOSXE Packet Tracing Configs:

Packet Infra debugs:

Ip Address                                               Port
------------------------------------------------------|----------
```

Step 4 **show platform hardware qfp active utd** {**config** | **status** [**all**] [**clear**] [**drop**] [**general**]}

Displays information about the IDS inspection in the Cisco Quantum Flow Processor (QFP).

Example:

```
Router# show platform hardware qfp active utd config

Global flags: 0x40004
Num divert interfaces: 1
Divert UIDBs: 65521 0
FIB information
[0][0] 0x309e3c30
[0][1] 0x0
[1][0] 0x309e4040
[1][1] 0x0
```

Additional References for Cisco Firepower Threat Defense for ISR

Related Documents

Related Topic	Document Title
IOS commands	Cisco IOS Master Command List, All Releases
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

Related Topic	Document Title
UCS E-Series Servers	http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/e/2-0/gs/guide/b_2_0_Getting_Start

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 Firepower Threat Defense for ISR

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 Firepower Threat Defense for ISR

Feature Name	Releases	Feature Information
Cisco Firepower Threat Defense for ISR	Cisco IOS XE Release 3.14S	Cisco Firepower Threat Defense is a premier network security option. It provides a comprehensive suite of Security features such as firewall capabilities, monitoring, alerts, and Intrusion Detection System (IDS). This feature is introduced on Cisco ISR 4000 Series Integrated Services Routers. The following commands were introduced or modified: **debug platform condition feature utd controlplane**, **debug platform condition feature utd dataplane submode**, **ids**, **mode (utd)**, **show platform hardware qfp active feature utd**, **service utd**, **utd**, **utd ids**.

Feature Name	Releases	Feature Information
Cisco Firepower Threat Defense for ISR	Cisco IOS Release 15.5(1)T	Cisco Firepower Threat Defense is a premier network security option. It provides a comprehensive suite of Security features such as firewall capabilities, monitoring, alerts, and Intrusion Detection System (IDS). The following commands were introduced or modified: **ids**, **utd**.

CHAPTER 2

Snort IPS

The Snort IPS feature enables Intrusion Prevention System (IPS) or Intrusion Detection System (IDS) for branch offices on Cisco 4000 Series Integrated Services Routers and Cisco Cloud Services Router 1000v Series. This feature uses the open source Snort solution to enable IPS and IDS. The Snort IPS feature is available in Cisco IOS XE Release 3.16.1S, 3.17S, and later releases.

Note The Virtual Routing and Forwarding (VRF) feature is supported on Snort IPS configuration from Cisco IOS XE Denali Release 16.3.1 and later releases.

This module explains the feature and how it works.

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.

Restrictions for Snort IPS

The following restrictions apply to the Snort IPS feature:

- When you enable boost license on Cisco 4000 Series ISRs, you cannot configure the virtual-service container for Snort IPS.
- Incompatible with the Zone-Based Firewall SYN-cookie feature.
- Network Address Translation 64 (NAT64) is not supported.
- SnortSnmpPlugin is required for SNMP polling in open source Snort. Snort IPS does not support SNMP polling capabilities or MIBs as the SnortSnmp plugin is not installed on UTD.
- **IOS syslog is rate limited and as a result, all alerts generated by Snort may not be visible via the IOS Syslog. However, you can view all Syslog messages if you export them to an external log server.**

Information About Snort IPS

Snort IPS Overview

The Snort IPS feature enables Intrusion Prevention System (IPS) or Intrusion Detection System (IDS) for branch offices on Cisco 4000 Series Integrated Services Routers and Cisco Cloud Services Router 1000v Series. This feature uses the Snort engine to provide IPS and IDS functionalities.

Snort is an open source network IPS that performs real-time traffic analysis and generates alerts when threats are detected on IP networks. It can also perform protocol analysis, content searching or matching, and detect a variety of attacks and probes, such as buffer overflows, stealth port scans, and so on. The Snort engine runs as a virtual container service on Cisco 4000 Series Integrated Services Routers and Cisco Cloud Services Router 1000v Series.

The Snort IPS feature works in the network intrusion detection and prevention mode that provides IPS or IDS functionalities. In the network intrusion detection and prevention mode, Snort performs the following actions:

- Monitors network traffic and analyzes against a defined rule set.
- Performs attack classification.
- Invokes actions against matched rules.

Based on your requirements, you can enable Snort either in IPS or IDS mode. In IDS mode, Snort inspects the traffic and reports alerts, but does not take any action to prevent attacks. In IPS mode, in addition to intrusion detection, actions are taken to prevent attacks.

The Snort IPS monitors the traffic and reports events to an external log server or the IOS syslog. Enabling logging to the IOS syslog may impact performance due to the potential volume of log messages. External third-party monitoring tools, which supports Snort logs, can be used for log collection and analysis.

Snort IPS Signature Package

The UTD OVA is included in the security license of the router. By default, the router is loaded only with community signature package. There are two types of subscriptions :

- Community Signature Package
- Subscriber-based Signature Package

The community signature package rule set offers limited coverage against threats. The subscriber-based signature package rule set offers the best protection against threats. It includes coverage in advance of exploits, and also provides the fastest access to the updated signatures in response to a security incident or the proactive discovery of a new threat. This subscription is fully supported by Cisco and the package will be updated on Cisco.com. You can download the subscriber-based signature package from the Download Software page.

If the user downloads the signature package manually from the download software page, then the user should ensure that the package has the same version as the Snort engine version. For example, if the Snort engine version is 2982, then the user should download the same version of the signature package. If there is a version mismatch, the signature package update will be rejected and it will fail.

Note When the signature package is updated, the engine will be restarted and the traffic will be interrupted or bypass inspection for a short period depending on their data plane fail-open/fail-close configuration.

Minimum Supported Cisco IOS XE Release and UTD Package Versions for Signature Updates

Table 1 below lists the minimum Cisco IOS XE releases and their respective UTD package versions that support signature package updates post January, 2020. The Cisco IOS XE releases and their respective UTD package versions that are prior to those listed in the table are not supported. The Cisco IOS XE releases and their respective UTD package versions that are more recent than those listed in the table are supported from their first release.

Table 2: UTD Signature Package Update Support Version Matrix

Cisco IOS XE Release	UTD Package Version
16.6.7	1.0.10_SV29111_XE_16_6
16.9.4	1.0.4_SV29111_XE_16_9
16.10.2	1.0.9_SV2.9.11.1_XE16.10

Note When UTD is oversubscribed, the threat defence channel state changes between green and red. The UTD dataplane either drops all further packets if fail-close is configured or forwards the packets un-inspected if fail-close is not configured (default). When the UTD serviceplane recovers from over-subscription, it responds to the UTD dataplane with the green status.

Snort IPS Solution

The Snort IPS solution consists of the following entities:

- Snort sensor—Monitors the traffic to detect anomalies based on the configured security policies (that includes signatures, statistics, protocol analysis, and so on) and sends alert messages to the Alert/Reporting server. The Snort sensor is deployed as a virtual container service on the router.
- Signature store—Hosts the Cisco Signature packages that are updated periodically. These signature packages are downloaded to Snort sensors either periodically or on demand. Validated signature packages are posted to Cisco.com. Based on the configuration, signature packages can be downloaded from Cisco.com or a local server.

  The following domains are accessed by the router in the process of downloading the signature package from cisco.com:

  - api.cisco.com
  - apx.cisco.com
  - cloudsso.cisco.com
  - cloudsso-test.cisco.com
  - cloudsso-test3.cisco.com
  - cloudsso-test4.cisco.com
  - cloudsso-test5.cisco.com
  - cloudsso-test6.cisco.com
  - cloudsso.cisco.com
  - download-ssc.cisco.com
  - dl.cisco.com
  - resolver1.opendns.com
  - resolver2.opendns.com

  Note If you are downloading signature packages from a local server to hold the signature packages, only HTTP is supported.

  Signature packages must be manually downloaded from Cisco.com to the local server by using Cisco.com credentials before the Snort sensor can retrieve them.

  The Snort container performs a domain-name lookup (on the DNS server(s) configured on the router) to resolve the location for automatic signature updates from Cisco.com or on the local server, if the URL is not specified as the IP address.

- Alert/Reporting server—Receives alert events from the Snort sensor. Alert events generated by the Snort sensor can either be sent to the IOS syslog or an external syslog server or to both IOS syslog and external syslog server. No external log servers are bundled with the Snort IPS solution.

- Management—Manages the Snort IPS solution. Management is configured using the IOS CLI. Snort Sensor cannot be accessed directly, and all configuration can only be done using the IOS CLI.

Overview of Snort Virtual Service Interfaces

The Snort sensor runs as a service on routers. Service containers use virtualization technology to provide a hosting environment on Cisco devices for applications.

You can enable Snort traffic inspection either on a per interface basis or globally on all supported interfaces. The traffic to be inspected is diverted to the Snort sensor and injected back. In Intrusion Detection System (IDS), identified threats are reported as log events and allowed. However, in Intrusion Prevention System (IPS), action is taken to prevent attacks along with log events.

The Snort sensor requires two VirtualPortGroup interfaces. The first VirtualPortGroup interface is used for management traffic and the second for data traffic between the forwarding plane and the Snort virtual container service. Guest IP addresses must be configured for these VirtualPortGroup interfaces. The IP subnet assigned to the management VirtualPortGroup interface should be able to communicate with the Signature server and Alert/Reporting server.

The IP subnet of the second VirtualPortGroup interface must not be routable on the customer network because the traffic on this interface is internal to the router. Exposing the internal subnet to the outside world is a security risk. We recommend the use of 192.0.2.0/30 IP address range for the second VirtualPortGroup subnet. The use of 192.0.2.0/24 subnet is defined in RFC 3330.

You can also use the management interface under the **virtual-service** command for management traffic. If you configure the management interface, you still need two VirtualPortGroup interfaces. However, do not configure the **guest ip address** for the first VirtualPortGroup interface.

You can assign the Snort virtual container service IP address on the same management network as the router on which the virtual service is running. This configuration helps if the syslog or update server is on the management network and is not accessible by any other interfaces.

Virtual Service Resource Profile

The Snort IPS virtual service supports three resource profiles: Low, Medium, and High. These profiles indicate the CPU and memory resources required to run the virtual service. You can configure one of these resource profiles. The resource profile configuration is optional. If you do not configure a profile, the virtual service is activated with its default resource profile. This table provides the resource profiles details for Cisco 4000 Series ISR and Cisco Cloud Services Router 1000v Series.

Platform	Profile	Virtual Service Resource Requirements		Platform Requirements
		System CPU	Memory	
Cisco 4321 ISR	Default	50%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)

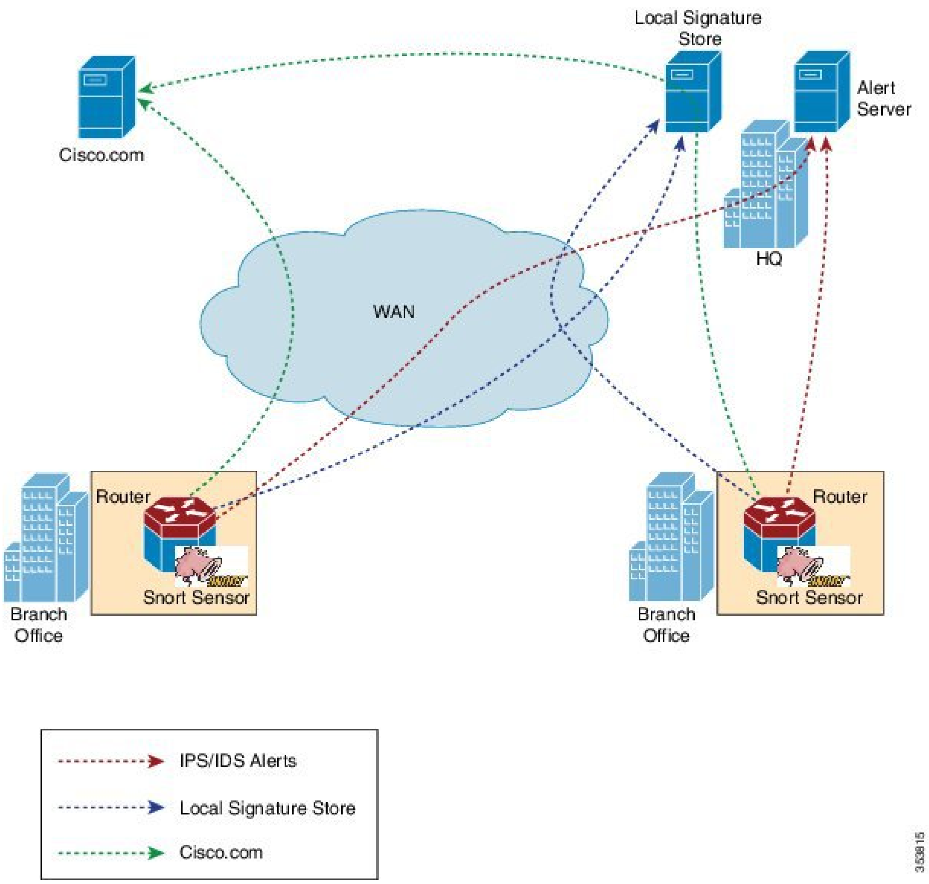
Platform	Profile	Virtual Service Resource Requirements		Platform Requirements
		System CPU	Memory	
Cisco 4331 ISR	Low (Default)	25%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	Medium	50%	Min: 2GB (RAM) Min: 1GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	High	75%	Min: 4GB (RAM) Min: 2GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
Cisco 4351 ISR	Low (Default)	25%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	Medium	50%	Min: 2GB (RAM) Min: 1GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	High	75%	Min: 4GB (RAM) Min: 2GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
Cisco 4431 ISR	Low (Default)	25%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	Medium	50%	Min: 2GB (RAM) Min: 1GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	High	75%	Min: 4GB (RAM) Min: 2GB (Disk/Flash)	Min: 12GB (RAM) Min: 12GB(Disk/Flash)

Platform	Profile	Virtual Service Resource Requirements		Platform Requirements
		System CPU	Memory	
Cisco 4451 ISR	Low (Default)	25%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	Medium	50%	Min: 2GB (RAM) Min: 1GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	High	75%	Min: 4GB (RAM) Min: 2GB (Disk/Flash)	Min: 12GB (RAM) Min: 12GB(Disk/Flash)
Cisco CSR 1000V	Low (Default)	25%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	Medium	50%	Min: 2GB (RAM) Min: 1GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	High	75%	Min: 3GB (RAM) Min: 2GB (Disk/Flash)	Min: 12GB (RAM) Min: 12GB(Disk/Flash)

Deploying Snort IPS

The figure illustrates a Snort IPS deployment scenario:

Figure 2: Snort IPS Deployment Scenario



The following steps describes the deployment of the Snort IPS solution:

- The Snort OVA file is copied to Cisco routers, installed, and then activated.
- Signature packages are downloaded either from Cisco.com or a configured local server to Cisco routers.
- Network intrusion detection or prevention functionality is configured.
- The Alert/Reporting server is configured to receive alerts from the Snort sensor.

Threat Inspection Alerts Visibility

From the Cisco IOS XE Fuji 16.8 release, you can get summarized details for the following threat-inspection alerts:

- The top 10 threat-inspection alerts (IDS/IPS) and counts are summarized for last 24 hours.
- For each signature-ID top 10 SIP, DIP, and VRF summary for the last 24 hours.

Note The last 24 hours period accounts for exact prior 24 hour duration from the time you request alert summary using CLI.

The visibility feature is available only on single tenancy and not on multi-tenancy.

Use **show utd engine standard logging threat-inspection statistics** *detail* command to view the alert summary.

Enabling and Disabling Logging of the Threat Inspection Alerts

To enable logging of the threat inspection alert statistics, perform the following steps:

```
Router(config)#utd engine standard
Router(config-utd-eng-std)#threat-inspection
Router(config-utd-engstd-insp)#logging statistics enable
Router(config-utd-engstd-insp)#exit
```

To disable logging of the threat inspection alert statistics, perform the following steps:

```
Router(config)#utd engine standard
Router(config-utd-eng-std)#threat-inspection
Router(config-utd-engstd-insp)#no logging statistics enable
Router(config-utd-engstd-insp)#exit
```

How to Deploy Snort IPS

To deploy Snort IPS on supported devices, perform the following tasks:

1. Provision the device.

   Identify the device to install the Snort IPS feature.

2. Obtain the license.

   The Snort IPS functionality is available only in Security Packages which require a security license to enable the service. This feature is available in Cisco IOS XE Release 3.16.1S, 3.17S, and later releases.

Note Contact Cisco Support to obtain the license.

3. Install the Snort OVA file.
4. Configure VirtualPortGroup interfaces and virtual-service.
5. Activate the Snort virtual container service.
6. Configure Snort IPS or IDS mode and policy.
7. Configure the reporting of events to an external alert/log server or IOS syslog or both.
8. Configure the Signature update method.
9. Update the Signatures.
10. Enable IPS globally or on desired interfaces.

Installing the Snort OVA File

An OVA file is an Open Virtualization Archive that contains a compressed, installable version of a virtual machine. The Snort IPS is available as a virtual container service. You must download this OVA file on to the router and use the **virtual-service install** CLI to install the service.

The service OVA file is not bundled with the Cisco IOS XE Release images that are installed on the router. However, the OVA files may be preinstalled in the flash of the router.

You must use a Cisco IOS XE image with security license. During the OVA file installation, the security license is checked and an error is reported if the license is not present.

SUMMARY STEPS

1. **enable**
2. **virtual-service install name** *virtual-service-name* **package** *file-url* **media** *file-system*
3. **show virtual-service list**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**virtual-service install name** *virtual-service-name* **package** *file-url* **media** *file-system* **Example:** `Device# virtual-service install name UTDIPS package harddisk:utd-ips-v102.ova media harddisk:`	Installs an application on the virtual services container of a device. • The length of the name is 20 characters. Hyphen (-) is not a valid character. • You must specify the complete path of the OVA package to be installed. **Note** OVA installation works on both hard disk and bootflash, the preferred filesystem to install the OVA will be hard disk.
Step 3	**show virtual-service list** **Example:** `Device# show virtual-service list`	Displays the status of the installation of all applications installed on the virtual service container.

Configuring VirtualPortGroup Interfaces and Virtual Service

You must configure two VirtualPortGroup interfaces and configure guest IP addresses for both interfaces. However, if you configure a management interface by using the **vnic management GigabitEthernet0** command, then do not configure the guest IP address for the first VirtualPortGroup interface.

Note The VirtualPortGroup interface for data traffic must use a private or nonroutable IP address. We recommend the use of 192.0.2.0/30 IP address range for this interface.

Note Before you change the Cisco IOS software image from any of the XE 3.x versions to XE 16.2.1, or from XE 16.2.1 to any of the XE 3.x versions, uninstall the virtual-service by using the **virtual-service uninstall name [name]** command for each virtual-service on the device. If one of the virtual-services is the ISR-WAAS service, which is installed with the **service waas enable** command, use the **service waas disable** command.

After the device is upgraded with the new version of Cisco IOS software image, re-install the virtual-services. For ISR-WAAS, use the **service wass enable** command, and for other virtual-services, use the **virtual-service install name [name] package [.ova file]** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *VirtualPortGroup number*
4. **ip address** *ip-address mask*
5. **exit**
6. **interface** *type number*
7. **ip address** *ip-address mask*
8. **exit**
9. **virtual-service** *name*
10. **profile** *profile-name*
11. **vnic gateway VirtualPortGroup** *interface-number*
12. **guest ip address** *ip-address*
13. **exit**
14. **vnic gateway VirtualPortGroup** *interface-number*
15. **guest ip address** *ip-address*
16. **exit**
17. **vnic management GigabitEthernet0**
18. **guest ip address** *ip-address*
19. **exit**
20. **activate**
21. **end**

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.

	Command or Action	Purpose
Step 3	**interface** *VirtualPortGroup number* **Example:** `Device(config)# interface VirtualPortGroup 0`	Configures an interface and enters interface configuration mode. • Configure a VirtualPortGroup interface. This interface is used for management traffic when the management interface GigabitEthernet0 is not used.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.252`	Sets a primary IP address for an interface. This interface needs to be routable to the signature update server and external log server.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**interface** *type number* **Example:** `Device(config)# interface VirtualPortGroup 1`	Configures an interface and enters interface configuration mode. • Configure a VirtualPortGroup interface. • This interface is used for data traffic.
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.0.2.1 255.255.255.252`	Sets a primary IP address for an interface. • This IP address should not be routable to the outside network. • The IP address is assigned from the recommended 192.0.2.0/30 subnet.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**virtual-service** *name* **Example:** `Device(config)# virtual-service UTDIPS`	Configures a virtual container service and enters virtual service configuration mode. • The *name* argument is the logical name that is used to identify the virtual container service.

Step 10	**profile** *profile-name*
	Example:
	`Device(config-virt-serv)#profile high`
	Example:
	`Device(config-virt-serv)#profile multi-tenancy`
		(Optional) Configures a resource profile. If you do not configure the resource profile, the virtual service is activated with its default resource profile. The options are: low, medium, high, and multi-tenancy. (For multi-tenancy mode (Cisco CSR 1000v only), a `profile multi-tenancy` command must be configured.)

Step 11	**vnic gateway VirtualPortGroup** *interface-number* **Example:**	Creates a virtual network interface card (vNIC) gateway interface for the virtual container service, maps the vNIC gateway interface to the virtual port group, and enters the virtual-service vNIC configuration mode.

	Command or Action	Purpose
	`Device(config-virt-serv)# vnic gateway VirtualPortGroup 0`	• The interface referenced in this command must be the one configured in Step 3. This command maps the interface that is used for management purposes.
Step 12	**guest ip address** *ip-address* **Example:** `Device(config-virt-serv-vnic)# guest ip address 10.1.1.2`	(Optional) Configures a guest vNIC address for the vNIC gateway interface. • **Note** Configure this command only if the **vnic management gigabitethernet0** command specified in Step 17 is not configured.
Step 13	**exit** **Example:** `Device(config-virt-serv-vnic)# exit`	Exits virtual-service vNIC configuration mode and returns to virtual service configuration mode.
Step 14	**vnic gateway VirtualPortGroup** *interface-number* **Example:** `Device(config-virt-serv)# vnic gateway VirtualPortGroup 1`	Creates a vNIC gateway interface for the virtual container service, maps the vNIC gateway interface to the virtual port group, and enters the virtual-service vNIC configuration mode. • This interface referenced in this command must be the one configured in Step 6. This command maps the interface in the virtual container service that is used by Snort for monitoring the user traffic.
Step 15	**guest ip address** *ip-address* **Example:** `Device(config-virt-serv-vnic)# guest ip address 192.0.2.2`	Configures a guest vNIC address for the vNIC gateway interface.
Step 16	**exit** **Example:** `Device(config-virt-serv-vnic)# exit`	Exits virtual-service vNIC configuration mode and returns to virtual service configuration mode.
Step 17	**vnic management GigabitEthernet0** **Example:** `Device(config-virt-serv)# vnic management GigabitEthernet0`	(Optional) Configures the GigabitEthernet interface as the vNIC management interface. • The management interface must either be a VirtualPortGroup interface or GibagitEthernet0 interface. • If you do not configure the **vnic management GigabitEthernet0** command, then you must configure the **guest ip address** command specified in Step 12.
Step 18	**guest ip address** *ip-address* **Example:** `Device(config-virt-serv-vnic)# guest ip address 209.165.201.1`	(Optional) Configures a guest vNIC address for the vNIC management interface and it must be in the same subnet as the management interface and GigabitEthernet0 configuration.

	Command or Action	Purpose
Step 19	**exit** **Example:** `Device(config-virt-serv-vnic)# exit`	Exits virtual-service vNIC configuration mode and returns to virtual service configuration mode.
Step 20	**activate** **Example:** `Device(config-virt-serv)# activate`	Activates an application installed in a virtual container service.

Step 21	**end**
	Example:
	`Device(config-virt-serv)# end`
		Exits virtual service configuration mode and returns to privileged EXEC mode.

Configuring Snort IPS Globally

Based on your requirements, configure the Intrusion Prevention System (IPS) or Intrusion Detection System (IDS) inspection at a global level or at an interface. Perform this task to configure IPS globally on a device.

Note The term global refers to Snort IPS running on all supported interfaces.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **utd threat-inspection whitelist**
4. **generator id** *generator-id* **signature id** *signature-id* [**comment** *description*]
5. **exit**
6. **utd engine standard**
7. **logging** {**host** *hostname* | **syslog**}
8. **threat-inspection**
9. **threat** {**detection** | **protection** }
10. **policy** {**balanced** | **connectivity** | **security**}
11. **whitelist**
12. **signature update occur-at** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hour minute*
13. **signature update server** {**cisco** | **url** *url* } [**username** *username* [**password** *password*]]
14. **logging level** {**alert** | **crit** | **debug** | **emerg** | **err** | **info** | **notice** | **warning**}
15. **exit**
16. **utd**
17. **redirect interface virtualPortGroup** *interface-number*
18. **all-interfaces**
19. **engine standard**
20. **fail close**
21. **exit**

22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter you password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**utd threat-inspection whitelist** **Example:** `Device(config)# utd threat-inspection whitelist`	(Optional) Enables the UTD allowed list configuration mode.
Step 4	**generator id** *generator-id* **signature id** *signature-id* [**comment** *description*] **Example:** `Device(config-utd-whitelist)# generator id 24 signature id 24245 comment traffic from branchoffice1`	Configures signature IDs to appear in the allowed list. • Signature IDs can be copied from alerts that needs to be suppressed. • You can configure multiple signature IDs. • Repeat this step for each signature ID that needs to be added to the allowed list.
Step 5	**exit** **Example:** `Device(config-utd-whitelist)# exit`	Exits UTD allowed list configuration mode and returns to global configuration mode.
Step 6	**utd engine standard** **Example:** `Device(config)# utd engine standard`	Configures the unified threat defense (UTD) standard engine and enters UTD standard engine configuration mode.

Step 7	**logging** {**host** *hostname* | **syslog**}
	Example:
	`Device(config-utd-eng-std)# logging host syslog.yourcompany.com`
		Enables the logging of emergency messages to a server.

Step 8	**threat-inspection** **Example:** `Device(config-utd-eng-std)# threat-inspection`	Configures threat inspection for the Snort engine.

Step 9	**threat** {**detection** | **protection** }
	Example:
	`Device(config-utd-eng-std-insp)# threat protection`
		Configures threat detection or Intrusion Prevention System (IPS) as the operating mode for the Snort engine.
		• The default is **detection**.

	Command or Action	Purpose
		• Configure the **detection** keyword to configure Intrusion Detection System (IDS).

Step 10	**policy** {**balanced** | **connectivity** | **security**}
	Example:
	`Device(config-utd-eng-std-insp)# policy security`
		Configures the security policy for the Snort engine.
		• The default policy option is **balanced**.

Step 11	**whitelist** **Example:** `Device(config-utd-eng-std-insp)# whitelist`	(Optional) Enables allowed listing under the UTD engine.
Step 12	**signature update occur-at** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hour minute* **Example:** `Device(config-utd-eng-std-insp)# signature update occur-at daily 0 0`	Configures the signature update interval parameters. This configuration will trigger the signature update to occur at midnight.
Step 13	**signature update server** {**cisco** | **url** *url* } [**username** *username* [**password** *password*]] **Example:** `Device(config-utd-eng-std-insp)# signature update server cisco username abcd password cisco123`	Configures the signature update server parameters. You must specify the signature update parameters with the server details. If you use Cisco.com for signature updates, you must provide the username and password. If you use local server for signature updates, based on the server settings you can provide the username and password.

Step 14	**logging level** {**alert** | **crit** | **debug** | **emerg** | **err** | **info** | **notice** | **warning**}
	Example:
	`Device(config-utd-eng-std-insp)# logging level emerg`
		Enables the log level.

Step 15	**exit** **Example:** `Device(config-utd-eng-std-insp)# exit`	Exits UTD standard engine configuration mode and returns to global configuration mode.
Step 16	**utd** **Example:** `Device(config)# utd`	Enables unified threat defense (UTD) and enters UTD configuration mode.
Step 17	**redirect interface virtualPortGroup** *interface-number* **Example:** `Device(config-utd)# redirect interface virtualPortGroup 1`	(Optional) Redirects to a VirtualPortGroup interface. This is the data traffic interface. If you do not configure this interface, it is auto-detected.
Step 18	**all-interfaces** **Example:** `Device(config-utd)# all-interfaces`	Configures UTD on all Layer 3 interfaces of the device.

	Command or Action	Purpose
Step 19	**engine standard** **Example:** `Device(config-utd)# engine standard`	Configures the Snort-based unified threat defense (UTD) engine and enters standard engine configuration mode.
Step 20	**fail close** **Example:** `Device(config-engine-std)# fail close`	(Optional) Defines the action when there is a UTD engine failure. Default option is fail-open. Fail-close option drops all the IPS/IDS traffic when there is an UTD engine failure. Fail-open option allows all the IPS/IDS traffic when there is an UTD engine failure.
Step 21	**exit** **Example:** `Device(config-eng-std)# exit`	Exits standard engine configuration mode and returns to global configuration mode.

Step 22	**end**
	Example:
	`Device(config-utd)# end`
		Exits UTD configuration mode and returns to global configuration mode.

Configuring Snort IDS Inspection Globally

Based on your requirements, configure either Intrusion Prevention System (IPS) or Intrusion Detection System (IDS) inspection at a global level or at an interface level. Perform this task to configure IDS on a per-interface basis.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **utd enable**
5. **exit**
6. Repeat Steps 3 to 5, on all interfaces that require inspection.
7. **utd threat-inspection whitelist**
8. **generator id** *generator-id* **signature id** *signature-id* [**comment** *description*]
9. **exit**
10. **utd engine standard**
11. **logging** {**host** *hostname* | **syslog**}
12. **threat-inspection**
13. **threat** {**detection** | **protection** }
14. **policy** {**balanced** | **connectivity** | **security**}
15. **whitelist**
16. **signature update occur-at** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hour minute*
17. **signature update server** {**cisco** | **url** *url*} [**username** *username* [**password** *password*]]
18. **logging level** {**alert** | **crit** | **debug** | **emerg** | **err** | **info** | **notice** | **warning**}
19. **exit**

20. **utd**
21. **redirect interface virtualPortGroup** *interface-number*
22. **engine standard**
23. **fail close**
24. **exit**
25. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter you password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**utd enable** **Example:** `Device(config-if)# utd enable`	Enables unified threat defense (UTD).
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	Repeat Steps 3 to 5, on all interfaces that require inspection.	–
Step 7	**utd threat-inspection whitelist** **Example:** `Device(config)# utd threat-inspection whitelist`	(Optional) Enables the UTD allowed list configuration mode.
Step 8	**generator id** *generator-id* **signature id** *signature-id* [**comment** *description*] **Example:** `Device(config-utd-whitelist)# generator id 24 signature id 24245 comment traffic from branchoffice1`	Configures signature IDs to appear on the allowed list. • Signature IDs can be copied from alerts that needs to be suppressed. • You can configure multiple signature IDs. • Repeat this step for each signature ID that needs to appear on the allowed list.
Step 9	**exit** **Example:** `Device(config-utd-whitelist)# exit`	Exits UTD allowed list configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 10	**utd engine standard** **Example:** `Device(config)# utd engine standard`	Configures the unified threat defense (UTD) standard engine and enters UTD standard engine configuration mode.

Step 11	**logging** {**host** *hostname* | **syslog**}
	Example:
	`Device(config-utd-eng-std)# logging syslog`
		Enables the logging of critical messages to the IOSd syslog.

Step 12	**threat-inspection** **Example:** `Device(config-utd-eng-std)# threat-inspection`	Configures threat inspection for the Snort engine.

Step 13	**threat** {**detection** | **protection** }
	Example:
	`Device(config-utd-eng-std-insp)# threat detection`
		Configures threat protection or Intrusion Detection System (IDS) as the operating mode for the Snort sensor.
		• Configure the **protection** keyword to configure Intrusion Prevention System (IPS).

Step 14	**policy** {**balanced** | **connectivity** | **security**}
	Example:
	`Device(config-utd-eng-std-insp)# policy balanced`
		Configures the security policy for the Snort sensor.

Step 15	**whitelist** **Example:** `Device(config-utd-eng-std-insp)# whitelist`	(Optional) Enables allowed listing of traffic.
Step 16	**signature update occur-at** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hour minute* **Example:** `Device(config-utd-eng-std-insp)# signature update occur-at daily 0 0`	Configures the signature update interval parameters. This configuration will trigger the signature update to occur at midnight.
Step 17	**signature update server** {**cisco** | **url** *url*} [**username** *username* [**password** *password*]] **Example:** `Device(config-utd-eng-std-insp)# signature update server cisco username abcd password cisco123`	Configures the signature update server parameters. You must specify the signature update parameters with the server details. If you use Cisco.com for signature updates, you must provide the username and password. If you use local server for signature updates, based on the server settings you can provide the username and password.

Step 18	**logging level** {**alert** | **crit** | **debug** | **emerg** | **err** | **info** | **notice** | **warning**}
	Example:
	`Device(config-utd-eng-std-insp)# logging level crit`
		Enables the log level.

Step 19	**exit** **Example:**	Exits UTD standard engine configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-utd-eng-std-insp)# exit`	
Step 20	**utd** **Example:** `Device(config)# utd`	Enables unified threat defense (UTD) and enters UTD configuration mode.
Step 21	**redirect interface virtualPortGroup** *interface-number* **Example:** `Device(config-utd)# redirect interface virtualPortGroup 1`	(Optional) Redirects to a VirtualPortGroup interface. This is the data traffic interface. If you do not configure this interface, it is auto-detected.
Step 22	**engine standard** **Example:** `Device(config-utd)# engine standard`	Configures the Snort-based unified threat defense (UTD) engine and enters standard engine configuration mode.
Step 23	**fail close** **Example:** `Device(config-engine-std)# fail close`	(Optional) Defines the action when there is a UTD engine failure. Default option is fail-open. Fail-close option drops all the IPS/IDS traffic when there is an UTD engine failure. Fail-open option allows all the IPS/IDS traffic when there is an UTD engine failure.
Step 24	**exit** **Example:** `Device(config-eng-std)# exit`	Exits standard engine configuration mode and returns to global configuration mode.

Step 25	**end**
	Example:
	`Device(config-utd)# end`
		Exits configuration mode and returns back to exec mode.

Displaying the List of Active Signatures

Active signatures are the ones that prompt Snort IDS/IPS to take action against threats. If the traffic matches with any of the active signatures, Snort container triggers alert in the IDS mode, and drops the traffic in the IPS mode.

The **utd threat-inspection signature active-list write-to bootflash: file name** command provides a list of active signatures and a summary of the total number of active signatures, drop signatures, and alert signatures.

Configuring Quality of Service Policy for Monitoring the Container's Health

It is recommended to configure a Quality of Service (QoS) policy to ensure the health probes that monitor the container's health are not impacted at high traffic rates.

SUMMARY STEPS

1. **ip access-list extended** {acl-name | acl-number}

2. sequence-number permit protocol source *source-wildcard destination destination-wildcard* [precedence] [tos *tos* tos] [log] [time-range*time-range-name*] [fragments]
3. **exit**
4. class-map { [type inspect match-all] | [match-any] } *class-map-name*
5. match access-group { *access-group* | name *access-group-name*}
6. **exit**
7. policy-map *policy-map-name*
8. class {*class-name* | class-default
9. priority level *level*
10. **exit**
11. **interface** *type number*
12. service-policy [history | {output} *policy-map-name* | type control *control-policy-name*]
13. **end**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**ip access-list extended** {acl-name | acl-number}
	Example:
	`Device(config)# ip access-list extended health_probes_accesslist`
		Enables extended ACL configuration mode. The CLI enters the extended ACL configuration mode in which all subsequent commands apply to the current extended access list.

Step 2	sequence-number permit protocol source *source-wildcard destination destination-wildcard* [precedence] [tos *tos* tos] [log] [time-range*time-range-name*] [fragments] **Example:** `Device(config-ext-nacl)# 10 permit udp any eq 3367 any eq 3367`	Specifies a permit statement in named IP access list mode. This access list happens to use a **permit**statement first, but a **deny** statement could appear first, depending on the order of statements you need.
Step 3	**exit** **Example:** `Device(config-ext-nacl)# exit`	Exits extended ACL configuration mode and returns to global configuration mode.

Step 4	class-map { [type inspect match-all] | [match-any] } *class-map-name*
	Example:
	`Device(config)# class-map match-all health_probes_cmap`
		Specifies the name of the class map to be created and enters QoS class map configuration mode.

Step 5	match access-group { *access-group* | name *access-group-name*}
	Example:
	`Device(config-cmap)# match access-group name health_probes_accesslist`
		Configure the match criteria for a class map to be successful match criteria for all packets.

Step 6	**exit** **Example:**	Exits class-map configuration mode and returns to global configuration mode.

	Command or Action	Purpose
	`Device(config-cmap)# exit`	
Step 7	policy-map *policy-map-name* **Example:** `Device(config)# policy-map health_probes_pmap`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy, and enters QoS policy-map configuration mode.

Step 8	class {*class-name* | class-default
	Example:
	`Device(config-pmap)# class health_probes_cmap`
		Specifies the name of the class whose policy you want to create or change or specifies the default class (commonly known as the class-default class) before you configure its policy, and enters policy-map class configuration mode.

Step 9	priority level *level* **Example:** `Device(config-pmap)# priority level 1`	Assigns priority to a traffic class at the priority level specified. • Enter the level of priority assigned to the priority class. Valid values are 1 (high priority) and 2 (low priority). The default is 1.
Step 10	**exit** **Example:** `Device(config-pmap)# exit`	Exits policy-map configuration mode and returns to global configuration mode.
Step 11	**interface** *type number* **Example:** `Device(config)# interface VirtualPortGroup 1`	Configures an interface and enters interface configuration mode. • Configure a VirtualPortGroup interface. • This interface is used for data traffic.
Step 12	service-policy [history | {output} *policy-map-name* | type control *control-policy-name*] **Example:** `Device(config-if)# service-policy output health_probes_pmap`	Attaches a policy map to a class. The name of a service policy map (created using the **policy-map** command) to be attached. The name can be a maximum of 40 alphanumeric characters.

Step 13	**end**
	Example:
	`Device(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Snort IPS

Example: Configuring VirtualPortGroup Interfaces and Virtual Service

```
Device# configure terminal
Device(config)# interface VirtualPortGroup 0
Device(config-if)# ip address 10.1.1.1 255.255.255.252
Device(config-if)# exit
Device(config)# interface VirtualPortGroup 1
Device(config-if)# ip address 192.0.2.1 255.255.255.252
Device(config-if)# exit
Device(config)# virtual-service UTDIPS
Device(config-virt-serv)# vnic gateway VirtualPortGroup 0
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# vnic gateway VirtualPortGroup 1
Device(config-virt-serv-vnic)# guest ip address 192.0.2.2
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# vnic management GigabitEthernet0
Device(config-virt-serv-vnic)# guest ip address 209.165.201.1
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# activate
Device(config-virt-serv-vnic)# end
```

Example: Configuring a Different Resource Profile

```
Device# configure terminal
Device(config)# virtual-service UTDIPS
Device(config-virt-serv)# no activate
Device(config-virt-serv)# end
Device# virtual-service uninstall name UTDIPS
Device# configure terminal
Device(config)# virtual-service UTDIPS
Device(config-virt-serv)# profile medium
Device(config-virt-serv)# end
Device# virtual-service install name UTDIPS package:utd.ova
Device# configure terminal
Device(config)# virtual-service UTDIPS
Device(config-virt-serv)# activate
Device(config-virt-serv)# end
```

Example: Configuring Snort IPS Globally

The following example shows how to configure Intrusion Prevention System (IPS) globally on a device:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# threat protection
Device(config-utd-eng-std-insp)# policy security
Device(config-utd-eng-std)# exit
Device(config)# utd
Device(config-utd)# all-interfaces
Device(config-utd)# engine standard
Device(config-utd-whitelist)# end
Device#
```

Example: Configuring Snort IPS Inspection per Interface

The following example shows how to configure Snort Intrusion Detection System (IDS) on a per-interface basis:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# threat detection
Device(config-utd-eng-std-insp)# policy security
Device(config-utd-eng-std)# exit
Device(config)# utd
Device(config-utd)# engine standard
Device(config-eng-std)# exit
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# utd enable
Device(config-if)# exit
```

Example: Configuring UTD with VRF on both Inbound and Outbound Interface

```
Device# configure terminal
Device(config)# vrf definition VRF1
Device(config-vrf)# rd 100:1
Device(config-vrf)# route-target export 100:1
Device(config-vrf)# route-target import 100:1
Device(config-vrf)# route-target import 100:2
!
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit
!
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit
!
Device(config-vrf-af)# vrf definition VRF2
Device(config-vrf)# rd 100:2
Device(config-vrf)# route-target export 100:2
Device(config-vrf)# route-target import 100:2
Device(config-vrf)# route-target import 100:1
!
Device(config-vrf)# address-family ipv4
Device(config-vrf-af)# exit
!
Device(config-vrf)# address-family ipv6
Device(config-vrf-af)# exit
!
Device(config-vrf)# interface VirtualPortGroup0
Device(config-if)# ip address 192.0.2.1 255.255.255.252
Device(config-if)# no mop enabled
Device(config-if)# no mop sysid
!
Device(config-if)# interface VirtualPortGroup1
Device(config-if)# ip address 192.0.2.5 255.255.255.252
Device(config-if)# no mop enabled
Device(config-if)# no mop sysid
!
Device(config-if)# interface GigabitEthernet0/0/2
Device(config-if)# vrf forwarding VRF1
Device(config-if-vrf)# ip address 192.1.1.5 255.255.255.0
Device(config-if-vrf)# ipv6 address A000::1/64
!
```

```
Device(config-if)# interface GigabitEthernet0/0/3
Device(config-if)# vrf forwarding VRF2
Device(config-if-vrf)# ip address 192.1.1.5 255.255.255.0
Device(config-if-vrf)# ipv6 address B000::1/64
!
Device(config-if-vrf)# router bgp 100
Device(config-if-vrf)# bgp log-neighbor-changes
!
Device(config-vrf)# address-family ipv4 vrf VRF1
Device(config-vrf-af)# redistribute connected
Device(config-vrf-af)# redistribute static
Device(config-vrf-af)# exit
!
Device(config-vrf)# address-family ipv6 vrf VRF1
Device(config-vrf-af)# redistribute connected
Device(config-vrf-af)# redistribute static
Device(config-vrf-af)# exit
!
Device(config-vrf)# address-family ipv4 vrf VRF2
Device(config-vrf-af)# redistribute connected
Device(config-vrf-af)# redistribute static
Device(config-vrf-af)# exit
!
Device(config-vrf)# address-family ipv6 vrf VRF2
Device(config-vrf-af)# redistribute connected
Device(config-vrf-af)# redistribute static
Device(config-vrf-af)# exit
!
Device(config)# utd
Device(config-utd)# all-interfaces
Device(config-utd)# engine standard
Device(config-utd)# exit

Device(config)# utd engine standard
Device(config-utd-eng-std)# logging syslog
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-engstd-insp)# threat protection
Device(config-utd-engstd-insp)# policy security
Device(config-utd-engstd-insp)# exist
Device(config-utd-eng-std)# exit
!
Device(config)# virtual-service utd
Device(config-virt-serv)# profile low
Device(config-virt-serv)# vnic gateway VirtualPortGroup0
Device(config-virt-serv-vnic)# guest ip address 192.0.2.2
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# vnic gateway VirtualPortGroup1
Device(config-virt-serv-vnic)# guest ip address 192.0.2.6
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# activate

UTD Snort IPS Drop Log
============================
2016/06/13-14:32:09.524475 IST [**] [Instance_ID: 1] [**] Drop [**]
[1:30561:1] BLACKLIST DNS request for known malware
domain domai.ddns2.biz - Win.Trojan.Beebone [**]
[Classification: A Network Trojan was Detected]
[Priority: 1] [VRF_ID: 2] {UDP} 11.1.1.10:58016 -> 21.1.1.10:53
```

Example: Configuring Logging IOS Syslog

The following example shows how to configure logging IOS syslog with the log levels on a device:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# logging syslog
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-engstd-insp)# logging level debug
Device(config-utd-eng-std-insp)# end
Device#
```

Example: Configuring Logging to Centralized Log Server

The following example shows how to configure logging to a centralized log server:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std-insp)# logging host syslog.yourcompany.com
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# logging level info
Device(config-utd-eng-std-insp)# end
Device#
```

Example: Configuring Signature Update from a Cisco Server

The following example shows how to configure the signature update from a Cisco server :

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# signature update server cisco username CCOuser password
passwd123
Device(config-utd-eng-std-insp)# end
Device#
```

Note Ensure that the DNS is configured to download signatures from the Cisco server.

Example: Configuring Signature Update from a Local Server

The following example shows how to configure the signature update from a local server:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# signature update server url http://192.168.1.2/sig-1.pkg
Device(config-utd-eng-std-insp)# end
Device#
```

Example: Configuring Automatic Signature Update

The following example shows how to configure the automatic signature update on a server:

```
Device# configure terminal
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# signature update occur-at daily 0 0
Device(config-utd-eng-std-insp)# signature update server cisco username abcd password
cisco123
Device(config-utd-eng-std-insp)# end
Device#
```

Example: Performing Manual Signature Update

The following examples show how to perform a manual signature update in different ways:

```
Device# utd threat-inspection signature update
```

It takes the existing server configuration to download from or the explicit server information configured with it. These commands perform a manual signature update with the below settings:

```
Device# show utd engine standard threat-inspection signature update status

Current signature package version: 2983.4.s
Current signature package name: UTD-STD-SIGNATURE-2983-4-S.pkg
Previous signature package version: 29.0.c
---------------------------------------
Last update status: Successful
---------------------------------------
Last successful update time: Mon Aug  7 02:02:32 2017 UTC
Last successful update method: Manual
Last successful update server: cisco
Last successful update speed: 3022328 bytes in 25 secs
---------------------------------------
Last failed update time: Mon Aug  7 01:53:21 2017 UTC
Last failed update method: Manual
Last failed update server: cisco
Last failed update reason: ('Connection aborted.', gaierror(-2, 'Name or service hnot
known'))
---------------------------------------
Last attempted update time: Mon Aug  7 02:02:32 2017 UTC
Last attempted update method: Manual
Last attempted update server: cisco
---------------------------------------
Total num of updates successful: 1
Num of attempts successful: 1
Num of attempts failed: 3
Total num of attempts: 4
---------------------------------------
Next update scheduled at: None
---------------------------------------
Current status: Idle
```

```
Device# utd threat-inspection signature update server cisco username ccouser password
passwd123
```

```
Device# utd threat-inspection signature update server url http://192.168.1.2/sig-1.pkg
```

Example: Configuring Signature Allowed Lists

The following example shows how to configure signature allowed list:

```
Device# configure terminal
Device(config)# utd threat-inspection whitelist
Device(config-utd-whitelist)# utd-whitelist)# generator id 1 signature id 23456 comment
"traffic from client x"
Device(config-utd-whitelist)# exit
Device(config)# utd engine standard
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-eng-std-insp)# whitelist
Device(config-utd-eng-std-insp)# end
Device#
```

Note After the allowed list signature ID is configured, Snort will allow the flow to pass through the device without any alerts and drops.

Examples for Displaying Active Signatures

Example: Displaying Active Signatures List With Connectivity Policy

```
Device# utd threat-inspection signature active-list write-to bootflash:siglist_connectivity
Device# more bootflash:siglist_connectivity
==================================================================================
Signature Package Version: 2982.1.s
Signature Ruleset: Connectivity
Total no. of active signatures: 581
Total no. of drop signatures: 452
Total no. of alert signatures: 129

For more details of each signature please go to www.snort.org/rule_docs to lookup
==================================================================================
List of Active Signatures:
--------------------------
<snipped>
```

Example: Displaying Active Signatures List With Balanced Policy

```
Device# utd threat-inspection signature active-list write-to bootflash:siglist_balanced
Device# more bootflash:siglist_balanced
==================================================================================
Signature Package Version: 2982.1.s
Signature Ruleset: Balanced
Total no. of active signatures: 7884
Total no. of drop signatures: 7389
Total no. of alert signatures: 495

For more details of each signature please go to www.snort.org/rule_docs to lookup
==================================================================================

List of Active Signatures:
--------------------------
<snipped>
```

Example: Displaying Active Signatures List With Security Policy

```
Device# utd threat-inspection signature active-list write-to bootflash:siglist_security
Device# more bootflash:siglist_security
=================================================================================
Signature Package Version: 2982.1.s
Signature Ruleset: Security
Total no. of active signatures: 11224
Total no. of drop signatures: 10220
Total no. of alert signatures: 1004

For more details of each signature please go to www.snort.org/rule_docs to lookup
=================================================================================

List of Active Signatures:
--------------------------
<snipped>
```

Verifying the Integrated Snort IPS Configuration

Use the following commands to troubleshoot your configuration.

SUMMARY STEPS

1. **enable**
2. **show virtual-service list**
3. **show virtual-service detail**
4. **show service-insertion type utd service-node-group**
5. **show service-insertion type utd service-context**
6. **show utd engine standard config**
7. **show utd engine standard status**
8. **show utd engine standard threat-inspection signature update status**
9. **show utd engine standard logging events**
10. **clear utd engine standard logging events**
11. **show platform hardware qfp active feature utd config**
12. **show platform software utd global**
13. **show platform software utd interfaces**
14. **show platform hardware qfp active feature utd stats**
15. **show utd engine standard statistics daq all**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show virtual-service list**

Displays the status of the installation of all applications on the virtual service container.

Example:

```
Device# show virtual-service list

Virtual Service List:


Name                    Status            Package Name
------------------------------------------------------------------------------
UTDIPS                  Activated         utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
```

Step 3 **show virtual-service detail**

Displays the resources used by applications installed in the virtual services container of a device.

Example:

```
Device# show virtual-service detail


Device#show virtual-service detail
Virtual service UTDIPS detail
  State                 : Activated
  Owner                 : IOSd
  Package information
    Name                : utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Path                : bootflash:/utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Application
      Name              : UTD-Snort-Feature
      Installed version : 1.0.1_SV2982_XE_16_3
      Description       : Unified Threat Defense
    Signing
      Key type          : Cisco development key
      Method            : SHA-1
    Licensing
      Name              : Not Available
      Version           : Not Available

  Detailed guest status

----------------------------------------------------------------------
Process               Status            Uptime           # of restarts
----------------------------------------------------------------------
climgr                 UP         0Y 0W 0D  0: 0:35        1
logger                 UP         0Y 0W 0D  0: 0: 4        0
snort_1                UP         0Y 0W 0D  0: 0: 4        0
Network stats:
eth0: RX  packets:43, TX  packets:6
eth1: RX  packets:8, TX  packets:6

Coredump file(s): lost+found

  Activated profile name: None
  Resource reservation
    Disk                : 736 MB
    Memory              : 1024 MB
    CPU                 : 25% system CPU

  Attached devices
```

```
    Type              Name        Alias
    ---------------------------------------------
    NIC               ieobc_1     ieobc
    NIC               dp_1_0      net2
    NIC               dp_1_1      net3
    NIC               mgmt_1      mgmt
    Disk              _rootfs
    Disk              /opt/var
    Disk              /opt/var/c
    Serial/shell                  serial0
    Serial/aux                    serial1
    Serial/Syslog                 serial2
    Serial/Trace                  serial3
    Watchdog          watchdog-2

  Network interfaces
    MAC address             Attached to interface
    ------------------------------------------------------
    54:0E:00:0B:0C:02       ieobc_1
    A4:4C:11:9E:13:8D       VirtualPortGroup0
    A4:4C:11:9E:13:8C       VirtualPortGroup1
    A4:4C:11:9E:13:8B       mgmt_1

  Guest interface
  ---
  Interface: eth2
  ip address: 48.0.0.2/24
Interface: eth1
  ip address: 47.0.0.2/24

  ---

  Guest routes
  ---
  Address/Mask                         Next Hop                          Intf.
-------------------------------------------------------------------------------
0.0.0.0/0                            48.0.0.1                          eth2
0.0.0.0/0                            47.0.0.1                          eth1

  ---

  Resource admission (without profile) : passed
    Disk space    : 710MB
    Memory        : 1024MB
    CPU           : 25% system CPU
    VCPUs         : Not specified
```

Step 4 **show service-insertion type utd service-node-group**

Displays the status of service node groups.

Example:

```
Device# show service-insertion type utd service-node-group

Service Node Group name : utd_sng_1
Service Context : utd/1
Member Service Node count : 1


Service Node (SN) : 30.30.30.2
Auto discovered : No
SN belongs to SNG : utd_sng_1
Current status of SN : Alive
```

```
Time current status was reached : Tue Jul 26 11:57:48 2016

Cluster protocol VPATH version : 1
Cluster protocol incarnation number : 1
Cluster protocol last sent sequence number : 1469514497
Cluster protocol last received sequence number: 1464
Cluster protocol last received ack number : 1469514496
```

Step 5 **show service-insertion type utd service-context**

Displays the AppNav and service node views.

Example:

```
Device# show service-insertion type utd service-context

Service Context : utd/1
Cluster protocol VPATH version : 1
Time service context was enabled : Tue Jul 26 11:57:47 2016
Current FSM state : Operational
Time FSM entered current state : Tue Jul 26 11:57:58 2016
Last FSM state : Converging
Time FSM entered last state : Tue Jul 26 11:57:47 2016
Cluster operational state : Operational

Stable AppNav controller View:
30.30.30.1

Stable SN View:
30.30.30.2

Current AppNav Controller View:
30.30.30.1

Current SN View:
30.30.30.2
```

Step 6 **show utd engine standard config**

Displays the unified threat defense (UTD) configuration.

Example:

```
Device# show utd engine standard config


UTD Engine Standard Configuration:
  Operation Mode : Intrusion Prevention
  Policy         : Security

Signature Update:
  Server    : cisco
  User Name : ccouser
  Password  : YEX^SH\fhdOeEGaOBIQAIcOVLgaVGf
  Occurs-at : weekly ;  Days:0 ; Hour: 23; Minute: 50

Logging:
  Server    :   IOS Syslog; 10.104.49.223
  Level     : debug

Whitelist Signature IDs:
  28878
```

Step 7 **show utd engine standard status**

Displays the status of the utd engine.

Example:

```
Device# show utd engine standard status

Profile : High
System memory :
Usage : 8.00 %
Status : Green
Number of engines : 4

Engine Running CFT flows Health Reason
=======================================================
Engine(#1): Yes 0 Green None
Engine(#2): Yes 0 Green None
Engine(#3): Yes 0 Green None
Engine(#4): Yes 0 Green None
=======================================================

Overall system status: Green

Signature update status:
=========================
Current signature package version: 2983.4.s
Last update status: Successful
Last successful update time: Mon Aug  7 02:02:32 2017 UTC
Last failed update time: Mon Aug  7 01:53:21 2017 UTC
Last failed update reason: ('Connection aborted.', gaierror(-2, 'Name or service not known'))
Next update scheduled at: None
Current status: Idle
```

Step 8 **show utd engine standard threat-inspection signature update status**

Displays the status of the signature update process.

Example:

```
Device# show utd engine standard threat-inspection signature update status

Current signature package version: 2983.4.s
Current signature package name: UTD-STD-SIGNATURE-2983-4-S.pkg
Previous signature package version: 29.0.c
---------------------------------------
Last update status: Successful
---------------------------------------
Last successful update time: Mon Aug  7 02:02:32 2017 UTC
Last successful update method: Manual
Last successful update server: cisco
Last successful update speed: 3022328 bytes in 25 secs
---------------------------------------
Last failed update time: Mon Aug  7 01:53:21 2017 UTC
Last failed update method: Manual
Last failed update server: cisco
Last failed update reason: ('Connection aborted.', gaierror(-2, 'Name or service hnot known'))
---------------------------------------
Last attempted update time: Mon Aug  7 02:02:32 2017 UTC
Last attempted update method: Manual
Last attempted update server: cisco
---------------------------------------
Total num of updates successful: 1
```

```
Num of attempts successful: 1
Num of attempts failed: 3
Total num of attempts: 4
---------------------------------------
Next update scheduled at: None
---------------------------------------
Current status: Idle
```

Step 9 **show utd engine standard logging events**

Displays log events from the Snort sensor.

Example:

```
Device# show utd engine standard logging events

2016/06/13-14:32:09.524475 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected] [Priority: 1]
[VRF_ID: 2] {UDP} 11.1.1.10:58016 -> 21.1.1.10:53
2016/06/13-14:32:21.524988 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected] [Priority: 1]
[VRF_ID: 2] {UDP} a000:0:0:0:0:0:0:10:59964 -> b000:0:0:0:0:0:0:10:53
```

Step 10 **clear utd engine standard logging events**

Example:

```
Device# clear utd engine standard logging events
```

Clears logged events from the Snort sensor.

Step 11 **show platform hardware qfp active feature utd config**

Displays information about the health of the service node.

Example:

```
Device# show platform hardware qfp active feature utd config

Global configuration
NAT64: disabled
SN threads: 12
CFT inst_id 0 feat id 1 fo id 1 chunk id 8
Context Id: 0, Name: Base Security Ctx
Ctx Flags: (0x60000)
Engine: Standard
SN Redirect Mode : Fail-open, Divert
Threat-inspection: Enabled, Mode: IDS
Domain Filtering : Not Enabled
URL Filtering : Not Enabled
SN Health: Green
```

Step 12 **show platform software utd global**

Displays the interfaces on which UTD is enabled.

Example:

```
Device# show platform software utd global

UTD Global state
Engine               : Standard
```

```
Global Inspection    : Enabled
Operational Mode     : Intrusion Prevention
Fail Policy          : Fail-open
Container techonlogy : LXC
Redirect interface   : VirtualPortGroup1
UTD interfaces
All dataplane interfaces
```

Step 13 **show platform software utd interfaces**

Displays the information about all interfaces.

Example:

```
Device# show platform software utd interfaces

UTD interfaces
All dataplane interfaces
```

Step 14 **show platform hardware qfp active feature utd stats**

Displays dataplane UTD statistics.

Example:

```
Device# show platform hardware qfp active feature utd stats

Security Context:    Id:0    Name: Base Security Ctx

Summary Statistics:
Pkts entered policy feature                         pkt                  228
                                                    byt                31083

Drop Statistics:

Service Node flagged flow for dropping                                    48
Service Node not healthy                                                  62

General Statistics:

Non Diverted Pkts to/from divert interface                             32913
Inspection skipped - UTD policy not applicable                         48892
Policy already inspected                                                2226
Pkts Skipped - L2 adjacency glean                                          1
Pkts Skipped - For Us                                                     67
Pkts Skipped - New pkt from RP                                           102
Response Packet Seen                                                     891
Feature memory allocations                                               891
Feature memory free                                                      891
Feature Object Delete                                                    863

Service Node Statistics:
SN Health: Green
SN down                                                                   85
SN health green                                                           47
SN health red                                                             13

Diversion Statistics
redirect                                                                2226
encaps                                                                  2226
decaps                                                                  2298
reinject                                                                2250
decaps: Could not locate flow                                             72
```

```
Redirect failed, SN unhealthy                                             62
Service Node requested flow bypass drop                                   48
```

Step 15 show utd engine standard statistics daq all

Displays serviceplane data acquistion (DAQ) statistics.

Example:

```
Device# show utd engine standard statistics daq all

IOS-XE DAQ Counters(Engine #1):
---------------------------------
Frames received                         :0
Bytes received                          :0
RX frames released                      :0
Packets after vPath decap               :0
Bytes after vPath decap                 :0
Packets before vPath decap              :0
Bytes before vPath decap                :0
Frames transmitted                      :0
Bytes transmitted                       :0

Memory allocation                       :2
Memory free                             :0
Merged packet buffer allocation         :0
Merged packet buffer free               :0

VPL buffer allocation                   :0
VPL buffer free                         :0
VPL buffer expand                       :0
VPL buffer merge                        :0
VPL buffer split                        :0
VPL packet incomplete                   :0

VPL API error                           :0
CFT API error                           :0
Internal error                          :0
External error                          :0
Memory error                            :0
Timer error                             :0

Kernel frames received                  :0
Kernel frames dropped                   :0

FO cached via timer                     :0
Cached fo used                          :0
Cached fo freed                         :0
FO not found                            :0
CFT full packets                        :0


VPL Stats(Engine #1):
------------------------
```

Deploying Snort IPS Using Cisco Prime CLI Templates

You can use the Cisco Prime CLI templates to provision the Snort IPS deployment. The Cisco Prime CLI templates make provisioning Snort IPS deployment simple. To use the Cisco Prime CLI templates to provision the Snort IPS deployment, perform these steps:

Step 1 Download the Prime templates from the Software Download page, corresponding to the IOS XE version running on your system.

Step 2 Unzip the file, if it is a zipped version.

Step 3 From Prime, choose **Configuration** > **Templates** > **Features and Technologies**, select **CLI Templates**.

Step 4 Click **Import**.

Step 5 Select the folder where you want to import the templates to and click **Select Templates** and choose the templates that you just downloaded to import.

The following Snort IPS CLI templates are available:

- Copy OVA to Device—Use this template to copy the Snort IPS OVA file to the router file system.
- Delete OVA—Use this template to delete the copied Snort IPS OVA file from the router file system.
- Dynamic NAT—Use this template if Dynamic NAT (Network Address Translation) is configured in your environment and an Access List is used to select the NAT translation that needs to be modified for Snort IPS Management Interface IP.
- Dynamic NAT Cleanup—Use this template to delete the NAT configuration for Snort IPS.
- Dynamic PAT—Use this template if Dynamic PAT (Port Address Translation) is configured in your environment and an Access List is used to select the PAT translation that needs to be modified for Snort IPS Management Interface IP.
- Dynamic PAT Cleanup—Use this template to delete the PAT configuration for Snort IPS.
- IP Unnumbered—Use this template to configure Snort IPS and required Virtual-Service for IP Unnumbered deployment.
- IP Unnumbered Cleanup—Use this template to delete the configured Snort IPS Management interface with IP Unnumbered.
- Management Interface—Use this template if you would like to use System Management interface (e.g. GigabitEthernet0) to route Snort IPS Management traffic.
- Management Interface Cleanup—Use this template to delete the configured System Management interface (e.g. GigabitEthernet0) to route the Snort IPS Management traffic.
- Static NAT—Use this template to configure Snort IPS and required Virtual-Service for existing Static NAT deployment.
- Static NAT Cleanup—Use this template to delete the configured Snort IPS in a Static NAT deployment.
- Upgrade OVA—Use this template to upgrade Snort IPS OVA file.

Migrating to IOx Container

This section provides information about Cisco IOx and UTD migration to IOx for extending UTD support on Cisco 1000 Series Integrated Service Routers (ISRs). Cisco IOx combines Cisco IOS and the Linux OS for highly secure networking.

About Cisco IOx

Cisco IOx is an application platform that provides uniform and consistent hosting capabilities for various types of applications across various Cisco platforms. This platform brings together the networking operating system-Cisco IOS, and the open source platform-Linux to bring together custom applications and interfaces on the network.

A virtual services container is a virtualized environment on a device. It is also referred to as a virtual machine (VM), virtual service, or container. You can install an application within a virtual services container. The application runs in the virtual services container of the operating system of a device. The application is delivered as an open virtual application (OVA), which is a tar file with a .ova extension. The OVA package is installed and enabled on a device through a command-line interface. Cisco Plug-in for OpenFlow is an example of an application that can be deployed within a virtual services container.

Virtual services container infrastructure that is used to host UTD OVA is not supported on Cisco 1100 Series ISRs. Currently, UTD supports both the containers. However, the OVA container feature support is continued on Cisco IOS XE Gibralter 16.10 release and is not supported for later releases.

Upgrading from Virtual Service Container to IOx

An OVA file is an Open Virtualization Archive that contains a compressed, installable version of a virtual machine. The Snort IPS is available as a virtual container service. You must download this OVA file on to the device and use the **virtual-service install** CLI to install the service.

For the UTD IOx infrastructure, the IOx based OVA is installed using IOx CLI commands. Before installing, start the IOx environment in global configuration mode.

The IOx based OVA is called a TAR file. You must use a Cisco IOS XE image with security license. During the OVA file installation, the security license is checked and an error is reported if the license is not present.

Perform the following steps to upgrade from virtual service to IOx container:

Step 1 **no activate**

Example:

```
Device# configure terminal
Device (config)# virtual-service utd
Device (config-virt-serv)# no activate
Device (config-virt-serv)# exit
Device (config)# no virtual-service utd
```

Deactivates virtual manager based virtual-service instance.

Step 2 **show virtual-service list**

Example:

```
Device# show virtual-service list
```

Displays the status of all applications installed on the virtual service container. Ensure that virtual service instance is deactivated.

Step 3 **virtual-service uninstall name** *virtual-service instance*

Example:

```
Device# virtual-service uninstall name utd
```

Uninstall virtual manager based virtual-service instance. Ensure that virtual service instance does not show up when you run **show virtual-service list** command.

Step 4 **iox**

Example:

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

Starts the IOx environment in Global Configuration mode.

Step 5 **app-hosting install appid** *name* **package** *bootflash:<tarfile>*

Example:

```
Device# app-hosting install appid UTD package bootflash:utd.tar
Device#
```

Copies and installs Iox based OVA tar file on to the device.

Step 6 **show app-hosting list**

Example:

```
Device# show app-hosting list
App id                                   State
---------------------------------------------------------
UTD                                      DEPLOYED
Device#
```

Displays the status of the installation. Ensure that the application is deployed.

Step 7 **app-hosting activate appid** *name*

Example:

```
Device# app-hosting activate appid UTD
```

Activates the IOx based TAR file on the device.

Step 8 **show app-hosting list**

Example:

```
Device# show app-hosting list
App id                                   State
---------------------------------------------------------
UTD                                      ACTIVATED

Device#
```

Displays the status of the activation. Ensure that the application is activated.

Step 9 **app-hosting start appid** *name*

Example:

```
Device# app-hosting start appid UTD
Device# show app-hosting list | in UTD
```

Starts the IOx based OVA.

Step 10 **show app-hosting list**

Example:

```
Example:
Device# show app-hosting list
App id                                   State
---------------------------------------------------------
UTD                                      RUNNING

Device#
```

Displays the status of the start. Ensure that the application is running.

Example of IOx Configuration

Following is the example configuration of IOx:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# iox
Device(config)# interface VirtualPortGroup0
Device(config-if)# no shutdown
Device(config-if)# ip address 192.0.2.1 255.255.255.252
Device(config-if)# exit
Device(config)# interface VirtualPortGroup1
Device(config-if)# no shutdown
Device(config-if)# ip address 192.0.2.5 255.255.255.252
Device(config-if)# exit
Device(config)# app-hosting appid utd
Device(config-app-hosting)# app-vnic gateway0 virtualportgroup 0 guest-interface 0
Device(config-app-hosting-gateway0)# guest-ipaddress 192.0.2.2 netmask 255.255.255.252
Device(config-app-hosting-gateway0)# exit
Device(config-app-hosting)# app-vnic gateway1 virtualportgroup 1 guest-interface 1
Device(config-app-hosting-gateway1)# guest-ipaddress 192.0.2.6 netmask 255.255.255.252
Device(config-app-hosting-gateway1)# exit
Device(config-app-hosting)# app-resource package-profile custom
Device(config-app-hosting)# start
Device(config-app-hosting)# exit
Device(config)# exit
Device#
```

Troubleshooting Snort IPS

Traffic is not Diverted

Problem Traffic is not diverted.

Possible Cause Vitual-service may not be activated.

Solution Check whether the virtual-service is activated by using the **show virtual-service list** command. The following is sample output from the command:

```
Device# show virtual-service list

Virtual Service List:


Name Status Package Name
------------------------------------------------------------------------------
snort Activated utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
```

Possible Cause Unified threat defense (UTD) may not be enabled for specified interface or interfaces.

Solution Use the **show platform software utd global** command to verify if UTD is enabled for the interface:

```
Device# show platform software utd global

UTD Global state
Engine               : Standard
Global Inspection    : Disabled
Operational Mode     : Intrusion Prevention
Fail Policy          : Fail-open
Container techonlogy : LXC
Redirect interface   : VirtualPortGroup1
UTD interfaces
GigabitEthernet0/0/0
```

Possible Cause The service node may not be working properly.

Solution Use the **show platform hardware qfp active feature utd config** command to verify if the health of the service node is green:

```
Device# show platform hardware qfp active feature utd config

Global configuration
NAT64: disabled
SN threads: 12
CFT inst_id 0 feat id 0 fo id 0 chunk id 4
Context Id: 0, Name: Base Security Ctx
Ctx Flags: (0x60000)
Engine: Standard
SN Redirect Mode : Fail-open, Divert
Threat-inspection: Enabled, Mode: IDS
Domain Filtering : Not Enabled
URL Filtering : Not Enabled
SN Health: Green
```

Possible Cause The Snort process may not be activated.

Solution Use the **show virtual-service detail** command to verify if the Snort process is up and running:

```
Device# show virtual-service detail

Virtual service UTDIPS detail
  State                 : Activated
  Owner                 : IOSd
  Package information
    Name                : utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Path                : bootflash:/utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Application
      Name              : UTD-Snort-Feature
      Installed version : 1.0.1_SV2982_XE_16_3
      Description       : Unified Threat Defense
    Signing
      Key type          : Cisco development key
      Method            : SHA-1
    Licensing
      Name              : Not Available
      Version           : Not Available

  Detailed guest status

----------------------------------------------------------------------
Process               Status            Uptime           # of restarts
----------------------------------------------------------------------
climgr                 UP         0Y 0W 0D  0: 0:35        1
logger                 UP         0Y 0W 0D  0: 0: 4        0
snort_1                UP         0Y 0W 0D  0: 0: 4        0
Network stats:
eth0: RX  packets:43, TX  packets:6
eth1: RX  packets:8, TX  packets:6

Coredump file(s): lost+found

  Activated profile name: None
  Resource reservation
    Disk                : 736 MB
    Memory              : 1024 MB
    CPU                 : 25% system CPU

  Attached devices
    Type              Name        Alias
    ---------------------------------------------
    NIC               ieobc_1     ieobc
    NIC               dp_1_0      net2
    NIC               dp_1_1      net3
    NIC               mgmt_1      mgmt
    Disk              _rootfs
    Disk              /opt/var
    Disk              /opt/var/c
    Serial/shell                  serial0
    Serial/aux                    serial1
    Serial/Syslog                 serial2
    Serial/Trace                  serial3
    Watchdog          watchdog-2

  Network interfaces
    MAC address             Attached to interface
    ------------------------------------------------------
    54:0E:00:0B:0C:02       ieobc_1
    A4:4C:11:9E:13:8D       VirtualPortGroup0
    A4:4C:11:9E:13:8C       VirtualPortGroup1
    A4:4C:11:9E:13:8B       mgmt_1

  Guest interface
  ---
  Interface: eth2
  ip address: 48.0.0.2/24
Interface: eth1
  ip address: 47.0.0.2/24

  ---

  Guest routes
  ---
  Address/Mask                         Next Hop                          Intf.
-------------------------------------------------------------------------------
0.0.0.0/0                           48.0.0.1                          eth2
0.0.0.0/0                           47.0.0.1                          eth1

  ---

  Resource admission (without profile) : passed
    Disk space    : 710MB
    Memory        : 1024MB
    CPU           : 25% system CPU
    VCPUs         : Not specified
```

Possible Cause The AppNav tunnel may not be activated.

Solution Use the **show service-insertion type utd service-node-group** and **show service-insertion type utd service-context** commands to verify if the AppNav tunnel is activated.

Solution The following is sample output from the **show service-insertion type utd service-node-group** command:

```
Device# show service-insertion type utd service-node-group

Service Node Group name : utd_sng_1
Service Context : utd/1
Member Service Node count : 1


Service Node (SN) : 30.30.30.2
Auto discovered : No
SN belongs to SNG : utd_sng_1
Current status of SN : Alive
Time current status was reached : Tue Jul 26 11:57:48 2016

Cluster protocol VPATH version : 1
Cluster protocol incarnation number : 1
Cluster protocol last sent sequence number : 1469514497
Cluster protocol last received sequence number: 1464
Cluster protocol last received ack number : 1469514496
```

Solution The following is sample output from the **show service-insertion type utd service-context** command:

```
Device# show service-insertion type utd service-context

Service Context : utd/1
Cluster protocol VPATH version : 1
Time service context was enabled : Tue Jul 26 11:57:47 2016
Current FSM state : Operational
Time FSM entered current state : Tue Jul 26 11:57:58 2016
Last FSM state : Converging
Time FSM entered last state : Tue Jul 26 11:57:47 2016
Cluster operational state : Operational

Stable AppNav controller View:
30.30.30.1

Stable SN View:
30.30.30.2

Current AppNav Controller View:
30.30.30.1
```

```
Current SN View:
30.30.30.2
```

Possible Cause Check data plane UTD statistics for the status of the traffic. If the traffic is not diverted, the number of packets diverted and rejected will be zero. If the numbers are nonzero, then traffic diversion is happening, and the Snort sensor is resending packets back to the dataplane.

Solution Use the **show platform hardware qfp active feature utd stats** commands to verify the status of the traffic.

```
Device# show platform hardware qfp active feature utd stats

Security Context:    Id:0    Name: Base Security Ctx

Summary Statistics:
Active Connections                                                            29
TCP Connections Created                                                   712910
UDP Connections Created                                                       80
Pkts entered policy feature                           pkt                3537977
                                                      byt              273232057
Pkts entered divert feature                           pkt                3229148
                                                      byt              249344841
Pkts slow path                                        pkt                 712990
                                                      byt               45391747
Pkts Diverted                                         pkt                3224752
                                                      byt              249103697
Pkts Re-injected                                      pkt                3224746
                                                      byt              249103373
….
```

Signature Update is not Working

Problem Signature update from Cisco Borderless Software Distribution (BSD) server is not working.

Possible Cause Signature update may have failed due to various reasons. Check for the reason for the last failure to update the signatures.

Solution Use the **show utd engine standard threat-inspection signature update status** command to display the reason for the last failure to update the signatures:

```
Device# show utd eng standard threat-inspection signature update status
Current signature package version: 29.0.c
Current signature package name: default
Previous signature package version: None
---------------------------------------
Last update status: Failed
---------------------------------------
Last successful update time: None
Last successful update method: None
Last successful update server: None
Last successful update speed: None
---------------------------------------
Last failed update time: Thu Jan 11 13:34:36 2018 PST
Last failed update method: Manual
Last failed update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
Last failed update reason: [Errno 113] No route to host
---------------------------------------
```

```
Last attempted update time: Thu Jan 11 13:34:36 2018 PST
Last attempted update method: Manual
Last attempted update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
---------------------------------------
Total num of updates successful: 0
Num of attempts successful: 0
Num of attempts failed: 1
Total num of attempts: 1
---------------------------------------
Next update scheduled at: None
---------------------------------------
Current status: Idle
```

Possible Cause Domain Name System (DNS) is not configured correctly.

Solution Use the **show running-config | i name-server** command to display the name server details:

```
Device#  show run | i name-server

ip name-server 10.104.49.223
```

Possible Cause System error—Failed to process the username and password combination.

Solution Ensure that you have provided the correct credentials for signature package download.

Signature Update from the Local Server is not Working

Problem Signature update from the local server not working.

Possible Cause Last failure Reason: Invalid scheme—only HTTP/HTTPS supported.

Solution Ensure that you have provided the HTTP or secure HTTP (HTTPS) as the local download method.

Possible Cause Last failure Reason: Name or service not known.

Solution Ensure that the hostname or IP address provided for the local server is correct.

Possible Cause Last failure Reason: Credentials not supplied.

Solution Ensure that you have provided the credentials for local HTTP/HTTPS server.

Possible Cause Last failure Reason: File not found.

Solution Ensure that the signature file name or URL that you have provided is correct.

Possible Cause Last failure Reason: Download corrupted.

Solution

- Verify whether the retry signature update is corrupted as the previous signature download.
- Ensure that the correct signature package is available.

Logging to IOSd Syslog is not Working

Problem Logging to IOSd syslog is not working.

Possible Cause Logging to syslog may not be configured in the unified threat defense (UTD) configuration.

Solution Use the **show utd engine standard config** command to display the UTD configuration and to ensure that logging to syslog is configured.

```
Device# show utd engine standard config

UTD Engine Standard Configutation:
  Operation Mode : Intrusion Prevention
  Policy         : Security

Signature Update:
  Server    : cisco
  User Name : ccouser
  Password  : YEX^SH\fhdOeEGaOBIQAIcOVLgaVGf
  Occurs-at : weekly ;  Days:0 ; Hour: 23; Minute: 50

Logging:
  Server    :   IOS Syslog; 10.104.49.223
  Level     : debug

Whitelist Signature IDs:
  28878
```

Solution Use the following **show utd engine standard logging events** command to display the event logs for the UTD engine.

```
Device# show utd engine standard logging events

2016/06/13-14:32:09.524475 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected]
[Priority: 1] [VRF_ID: 2] {UDP} 11.1.1.10:58016 -> 21.1.1.10:53
2016/06/13-14:32:21.524988 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected] [Priority: 1]
[VRF_ID: 2] {UDP} a000:0:0:0:0:0:0:10:59964 -> b000:0:0:0:0:0:0:10:53
```

Logging to an External Server is not Working

Problem Logging to an external server is not working.

Possible Cause Syslog may not be running on the external server.

Solution Verify whether syslog server is running on the external server. Configure the following command on the external server to view its status:

```
ps -eaf | grep syslog

 root 2073 1 0 Apr12 ? 00:00:02 syslogd -r -m
```

Possible Cause Connectivity between unified threat defense (UTD) Linux Container (LXC) and external server may be lost.

Solution Verify the connectivity from the management interface to the external syslog server.

UTD Conditional Debugging

Conditional debugging is supported by multi-tenancy for Unified Threat Defense. For further details about how to configure conditional debugging, see:

http://www.cisco.com/c/en/us/td/docs/routers/asr1000/troubleshooting/guide/Tblshooting-xe-3s-asr-1000-book.html#task_AC969BB06B414DCBBDEF7ADD29EF8131

Additional References for Snort IPS

Related Documents

Related Topic	Document Title
IOS commands	Cisco IOS Master Command List, All Releases
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

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 Snort IPS

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 3: Feature Information for Snort IPS

Feature Name	Releases	Feature Information
Snort IPS	Cisco IOS XE 3.16.1S, 3.17S and later releases	The Snort IPS feature, enables Intrusion Prevention System (IPS) and Intrusion Detection System (IDS) for branch offices on Cisco IOS XE-based platforms. This feature uses the open source Snort solution to enable IPS and IDS.
VRF support on Snort IPS	Cisco IOS XE Denali 16.3.1	Supports Virtual Fragmentation Reassembly (VFR) on Snort IPS configuration.
Snort IPS support on Cisco Cloud Services Router 1000v Series	Cisco IOS XE Denali 16.3.1	Cisco Cloud Services Router 1000v Series supports Snot IPS.
UTD Snort IPS Enhancements for 16.4 Release	Cisco IOS XE Everest 16.4.1	The UTD Snort IPS enhancements for 16.4 release adds a feature for displaying the list of active signatures.
Threat Inspection Alerts Visibility UTD Serviceability enhancements	Cisco IOS XE Fuji 16.8.1	This feature provides summary of threat inspection alerts. The following commands are introduced: • **show utd engine standard logging statistics threat-inspection** • **show utd engine standard logging statistics threat-inspection** *detail* Following commands are modified as part of UTD Serviceability Enahancement: • **show utd engine standard status** • **show utd engine standard threat-inspection signature update status**
UTD (IPS and URL filtering) migration to IOX Containers	Cisco IOS XE Gibraltar 16.10.1	UTD is supported on Cisco 1100 Series ISRs by migrating virtual service container to IOx from OVA.

CHAPTER 3

Web Filtering

The Web Filtering feature enables the user to provide controlled access to Internet websites or Interanet sites by configuring the domain-based or URL-based policies and filters on the device. The user can configure the web filtering profiles to manage the web access. The Web Filtering feature is implemented using the container service and it is similar to the Snort IPS solution.

Web Filtering can either allow or deny access to a specific domain or URL based on:

- Allowed list and Blocked list—These are static rules, which helps the user to either allow or deny domains or URLs. If the same pattern is configured under both the allowed list and blocked list, the traffic will be allowed.
- Category—URLs can be classified into multiple categories such as News, Social Media, Education, Adult and so on. Based on the requirements, user has the option to block or allow one or more categories.
- Reputation—Each URL has a reputation score associated with it. The reputation score range is from 0-100, and it is categorized as: high-risk (reputation score (0-20), suspicious (0-40), moderate-risk (0-60), low-risk (0-80), and trustworthy (0-100). Based on the reputation score of a URL and the configuration, a URL is either blocked or allowed. If the user defines a reputation threshold through the CLI, all the URLs, with a reputation score lower than the user-defined threshold will be blocked.

Web Filtering

The Web Filtering feature enables the user to provide controlled access to Internet websites by configuring the domain-based or URL-based policies and filters on the device. Domain-based Filtering enables the user to control access to websites/servers at domain level, and URL-based Filtering enables the user to control access to websites at URL level. This section includes the following topics:

Domain-based Filtering

Domain-based filtering allows the user to control access to a domain by permitting or denying access based on the domain-based policies and filters configured on the device. When the client sends a DNS request through the Cisco Cloud Services Router 1000V Series, the DNS traffic is inspected based on the domain-based policies (allowed list/blocked list). Domains that are on the allowed list or blocked list will not be subjected to URL-based filtering even if they are configured. Graylist traffic does not match both allowed list and blocked list, and it is subjected to URL-based filtering if it is configured.

Domain-based Filtering Using Allowed List Filter

To allow the complete domain (cisco.com) without subjecting to any filtering, use the allowed list option . When a user makes a request to access a website using a browser, the browser makes a DNS request to get the IP address of the website. Domain filtering applies the filter on the DNS traffic. If the website's domain name matches to one of the allowed list patterns, domain filtering adds the website's address to the allowed list. The browser receives the IP address for the website and sends the HTTP(s) request to the IP address of the website. Domain filtering treats this traffic as allowed traffic. This allowed traffic is not further subjected to URL-based filtering even if it is configured. If the Snort IPS is configured, the traffic will be subjected to Snort IPS .

Domain-based Filtering Using Blocked List Filter

When a user want to block a complete domain (badsite.com), use the blocked list option. Domain filtering applies the filter on the DNS traffic. If the website's domain name matches to one of the patterns on the blocked list, domain filtering will send the configured blocked server's IP address in the DNS response to the end user instead of the actual resolved IP address of the website. The browser receives the blocked server's IP address as the IP address for the website and sends the HTTP(s) request to this IP address. This traffic is not further subjected to URL filtering or Snort IPS even if they are configured.The block server receives the HTTP(s) request and serves a block page to the end user. Also, when the DNS request matches a blocked list, all application traffic to that domain will be blocked.

Domain filtering is applied to all the DNS traffic even if the DNS requests are made in the context of non-HTTP(S) requests such as FTP, telnet, and so on. The blocked listed non-HTTP(S) traffic (FTP, telnet, and so on.) will also be forwarded to the block server. It is block server's responsibility to serve a block page or deny the request. You can configure an internal or external block server. For configuration steps, see Configure Domain-based Web Filtering with an External Block Server, on page 76 and Configure Domain-based Web Filtering with a Local Block Server , on page 77.

If the traffic is not part of the allowed list or on the blocked list during domain filtering, it will be subjected to URL filtering and Snort IPS if they are configured.

A user may consider using a combination of domain filtering allowed and blocked pattern lists to design the filters. For example, if a user wants to create an allowed list *www\.foo\.com*but also wants other domains on a blocked list, such as *www\.foo\.abc* and *www\.foo\.xyz*, configure the *www\.foo\.com* in the allowed list pattern and *www\.foo\.* in the blocked list pattern.

URL-based Filtering

URL-based filtering allows a user to control access to Internet websites by permitting or denying access to specific websites based on the allowed list/blocked list, category, or reputation configuration. For example, when a client sends a HTTP/HTTP(s) request through the Cisco CSR 1000V Cloud Services Router, the HTTP/HTTP(s) traffic is inspected based on the URL filtering policies (Allowed list, Blocked list, Category, and Reputation). If the HTTP/HTTP(s) request matches the blocked list, the HTTP(s) request is blocked either by inline block page response or redirects the URL to a block server. If the HTTP/HTTP(s) request matches the allowed list, the traffic is allowed without further URL filtering inspection.

For HTTPS traffic, the inline block page will not be displayed. URL-based filtering will not decode any encoded URL before performing a lookup.

When there is no allowed list/blocked list configuration on the device, based on the category and reputation of the URL, traffic is allowed or blocked either using a block page or redirect URL for HTTP. For HTTP(s), there is no block page or redirect URL, the flow will be dropped.

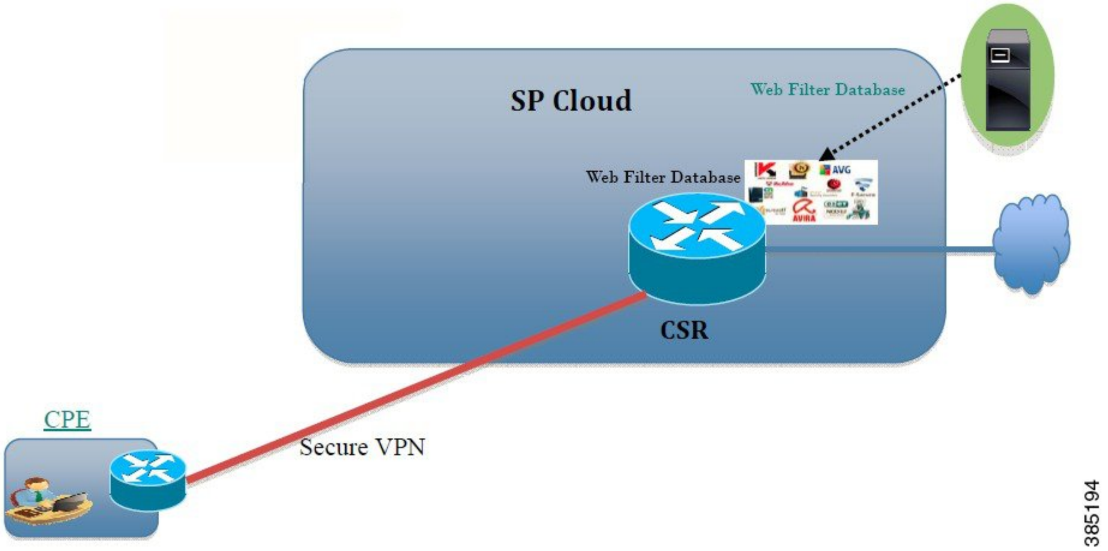
The URL database is downloaded from the cloud when the user configures the category/reputation-based URL filtering. The URL category/reputation database has only a few IP address based records and the category/reputation look up occurs only when the host portion of the URL has the domain name. After the full database is downloaded from the cloud, if there are any updates to the existing database, the incremental updates will be automatically downloaded in every 15 minutes. The complete database size is approximately 440 MB and the downloaded database should always synchronize with the cloud. The database will be invalid if the connection to the cloud is lost for more than 24 hours.

If the device does not get the database updates from the cloud, the fail-open option ensures that the traffic designated for URL filtering is not dropped. When you configure the fail-close option, all the traffic destined for URL filtering will be dropped when the cloud connectivity is lost.

Note The web filtering database is periodically updated from the cloud in every 15 minutes.

The figure illustrates the Web Filtering topology.

Figure 3: Web Filtering Network Topology



Virtual Service Resource Profiles for URL Filtering

The Cisco ISR 4000 Series Integrated Services Routers support *urlf-medium* and *urlf-high* resource profiles along with *urlf-low* profile. These profiles indicate the CPU and memory resources required to run the virtual service.

Platform	Profile	Virtual Service Resource Requirements		Platform Requirements
		System CPU	SP Memory	
CSR1000v, ISRv	*urlf-low*	25%	3 GB	8 GB (RAM)
	urlf-medium	50%	4 GB	8 GB (RAM)
	urlf-high	75%	6 GB	12 GB (RAM)

Cloud-Lookup

The Cloud-Lookup feature operates in single-tenancy mode to retrieve the category and reputation score of URLs that are not available in the local database. The Cloud-Lookup feature is enabled by default.

The Cloud-Lookup feature is an enhancement over the on-box database lookup feature. Earlier, the on-box database lookup feature allowed URLs that are not present in the on-box database and have a reputation score of 0. When Cloud-Lookup is enabled, the URLs that were allowed earlier may be dropped based on the reputation score and the configured block-threshold. In order to allow such URLs, one must add them to an allowed list. Category and reputation scores for different URLs from Cloud-Lookup are explained below.

There are two kinds of URLs:

- Name based URLs
- IP based URLs

When the Cloud-Lookup feature is enabled, the category and reputation score of unknown URLs are returned as follows:

Name based URLs

- Valid URL — corresponding category and reputation score is received.
- Unknown URL (new URL or unknown to the cloud) — category is 'uncategorized' and reputation score is 40
- Internal URLs with proper domain name (for example, internal.abc.com) — category and reputation score is based on the base domain name (abc.com from the example above).
- Completely internal URLs (for example, abc.xyz) — category is 'uncategorized' and reputation score is 40

IP based URLs

- Public hosted IP — corresponding category and reputation score is received.
- Private IP like 10.<>, 192.168.<> — category is 'uncategorized' and reputation score is 100
- Non-hosted/Non-routable IP — category is 'uncategorized' and reputation score is 40

The Cloud-Lookup score is different from the on-box database for these URLs (Unknown/Non-hosted/Non-routable/Internal URLs).

Note The Cloud-Lookup feature is not available in multi-tenancy mode.

Benefits of Web Filtering

The Web Filtering feature allows a user to provide controlled access to the internet by configuring domain and URL based policies and filters. It helps to secure the network by blocking malicious or unwanted websites.Web Filtering comprises of URL-based filtering and the Domain-based filtering. Domain-based filtering helps control access to websites/servers at domain level and the URL-based filtering helps control access to websites at URLs level. A user can use web filtering to add an individual URL to a blocked list or domain names and configure allowed listing policies for the same. A user can also provision to allow or block a URL based on reputation or category.

Prerequisites for Web Filtering

Before you configure the web filtering feature on the Cisco CSR 1000V Cloud Services Router, ensure that you have the following:

- The Cisco CSR 1000V Cloud Services Router runs the Cisco IOS XE Denali 16.3 software image or later.
- The Cisco CSR 1000V Cloud Services Router requires 2 vCPU, 8GB memory, and 2GB extra disk space for deploying the container service.
- The Cisco CSR 1000V Cloud Service Router must have a security K9 license to enable the web filtering feature.

Restrictions for Web Filtering

The following restrictions apply to the web filtering feature:

- This feature is only supported on Cisco CSR 1000V Cloud Services Router and it is not supported on Cisco 4000 Series Integrated Services Routers.
- The allowed list/blocked list pattern supports only regex pattern, and currently 64 patterns are supported for allowed list/blocked list. For more information on regex pattern, see the Regular Expressions chapter.
- Domain filtering supports only the IPv4 domains resolved through DNS protocol using IPv4 UDP transport. Domain filtering alerts are sent only to IOS syslog.
- Domain filtering with OpenDNS is not supported.
- URL filtering with Virtual Routing and Forwarding (VRF) is not supported.
- Domain filtering with CWS is not supported.
- Domain filtering does not support category and reputation.
- Local block server does not support serving HTTPS block page.When the URL filter tries to inject block page or redirect message, it does not support HTTPS traffic.
- When there is a username and password in the URL, URL filter does not remove them from the URL before matching the allowed list/blocked list pattern. However, the category/reputation lookup does not have this limitation and removes the username and password from the URL before lookup.

- HTTPS inspection is limited. Web filtering uses server certificate to obtain the URL/domain information. It is not possible to inspect the full URL path.
- UTD does not inter-operate with WCCP, and NBAR under inter-VRF scenario.
- Web filter profile names for URL, domain, block and sourcedb can have only alpha-numeric characters, dashes and underscores.
- If a virtual-service profile is modified, the virtual-service must be re-installed for the profile change to take effect.

How to Deploy Web Filtering

To deploy web filtering on supported devices, perform the following tasks:

Before you begin

- **Provision the device:** Identify the device to install the Web Filtering feature. This feature is supported on Cisco CSR 1000V Cloud Services Router.
- **Obtain the license:** The web filtering functionality is available only in security packages which require a security license to enable the service. Contact Cisco Support to obtain the license.

Step 1 Install and activate the virtual container service—How to Install and Activate the Virtual Container Service , on page 74

Step 2 Configure the domain-based web filtering with an external block server—Configure Domain-based Web Filtering with an External Block Server, on page 76

Step 3 Configure the domain-based web filtering with local block server—Configure Domain-based Web Filtering with a Local Block Server , on page 77

Step 4 Configure the URL-based web filtering with a local block server—Configure URL-based Web Filtering with a Local Block Server, on page 78

Step 5 Configure the URL-based web filtering with an Inline block server—Configure URL-based Web Filtering with an Inline Block Page, on page 80

Step 6 Configure the Snort IPS/IDS—Configuring Domain/URL based Web Filtering and Snort IPS, on page 82

How to Install and Activate the Virtual Container Service

To install and activate the virtual container service, perform the following task:

Step 1 Install the UTD OVA file—Installing the UTD OVA File, on page 75.

Step 2 Configure the VirtualPortGroup interfaces and virtual-service—Configuring VirtualPortGroup Interfaces and Virtual Service, on page 75.

Step 3 Activate the Snort virtual container service.

Installing the UTD OVA File

An OVA file is an Open Virtualization Archive that contains a compressed, installable version of a virtual machine. You must download this OVA file on to the router and use the virtual-service install CLI to install the service. The service OVA file is not bundled with the Cisco IOS XE Release images that are installed on the router. However, the OVA files may be preinstalled in the flash of the router.

You must use a Cisco IOS XE image with security license. During the OVA file installation, the security license is checked and an error is reported if the license is not present.

This is the sample configuration:

```
Device> enable
Device# virtual-service install name UTDIPS package harddisk:utd-ips-v102.ova media harddisk:

Device# show virtual-service list
Virtual Service List:
Name Status Package Name
------------------------------------------------------------------------------
snort Installed utdsnort.1_2_2_SV2982_XE_main.20160
```

Configuring VirtualPortGroup Interfaces and Virtual Service

You must configure two VirtualPortGroup interfaces and configure guest IP addresses for both interfaces.

Note The VirtualPortGroup interface for data traffic must use a private or nonroutable IP address. We recommend the use of 192.0.2.0/30 IP address range for this interface.

This is the sample configuration:

```
Device# configure terminal
evice(config)# interface VirtualPortGroup0
Device(config-if)# ip address 192.0.2.1 255.255.255.252
Device(config-if)# exit
Device(config)# interface VirtualPortGroup 1
Device(config-if)# ip address 192.0.2.5 255.255.255.252
Device(config-if)# exit
Device(config)# virtual-service UTDIPS

Device(config-virt-serv)# profile urlf-low (This is minimum requirement for web filtering
to work.)

Device(config-virt-serv)# vnic gateway VirtualPortGroup 0 (The IP-address configured in
VPG0 interface should have access to Internet over http(s).If the VPG0 interface does not
have access to Internet, the web filter database will not be updated.)
Device(config-virt-serv-vnic)# guest ip address 192.0.2.2
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# vnic gateway VirtualPortGroup 1
Device(config-virt-serv-vnic)# guest ip address 192.0.2.6
Device(config-virt-serv-vnic)# exit
Device(config-virt-serv)# activate
Device(config-virt-serv)# end

Device# show virtual-service list
Virtual Service List:

Name                    Status            Package Name
```

```
------------------------------------------------------------------------------
snort                   Activated          utdsnort.1_2_2_SV2982_XE_main.20160
```

Configure Domain-based Web Filtering with an External Block Server

To configure domain-based web filtering with an external block server, perform these steps:

Step 1 Install and activate the virtual service. For more information, see Configuring VirtualPortGroup Interfaces and Virtual Service, on page 75.

Step 2 Configure the blocked list parameter-map:

```
parameter-map type regex domainfilter_blacklist_pmap1
  pattern examplebook\.com
  pattern bitter\.com
```

Step 3 Configure the allowed list parameter-map:

```
parameter-map type regex domainfilter_whitelist_pmap1
  pattern example\.com
  pattern exmaplegogle\.com
```

Step 4 Configure the domain profile and associate the blocked list and allowed list parameter-maps:

```
utd web-filter domain profile 1
 blacklist
  parameter-map regex domainfilter_blacklist_pmap1
 whitelist
  parameter-map regex domainfilter_whitelist_pmap1
```

Step 5 (Optional) By default the domain filtering alerts are not enabled. Configure the alerts for the blocked list or allowed list, or both under the domain profile:

```
alert {all | blacklist | whitelist}
```

Step 6 Configure the external redirect-server under the domain profile:

```
redirect-server external x.x.x.x (This is the IP address that is used for serving block page when a
 page is on the blocked list)
```

Step 7 Configure the UTD engine standard with domain profile:

```
utd engine standard
 web-filter
  domain-profile 1
```

Step 8 Configure the UTD with engine standard and enable it globally or on a specific interface:

```
utd
  all-interfaces
  engine standard
```

This example shows how to configure domain-based web filtering with an external block server:

```
parameter-map type regex domainfilter_blacklist_pmap1
  pattern examplebook\.com
  pattern bitter\.com
parameter-map type regex domainfilter_whitelist_pmap1
  pattern exmaplegogle\.com
  pattern exmaplegogle\.com
utd engine standard
  web-filter
```

```
    domain-profile 1
!
utd web-filter domain profile 1
  alert all
  blacklist
    parameter-map regex domainfilter_blacklist_pmap1
  whitelist
    parameter-map regex domainfilter_whitelist_pmap1
  redirect-server external 192.168.1.1
!
utd
  all-interfaces
  engine standard
```

Configure Domain-based Web Filtering with a Local Block Server

To configure domain-based web filtering with a local block server, perform these steps:

Step 1 Install and activate the virtual service. For more information, see Configuring VirtualPortGroup Interfaces and Virtual Service, on page 75.

Step 2 Configure a loopback interface or use any existing interface that the client can access:

```
interface loopback 110
 ip address 10.1.1.1 255.255.255.255
exit
```

Step 3 Configure the UTD web filter with the local block server profile:

```
utd web-filter block local-server profile 1
 block-page-interface loopback 110
 http-ports 80
 content text "Blocked by Web-Filter"
```

Step 4 Configure the blocked list parameter-map:

```
parameter-map type regex domainfilter_blacklist_pmap1
  pattern bitter\.com
```

Step 5 Configure the allowed list parameter-map:

```
parameter-map type regex domainfilter_whitelist_pmap1
  pattern sweet\.com
```

Step 6 Configure the domain profile and associate the blocked list and allowed list parameter-maps:

```
utd web-filter domain profile1
 blacklist
  parameter-map regex domainfilter_blacklist_pmap1
 whitelist
  parameter-map regex domainfilter_whitelist_pmap1
```

Step 7 (Optional) By default the domain filtering alerts are not enabled. Configure the alerts for blocked list or allowed list, or both under the domain profile:

```
alert {all |blacklist | whitelist}
```

Step 8 Configure the redirect-server as local block server under the domain profile:

```
redirect-server local-block-server 1
```

Step 9 Configure the UTD engine standard with domain profile:

```
utd engine standard
 web-filter
  domain-profile 1
```

Step 10 Configure the UTD with engine standard and enable it globally or on a specific interface:

```
utd
  all-interfaces
  engine standard
```

This example shows how to configure a domain-based web filtering with a local block server:

```
interface loopback 110
  ip address 10.1.1.1 255.255.255.255
exit
parameter-map type regex domainfilter_blacklist_pmap1
  pattern bitter\.com
parameter-map type regex domainfilter_whitelist_pmap1
  pattern sweet\.com
utd engine standard
  web-filter
    domain-profile 1
!
utd web-filter block local-server profile 1
  block-page-interface Loopback110
  content text "Blocked by Web-Filter"
  http-ports 80
!
utd web-filter domain profile 1
  alert all
  blacklist
    parameter-map regex domainfilter_blacklist_pmap1
  whitelist
    parameter-map regex df_whitelist_pmap1
  redirect-server local-block-server 1
!
utd
  all-interfaces
  engine standard
```

Configure URL-based Web Filtering with a Local Block Server

To configure URL-based web filtering with a local block server, perform these steps:

Step 1 Install and activate the virtual service. For more information, see Configuring VirtualPortGroup Interfaces and Virtual Service, on page 75.

Step 2 Configure a loopback interface or use any existing interface that the client can access:

```
interface loopback 110
 ip address 10.1.1.1 255.255.255.255
exit
```

Step 3 Configure the UTD web filter with the local block server profile:

```
utd web-filter block local-server profile 1
 block-page-interface loopback 110
 http-ports 80
 content text "Blocked by Web-Filter"
```

Step 4 Configure the blocked list parameter-map:

```
parameter-map type regex urlf_blacklist_pmap1
 pattern exmplee.com/sports
```

Step 5 Configure the allowed list parameter-map:

```
parameter-map type regex urlf_whitelist_pmap1
 pattern examplehoo.com/finance
```

Step 6 Configure the URL profile and do the following:

```
utd web-filter url profile 1
```

a) Associate the blocked list and allowed list parameter-maps:

```
blacklist
  parameter-map regex urlf_blacklist_pmap1
whitelist
  parameter-map regex urlf_whitelist_pmap1
```

b) Configure the alerts for blocked list, allowed list or both under the local block-server profile:

```
alert {all | blacklist | whitelist}
```

c) Configure the categories to be allowed or blocked:

```
categories allow
  sports
```

d) Configure the reputation block threshold:

```
reputation
  block-threshold high-risk
```

e) Configure the URL source database with the fail option:

```
sourcedb fail close
```

f) Configure the log level. The default option is error. When you set the option to **info** or **detail**, the performance may impact:

```
log level error
```

g) Configure local block server:block

```
block local-server 1
```

Step 7 Configure the UTD engine standard with URL profile:

```
utd engine standard
 web-filter
   url-profile 1
```

Step 8 Configure the UTD engine standard and enable the UTD on a global or specific interface:

```
utd
  all-interfaces
  engine standard
```

This example shows how to configuration a URL-based web filtering with a local block server:

```
parameter-map type regex urlf_blacklist_pmap1
 pattern examplee.com/sports
parameter-map type regex urlf_whitelist_pmap1
 pattern exmaplehoo.com/finance
!
interface loopback 110
 ip address 10.1.1.1 255.255.255.255
 exit
utd web-filter block local-server profile 1
 block-page-interface loopback 110
 http-ports 80
 content text "Blocked by Web-Filter"
utd web-filter url profile 1
 blacklist
  parameter-map regex urlf_blacklist_pmap1
 whitelist
  parameter-map regex urlf_whitelist_pmap1
 alert all
 categories allow
  sports
 reputation
  block-threshold high-risk
 sourcedb fail close
 log level error
 block local-server 1
!
utd engine standard
 web-filter
  url-profile 1
!
utd
 all-interfaces
 engine standard
```

Configure URL-based Web Filtering with an Inline Block Page

To configure URL-based web filtering with an in-line block page, perform these steps:

Step 1 Install and activate the virtual service. For more information, see Configuring VirtualPortGroup Interfaces and Virtual Service, on page 75.

Step 2 Configure the blocked list parameter-map:

```
parameter-map type regex urlf_blacklist_pmap1
 pattern exmaplegogle.com/sports
```

Step 3 Configure the allowed list parameter-map:

```
parameter-map type regex urlf_whitelist_pmap1
 pattern exmaplehoo.com/finance
```

Step 4 Configure the UTD block page profile:

```
utd web-filter block page profile 1
 text "Blocked by Web-Filter URLF" (The other options are file and redirect-url)
```

Step 5 Configure the URL profile and do the following:

```
utd web-filter url profile 1
```

a) Associate the blocked list and allowed list parameter-maps:

```
blacklist
  parameter-map regex urlf_blacklist_pmap1
whitelist
  parameter-map regex urlf_whitelist_pmap1
```

b) Configure the alerts for blocked list, allowed list or both under the local block-server profile:

```
alert {all | blacklist | whitelist | categories-reputation}
```

c) Configure the categories to be allowed or blocked:

```
categories allow
  sports
```

d) Configure the reputation block threshold:

```
reputation
  block-threshold high-risk
```

e) Configure the URL source database with the fail option:

```
sourcedb fail close
```

f) Configure the log level. The default option is error. When you set the option to **info** or **detail**, the performance may impact:

```
log level error
```

g) Configure local block server:block

```
block local-server 1
```

Step 6 Configure the UTD engine standard with URL profile:

```
utd engine standard
 web-filter
   url-profile 1
```

Step 7 Configure the UTD engine standard and enable the UTD on a global or specific interface:

```
utd
  all-interfaces
  engine standard
```

This example shows how to configuration an URL-based web filtering with an inline block server:

```
parameter-map type regex urlf_blacklist_pmap1
  pattern exmaplegogle.com/sports
parameter-map type regex urlf_whitelist_pmap1
 pattern exmaplehoo.com/finance
!
utd web-filter block page profile 1
 text "Blocked by Web-Filter URLF"
!
utd web-filter url profile 1
 blacklist
```

```
  parameter-map regex urlf_blacklist_pmap1
 whitelist
  parameter-map regex urlf_whitelist_pmap1
 alert all
 categories allow
  sports
 reputation
  block-threshold high-risk
 sourcedb fail close
 log level error
!
utd engine standard
 web-filter
  url-profile 1
!
utd
 all-interfaces
 engine standard
```

Configuring Domain/URL based Web Filtering and Snort IPS

To configure Domain/URL based web filtering and Snort IPS, perform these steps:

Step 1 Configure the domain profile:

```
utd web-filter domain profile 1
```

Step 2 Configure the URL profile:

```
utd web-filter url profile 1
```

Step 3 Configure the threat-inspection under UTD engine standard:

```
utd engine standard
 threat-inspection
```

Step 4 Configure the web-filter under UTD engine standard with the domain and URL profiles:

```
utd engine standard
 logging syslog
 threat-inspection
  threat protection
  policy security
 signature update server cisco username xxx password QhLb]Z[ifMbFgLYgR]^KLDUZ
  signature update occur-at daily 0 0
  logging level error
 web-filter
  domain-profile 1
  url-profile 1
```

Step 5 Configure the UTD engine standard and enable it globally or on a specific interface:

```
utd
  all-interfaces
  engine standard
```

Verifying the Web Filter Configuration

You can verify the Web Filtering configuration using the following commands:

```
Device# show utd engine standard config

UTD Engine Standard Configuration:
  Operation Mode : Intrusion Detection
  Policy         : Balanced

  Signature Update: Not Configured

  Logging:
    Server    :  IOS Syslog
    Level     : err (Default)
    Statistics    : Disabled

  Whitelist : Disabled
  Whitelist Signature IDs:

Web-Filter      : Enabled

  Whitelist :
    www.cisco.com
  Blacklist :
    www.hotstar.com

  Categories Action : Block
  Categories :
    Fashion and Beauty

  Block Profile:
   No config present

  Reputation Block Threshold : Moderate risk
  Alerts Enabled : Blacklist
  Cloud Lookup : Enabled
  Debug level : Error
Conditional debug level : Error
```

Troubleshooting Web Filtering

To collect the logs, use the **virtual-service move name "CONTAINER_NAME" log to bootflash:** command. You can troubleshoot issues that are related to enabling Web Filtering feature using the following commands on the device:

- **debug utd engine standard all**
- **debug utd engine standard climgr**
- **debug utd engine standard daq**
- **debug utd engine standard internal**
- **debug utd engine standard onep**

For release 16.8.1, configuration error recovery on container is enhanced in order to apply configuration and signature updates to the container. With the improved error recovery, you can have:

- Greater robustness during configuration download to detect and act upon errors.
- Efficient way of handling signature and configuration updates occuring together.
- Early detect and recover from the loss of the oneP connection between IOSd and CLIMGR. For example, when CLIMGR crashes.
- Improved visibility to the detailed results of the (current or recent) configuration download, without requiring you to enable debugs.

Configuration Examples

The following example shows how to enable domain filtering on CSR 1000V Cloud Services Router:

```
Device# configure terminal
Device(config)# parameter-map type regex wlist1
Device(config-profile)# pattern google.com
Device(config-profile)# pattern cisco.com
Device(config-profile)# exit
Device(config)# parameter-map type regex blist1
Device(config-profile)# pattern exmaplehoo.com
Device(config-profile)# pattern bing.com
Device(config-profile)# exit
Device(config)# utd web-filter block local-server profile 1
Device(config--utd-webf-blk-srvr)# content file bootflash:test.utd.file
Device(config--utd-webf-blk-srvr)# end
```

For the local block server to work, HTTP server should be running. Use the ip http server command to configure the block server. The show ip http server status command displays the server status as enabled.

```
Device# show ip http server status
HTTP server status: Enabled
HTTP server port: 80
```

Example: Configuring Web Filter Domain Profile

The following example shows how to configure web filter domain profile:

```
Device(config)# utd web-filter domain profile 1
Device(config-utd-webfltr-domain)# blacklist
Device(config-utd-webf-dmn-bl)# parameter-map regex blist1
Device(config-utd-webf-dmn-bl)# whitelist
Device(config-utd-webf-dmn-wl)# parameter-map regex wlist1
Device(config-utd-webf-dmn-wl)# exit
Device(config-utd-webfltr-domain)# alert all
Device(config-utd-webfltr-domain)# redirect-server external 1.2.3.4
Device(config-utd-webfltr-domain)# exit
```

Configuring Web Filter URL Profile

The following example shows how to configure web filter URL profile:

```
Device(config)# utd web-filter url profile 1
Device(config-utd-webfltr-url)# blacklist
Device(config-utd-webf-url-bl)# parameter-map regex blist1
Device(config-utd-webf-url-bl)# whitelist
Device(config-utd-webf-url-wl)# parameter-map regex wlist1
Device(config-utd-webf-url-wl)# exit
Device(config-utd-webfltr-url)# categories allow
Device(config-utd-webf-url-cat)# news-and-media
Device(config-utd-webf-url-cat)# search-engines
Device(config-utd-webf-url-cat)# computer-and-internet-info
Device(config-utd-webf-url-cat)# computer-and-internet-security
Device(config-utd-webf-url-cat)# financial-services
Device(config-utd-webf-url-cat)# image-and-video-search
Device(config-utd-webf-url-cat)# job-search
Device(config-utd-webf-url-cat)#exit
Device(config-utd-webfltr-url)# alert all
Device(config-utd-webfltr-url)# reputation
Device(config-utd-webf-url-rep)# block-threshold suspicious
Device(config-utd-webf-url-rep)# exit
Device(config-utd-webfltr-url)# block local-server 1
Device(config-utd-webfltr-url)# exit
```

Configuring UTD Snort IPS/IDS Allowed List Signatures

The following example shows how to configure signature allowed lists:

```
Device(config)# utd threat-inspection whitelist
Device(config-utd-whitelist)# generator id 1 signature id 1
Device(config-utd-whitelist)# generator id 1 signature id 2
Device(config-utd-whitelist)# exit
```

Example: Configuring Web Filter Profile

The following example shows how to configure web filter profile:

```
Device(config)# utd engine standard
Device(config-utd-eng-std)# logging server 1.2.3.4
Device(config-utd-eng-std)# threat-inspection
Device(config-utd-engstd-insp)#threat protection
Device(config-utd-engstd-insp)# policy security
Device(config-utd-engstd-insp)# logging level emerg
Device(config-utd-engstd-insp)# whitelist
Device(config-utd-engstd-insp)# web-filter
Device(config-utd-engstd-webf)# domain-profile 1
Device(config-utd-engstd-webf)# url-profile 1
Device(config-utd-engstd-webf)# exit
```

Example: Alert Messages for Web Filtering Events

The following example shows alert messages for web filtering events:

```
016/06/02-14:44:41.061501 IST [**] [Instance_ID: 1] [**] Drop [**] UTD WebFilter Blacklist
 [**] [URL: www.edition.cnn.com/2016/03/31/asia/kolkata-bridge-collapse/index.html]
[Initiator_VRF: 0] {TCP} 1.0.0.9:56608 -> 2.0.0.29:80

2016/06/02-14:48:06.636270 IST [**] [Instance_ID: 1] [**] Pass [**] UTD WebFilter Whitelist
 [**] [URL: www.ndtv.com/index.html] [Initiator_VRF: 0] {TCP} 1.0.0.9:56611 -> 2.0.0.23:80

Jun 2 14:37:57.856 IST: %IOSXE-6-PLATFORM: F0: cpp_cp: QFP:0.0 Thread:000
TS:00000618422205723793 %UTD-6-UTD_DF_BLACKLIST_MATCH: UTD WebFilter Domain Blacklist [**]
 [Domain: www.cricinfo.com] [Matched Pattern: www.cricinfo.com] {UDP} 2.0.0.10:53 ->
1.0.0.9:55184
```

```
Jun 2 14:39:22.653 IST: %IOSXE-6-PLATFORM: F0: cpp_cp: QFP:0.0 Thread:000
TS:00000618507002407540 %UTD-6-UTD_DF_WHITELIST_MATCH: UTD WebFilter Domain Whitelist [**]
 [Domain: www.cricinfo.com] [Matched Pattern: www.cricinfo.com] {UDP} 2.0.0.10:53 ->
1.0.0.9:55286
```

Example: Unconfigure Cloud-Lookup

The following example shows how to unconfigure Cloud-Lookup feature in Web Filtering:

```
Device(config)# utd engine standard
Device(config-utd-eng-std)# web-filter
% Please ensure urlf-<low/medium/high> virtual-service profile is configured to use the
web-filter feature

Device(config-utd-engstd-webf)# no cloud-lookup
Device(config-utd-engstd-webf)# end
Device # exit
```

Additional References for Cisco Web Filtering

Related Documents

Related Topic	Document Title
IOS commands	Cisco IOS Master Command List, All Releases
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
UCS E-Series Servers	http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/e/2-0/gs/guide/b_2_0_Getting_Start

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 Web Filtering

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 4: Feature Information for Cisco Web Filtering

Feature Name	Releases	Feature Information
Cisco Web Filtering	Cisco IOS XE Denali Release 16.3.1	The Web Filtering feature enables the user to provide controlled access to Internet websites by configuring the domain-based or URL-based policies and filters on the device. The user can configure the web filtering profiles to manage the web access.Web Filtering feature is implemented using the container service and it is similar to the Snort IPS solution.
UTD feature parity on ISRv UTD Serviceability Enhancements	Cisco IOS XE Fuji Release 16.8.1	Domain and URL filtering in both single-tenant and multi-tenant mode are supported for CSR. For ISRv, only single-tenant is supported. This feature is available on all models of the ENCS platforms. Error recovery feature in UTD is enhanced to allow the container to recover from internal error by initiating a bulk configuration download from IOS. The command **utd web-filter** *profile name* is modifed.
Web Root URL Filtering Enhancements	Cisco IOS XE Fuji Release 16.9.1	The URLF Virtual Resource Profiles in Web Filtering are supported only on platforms CSR1000v and ISRv. The URL Filtering supports cloud-lookup feature to search for the URLs in cloud that are not present in the database.

CHAPTER 4

Configuring Multi-Tenancy for Unified Threat Defense

Multi-tenancy for Unified Threat Defense provides Snort IPS and Web Filtering for multiple users. You can define policies for one or more tenants in a single Cisco CSR 1000v instance. Each policy can have a threat inspection profile and a web filtering profile. The following sections describe how to configure multi-tenancy for Unified Threat Defense. Many of the commands used in these configuration steps are similar to those used in configuring single-tenancy—see: Snort IPS, on page 21 and Web Filtering , on page 69.

Information About Multi-Tenancy for Unified Threat Defense

Multi-tenancy for Snort IPS and Web Filtering allows you to define policies for one or more tenants, in one Cisco CSR 1000v instance. This feature was introduced in Cisco IOS XE Everest 16.6.1.

Each tenant is a VPN routing and forwarding instance with one or more VPN routing and forwarding tables (VRFs). A Unified Threat Defense (UTD) policy is associated with a threat inspection profile and web filtering profile. Multiple tenants can share a UTD policy.

The system logs include the name of the VRF which allows you to produce statistics per-tenant.

The CLI commands used in multi-tenancy mode are similar to those used in single-tenancy mode (see Snort IPS, on page 21 and Web Filtering , on page 69). In multi-tenancy, you enter a sub-mode `utd engine standard multi-tenancy` and configure UTD policies, web filtering and threat-inspection profiles. After exiting the `utd engine standard multi-tenancy` sub-mode, the UTD policies are applied.

The benefits of web filtering and threat inspection (Snort IPS/IDS) are explained in the following sections:

- Benefits of Web Filtering

- Overview of Snort Virtual Service Interfaces

Web Filtering Overview

Web Filtering allows you to provide controlled access to the internet by configuring URL-based policies and filters. Web Filtering helps to control access to websites by blocking malicious or unwanted websites and therefore making the network more secure. You can blocked list individual URLs or domain names and configure allowed list policies for the same. You can also make provision to allow or block a URL based on reputation or category.

Snort IPS Overview

The Snort IPS feature enables Intrusion Prevention System (IPS) or Intrusion Detection System (IDS) for branch offices on Cisco 4000 Series Integrated Services Routers and Cisco Cloud Services Router 1000v Series. This feature uses the Snort engine to provide IPS and IDS functionalities.

Snort is an open source network IPS that performs real-time traffic analysis and generates alerts when threats are detected on IP networks. It can also perform protocol analysis, content searching or matching, and detect a variety of attacks and probes, such as buffer overflows, stealth port scans, and so on. The Snort engine runs as a virtual container service on Cisco 4000 Series Integrated Services Routers and Cisco Cloud Services Router 1000v Series.

The Snort IPS feature works in the network intrusion detection and prevention mode that provides IPS or IDS functionalities. In the network intrusion detection and prevention mode, Snort performs the following actions:

- Monitors network traffic and analyzes against a defined rule set.
- Performs attack classification.
- Invokes actions against matched rules.

Based on your requirements, you can enable Snort either in IPS or IDS mode. In IDS mode, Snort inspects the traffic and reports alerts, but does not take any action to prevent attacks. In IPS mode, in addition to intrusion detection, actions are taken to prevent attacks.

The Snort IPS monitors the traffic and reports events to an external log server or the IOS syslog. Enabling logging to the IOS syslog may impact performance due to the potential volume of log messages. External third-party monitoring tools, which supports Snort logs, can be used for log collection and analysis.

Snort IPS Solution

The Snort IPS solution consists of the following entities:

- Snort sensor—Monitors the traffic to detect anomalies based on the configured security policies (that includes signatures, statistics, protocol analysis, and so on) and sends alert messages to the Alert/Reporting server. The Snort sensor is deployed as a virtual container service on the router.
- Signature store—Hosts the Cisco Signature packages that are updated periodically. These signature packages are downloaded to Snort sensors either periodically or on demand. Validated signature packages are posted to Cisco.com. Based on the configuration, signature packages can be downloaded from Cisco.com or a local server.

  The following domains are accessed by the router in the process of downloading the signature package from cisco.com:

  - api.cisco.com

- apx.cisco.com
- cloudsso.cisco.com
- cloudsso-test.cisco.com
- cloudsso-test3.cisco.com
- cloudsso-test4.cisco.com
- cloudsso-test5.cisco.com
- cloudsso-test6.cisco.com
- cloudsso.cisco.com
- download-ssc.cisco.com
- dl.cisco.com
- resolver1.opendns.com
- resolver2.opendns.com

Note If you are downloading signature packages from a local server to hold the signature packages, only HTTP is supported.

Signature packages must be manually downloaded from Cisco.com to the local server by using Cisco.com credentials before the Snort sensor can retrieve them.

The Snort container performs a domain-name lookup (on the DNS server(s) configured on the router) to resolve the location for automatic signature updates from Cisco.com or on the local server, if the URL is not specified as the IP address.

- Alert/Reporting server—Receives alert events from the Snort sensor. Alert events generated by the Snort sensor can either be sent to the IOS syslog or an external syslog server or to both IOS syslog and external syslog server. No external log servers are bundled with the Snort IPS solution.
- Management—Manages the Snort IPS solution. Management is configured using the IOS CLI. Snort Sensor cannot be accessed directly, and all configuration can only be done using the IOS CLI.

Overview of Snort Virtual Service Interfaces

The Snort sensor runs as a service on routers. Service containers use virtualization technology to provide a hosting environment on Cisco devices for applications.

You can enable Snort traffic inspection either on a per interface basis or globally on all supported interfaces. The traffic to be inspected is diverted to the Snort sensor and injected back. In Intrusion Detection System (IDS), identified threats are reported as log events and allowed. However, in Intrusion Prevention System (IPS), action is taken to prevent attacks along with log events.

The Snort sensor requires two VirtualPortGroup interfaces. The first VirtualPortGroup interface is used for management traffic and the second for data traffic between the forwarding plane and the Snort virtual container service. Guest IP addresses must be configured for these VirtualPortGroup interfaces. The IP subnet assigned to the management VirtualPortGroup interface should be able to communicate with the Signature server and Alert/Reporting server.

The IP subnet of the second VirtualPortGroup interface must not be routable on the customer network because the traffic on this interface is internal to the router. Exposing the internal subnet to the outside world is a security risk. We recommend the use of 192.0.2.0/30 IP address range for the second VirtualPortGroup subnet. The use of 192.0.2.0/24 subnet is defined in RFC 3330.

You can assign the Snort virtual container service IP address on the same management network as the router on which the virtual service is running. This configuration helps if the syslog or update server is on the management network and is not accessible by any other interfaces

Restrictions for Configuring Multi-Tenancy for Unified Threat Defense

- Multi-tenancy for Unified Threat Defense is only supported on the Cisco CSR 1000v.
- Domain-based filtering is not supported.
- Up to 25 tenants are supported on each Cisco CSR 1000v instance.
- A maximum of 25 policies are supported.
- A maximum of 50,000 concurrent sessions are supported on a Cisco CSR 1000v.
- Bringing up (or reloading/updating) the Snort IPS/IDS package may take up to 20 minutes, depending on the number of policies configured with threat inspection. Updating the signatures will reload Snort IPS and will also take up to 20 minutes.
- The blocked list/allowed list rules support only a regular expression (regex) pattern. Currently, 64 patterns are supported for each blocked list/allowed list rule. However, each tenant can have multiple rules.
- Local block server does not support serving HTTPS block page.When the URL filter tries to inject block page or redirect message, it does not support HTTPS traffic.
- When there is a username and password in the URL, URL filter does not remove them from the URL before matching the blocked list/allowed list pattern. However, the category/reputation lookup does not have this limitation and removes the username and password from the URL before lookup.
- HTTPS inspection is limited. Web filtering uses server certificate to obtain the URL/domain information. It is not possible to inspect the full URL path.
- UTD does not inter-operate with WCCP, and NBAR under inter-VRF scenario.
- The Snort IPS command `threat inspection profile` *`profile-name`* uses an alphanumeric profile-name, not an ID (number).

Prerequisites for Configuring Multi-Tenancy for Unified Threat Defense

Before you configure the multi-tenancy for UTD feature on the Cisco CSR 1000v, ensure that the router is set up as follows:

- The Cisco CSR 1000v running Cisco IOS XE Everest 16.6.1 or later.
- The Cisco CSR 1000v must have a security K9 license to enable web filtering.
- The Cisco CSR 1000v "multi-tenancy" profile requires the following virtual service System CPU, virtual service Memory, and Platform Requirements:

  System CPU—25%

  Platform Memory Requirements—Min. 12GB RAM (8GB disk/flash)

How to Configure Multi-Tenancy for Unified Threat Defense

To deploy multi-tenancy for Unified Threat Defense on supported devices, perform the following tasks:

Before you begin

Provision the device upon which you wish to install web filtering and threat inspection for multi-tenancy. This feature is currently only supported on the Cisco CSR 1000v.

Obtain the license. UTD is available only for routers running security packages and you will require a security license to enable the service. Contact Cisco Support to obtain a security license.

SUMMARY STEPS

1. Install and activate the virtual-service: Installing the UTD OVA File for Multi-Tenancy, on page 94.
2. Configure the VirtualPortGroup interfaces and the virtual-service: How to Configure VirtualPortGroup Interfaces and Virtual Service for Multi-Tenancy, on page 94.
3. Configure the VRFs: How to Configure VRFs for Multi-Tenancy, on page 97.
4. Configure threat inspection and web filtering for multi-tenancy: How to Configure Multi-Tenancy Web Filtering and Threat Inspection, on page 98

DETAILED STEPS

Step 1 Install and activate the virtual-service: Installing the UTD OVA File for Multi-Tenancy, on page 94.

Step 2 Configure the VirtualPortGroup interfaces and the virtual-service: How to Configure VirtualPortGroup Interfaces and Virtual Service for Multi-Tenancy, on page 94.

Step 3 Configure the VRFs: How to Configure VRFs for Multi-Tenancy, on page 97.

Step 4 Configure threat inspection and web filtering for multi-tenancy: How to Configure Multi-Tenancy Web Filtering and Threat Inspection, on page 98

Installing the UTD OVA File for Multi-Tenancy

The virtual-service OVA file is an Open Virtualization Archive file that contains a compressed, installable version of a virtual machine. You must download this OVA file to the router and then install the virtual-service. The virtual-service OVA file is not bundled with Cisco IOS XE release images that are installed on the router. OVA files may be available pre-installed in the router's flash memory.

For installing the OVA file, you must use a Cisco IOS XE image with a security license. During installation, the security license is checked.

Example of installing the virtual service:

```
Device> enable
Device# virtual-service install name utd package
bootflash:utdsnort.1.0.4_SV2983_XE_16_6.20170623_174453_RELEASE.ova
Device# show virtual-service list

Name Status   Package Name
------------------------------------------------------------------------------
utd Activated utdsnort.1.0.4_SV2983_XE_16_6.20170
```

Example of upgrading the virtual service:

```
Device> enable
Device# virtual-service upgrade name utd package
bootflash:utdsnort.1.0.4_SV2983_XE_16_6.20170623_174453_RELEASE.ova
Device# show virtual-service list

Name Status   Package Name
------------------------------------------------------------------------------
utd Activated utdsnort.1.0.4_SV2983_XE_16_6.20170
```

Example of uninstalling the virtual service:

```
Device> enable
Device# virtual-service uninstall name utd
Device# show virtual-service list

Virtual Service List:
```

How to Configure VirtualPortGroup Interfaces and Virtual Service for Multi-Tenancy

As shown in this procedure, for multi-tenancy you must configure two VirtualPortGroup interfaces and guest IP addresses for both interfaces.

Note The VirtualPortGroup interface for data traffic must use a private or nonroutable IP address. We recommend the use of 192.0.2.0/30 IP address range for this interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface VirtualPortGroup** *interface-number*

4. **ip address** *ip-address mask*
5. **exit**
6. **interface VirtualPortGroup** *interface-number*
7. **ip address** *ip-address mask*
8. **exit**
9. **virtual-service** *name*
10. **profile multi-tenancy**
11. **vnic gateway VirtualPortGroup** *interface-number*
12. **guest ip address** *ip-address*
13. **exit**
14. **vnic gateway VirtualPortGroup** *interface-number*
15. **guest ip address** *ip-address*
16. **exit**
17. **activate**
18. **end**
19. **show virtual-service list**

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	**interface VirtualPortGroup** *interface-number* **Example:** `Device(config)# interface VirtualPortGroup 0`	Enters interface configuration mode and configures a VirtualPortGroup interface. This interface is used for management traffic when the management interface GigabitEthernet0 is not used.
Step 4	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 10.1.1.1 255.255.255.252`	Sets a primary IP address for an interface. This interface needs to be routable to the signature update server and external log server.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 6	**interface VirtualPortGroup** *interface-number* **Example:** `Device(config)# interface VirtualPortGroup 1`	Configures an interface and enters interface configuration mode. Configure a VirtualPortGroup interface. This interface is used for data traffic.

	Command or Action	Purpose
Step 7	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.0.2.1 255.255.255.252`	Sets a primary IP address for an interface. This IP address should not be routable to the outside network. The IP address is assigned from the recommended 192.0.2.0/30 subnet.
Step 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 9	**virtual-service** *name* **Example:** `Device(config)# virtual-service utd`	Configures a virtual container service and enters virtual service configuration mode. The *name* argument is the logical name that is used to identify the virtual container service.
Step 10	**profile multi-tenancy** **Example:** `Device(config-virt-serv)#profile multi-tenancy`	Configures a resource profile. For multi-tenancy mode (Cisco CSR 1000v only), this `profile multi-tenancy` command must be configured.
Step 11	**vnic gateway VirtualPortGroup** *interface-number* **Example:** `Device(config-virt-serv)# vnic gateway VirtualPortGroup 0`	Enters the virtual-service virtual network interface card (vNIC) configuration mode. Creates a vNIC gateway interface for the virtual container service and maps the vNIC gateway interface to the virtual port group interface. This is the interface that was configured in Step 3.
Step 12	**guest ip address** *ip-address* **Example:** `Device(config-virt-serv-vnic)# guest ip address 10.1.1.2`	Configures a guest vNIC address for the vNIC gateway interface.
Step 13	**exit** **Example:** `Device(config-virt-serv-vnic)# exit`	Exits virtual-service vNIC configuration mode and returns to virtual service configuration mode.
Step 14	**vnic gateway VirtualPortGroup** *interface-number* **Example:** `Device(config-virt-serv)# vnic gateway VirtualPortGroup 1`	Enters virtual-service vNIC configuration mode. Configures a vNIC gateway interface for the virtual container service and maps the interface to the virtual port group. The interface (*interface-number*) configured in Step 6) is used by the Snort engine for monitoring user traffic.
Step 15	**guest ip address** *ip-address* **Example:** `Device(config-virt-serv-vnic)# guest ip address 192.0.2.2`	Configures a guest vNIC address for the vNIC gateway interface.
Step 16	**exit** **Example:** `Device(config-virt-serv-vnic)# exit`	Exits virtual-service vNIC configuration mode and returns to virtual service configuration mode.

	Command or Action	Purpose
Step 17	**activate** **Example:** `Device(config-virt-serv)# activate`	Activates an application installed in a virtual container service.

Step 18	**end**
	Example:
	`Device(config-virt-serv)# end`
		Exits virtual service configuration mode and returns to privileged EXEC mode.

Step 19	**show virtual-service list**
	Example:
	`Device# show virtual-service list`
	`Virtual Service List:`
	`Name  Status      Package Name`
	`------------------------------------------------`
	`utd   Activated`
	`utdsnort.1.0.4_SV2983_XE_16_6.20170`	

How to Configure VRFs for Multi-Tenancy

This procedure describes the typical steps required for configuring VRFs for the tenants, which are later used in: How to Configure Multi-Tenancy Web Filtering and Threat Inspection, on page 98.

Note For inter-VRF traffic, if the traffic flowing between two VRFs has ingress and egress interfaces configured for UTD, rules are applied to decide which VRF represents the session. The UTD policy for the selected VRF then applies to all packets in the inter-VRF traffic.

SUMMARY STEPS

1. **vrf definition** *vrf-name*
2. **rd** *route-distinguisher*
3. **address-family ipv4**
4. **exit address-family**
5. Repeat steps 1 to 4 for each VRF.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**vrf definition** *vrf-name* **Example:** `Device(config)# vrf definition 100`	Defines the name of the VRF and enters VRF configuration mode.

	Command or Action	Purpose
Step 2	**rd** *route-distinguisher* **Example:** `Device(config-vrf)# rd 100:1`	Creates the routing and forwarding tables and associates the *route-distinguisher* with the VRF instance named *vrf-name*. The router uses the route-distinguisher to identify the VRF to which a packet belongs. The route-distinguisher is of one of the following two types: • Autonomous System-related. An AS number xxx and an arbitrary number y—xxx:y • IP address-related. An IP address A.B.C.D and an arbitrary number y—A.B.C.D:y
Step 3	**address-family ipv4** **Example:** `Device(config-vrf)# address-family ipv4`	Enters address family configuration mode for configuring routing sessions using the IP Version 4 address.
Step 4	**exit address-family** **Example:** `Device(config-vrf-af)# exit`	Exits address family configuration mode.
Step 5	Repeat steps 1 to 4 for each VRF.	

How to Configure Multi-Tenancy Web Filtering and Threat Inspection

To configure threat inspection (IPS/IDS) and web filtering for multi-tenancy (multiple tenants/VRFs), perform the following steps.

In this procedure, the definition of blocked list and allowed lists are shown in the initial steps 1 to 5. The main configuration steps (in UTD standard engine configuration mode for multi-tenancy) are shown in step 6 onwards.

Note For details about threat inspection and web filtering for single-tenancy, see Snort IPS, on page 21 and Web Filtering , on page 69.

Before you begin

Remove any existing single-tenancy UTD configuration, using the `no utd engine standard` command.

You must have previously configured a VRF for each tenant—see How to Configure VRFs for Multi-Tenancy, on page 97.

Procedure

	Command or Action	Purpose
Step 1	**parameter-map type regex** *blacklist-name* **Example:**	Defines a blocked list parameter map, which is used later in step 17.

	Command or Action	Purpose

	`Device(config)# parameter-map type regex`
	`urlf-blacklist1`	

Step 2	**pattern** *URL-name*
	Example:
	`Device(config-profile)# pattern www\.cnn\.com`
	`Device(config-profile)# pattern www\.msnbc\.com`
		Defines the URL to be on the blocked list. Note that the periods within *URL-name* must be preceded by an escape "\" character. Repeat this step to configure multiple URLs to be on the blocked list.

Step 3	**parameter-map type regex** *whitelist-name*
	Example:
	`Device(config-profile)# parameter-map type regex`
	` urlf-whitelist1`
		Defines an allowed list parameter map, which is used later in step 20.

Step 4	**pattern** *URL-name* **Example:** `Device(config-profile)# pattern www\.nfl\.com`	Defines the URL(s) to be on the allowed list. Note that, for URLs on the blocked list, periods within *URL-name* must be preceded by an escape "\" character. Repeat this step to configure multiple URLs to be on the allowed list.
Step 5	**exit** **Example:** `Device(config-profile)# exit`	
Step 6	**utd multi-tenancy** **Example:** `Device(config)# utd multi-tenancy`	This command acts a switch, in preparation for the following `utd engine standard multi-tenancy` command.
Step 7	**utd engine standard multi-tenancy** **Example:** `Device(config)# utd engine standard multi-tenancy`	Enters UTD standard engine configuration mode for multi-tenancy. **Note** Later. after you exit the UTD standard engine configuration mode in step 50, the policy configurations are applied.
Step 8	**web-filter sourcedb** *sourcedb-number* **Example:** `Device(config)# web-filter sourcedb 1`	Configures a web filtering sourcedb profile—*sourcedb-number*, which is numeric. This is used later in step 29.

Step 9	**logging level** {**alerts** | **critical** | **debugging** | **emergencies** | **errors** | **informational** | **notifications** | **warnings**}
	Example:
	`Device(config)# logging level errors`
		Sets the level of system messages that are reported upon for web filtering events. Messages of the specified level and lower are reported. (Each level has a numeric value as shown in the table below.)

Table 5: System Message Severity Levels

Level	Description
0 – emergencies	System unusable

<table>
<tr><th></th><th>Command or Action</th><th>Purpose</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Level</th><th>Description</th></tr>
<tr><td>1 – alerts</td><td>Immediate action needed</td></tr>
<tr><td>2 – critical</td><td>Critical condition</td></tr>
<tr><td>3 – errors</td><td>Error condition</td></tr>
<tr><td>4 – warnings</td><td>Warning condition</td></tr>
<tr><td>5 – notifications</td><td>Normal but significant condition</td></tr>
<tr><td>6 – informational</td><td>Informational messages only</td></tr>
<tr><td>7 – debugging</td><td>Appears during debugging only</td></tr>
</table>
</td></tr>
<tr><td><b>Step 10</b></td><td><b>web-filter block local-server profile</b> <i>profile-id</i>
<b>Example:</b>
<code>Device(config-utd-multi-tenancy)# web-filter block local-server profile 1</code>
The content text is displayed by the local server.</td><td>Configures the a local block server profile for web filtering. The range of values for <i>profile-id</i> is 1–255.
See Configure URL-based Web Filtering with a Local Block Server.
<b>Note</b> When configuring commands for multi-tenancy, compared to single-tenancy, you do not use the initial <code>utd</code> keyword.</td></tr>
<tr><td><b>Step 11</b></td><td><b>block-page-interface loopback</b> <i>id</i>
<b>Example:</b>
<code>Device(config-utd-mt-webf-blk-srvr)# block-page-interface loopback 110</code></td><td>Associates a loopback interface with this profile. The IP address of this loopback interface is then used as the IP address of the block local-server.</td></tr>
<tr><td><b>Step 12</b></td><td><b>content text</b> <i>display-text</i>
<b>Example:</b>
<code>Device(config-utd-mt-webf-blk-srvr)# content text "Blocked by Web-Filter"</code></td><td>Specifies the warning text that appears after a blocked page is accessed.</td></tr>
<tr><td><b>Step 13</b></td><td><b>http-ports</b> <i>port-number</i>
<b>Example:</b>
<code>Device(config-utd-mt-webf-blk-srvr)# http-ports 80</code></td><td>The http-ports value is a string of ports separated by commas. The nginx HTTP server listens to these ports.</td></tr>
<tr><td><b>Step 14</b></td><td><b>web-filter block page profile</b> <i>profile-name</i>
<b>Example:</b>
<code>Device(config-utd-multi-tenancy)# web-filter block page profile 1</code></td><td>See Configure URL-based Web Filtering with an Inline Block Page, on page 80, except that the command used here for multi-tenancy does not use the <code>utd</code> keyword which is used for single-tenancy.).</td></tr>
</table>

	Command or Action	Purpose
	`Device(config-utd-mt-webf-block-urc)# text "this page is blocked"`	

Step 15	**web-filter url profile** *web-filter-profile-id*
	Example:
	`Device(config-utd-multi-tenancy)# web-filter url profile 1`
	`Device(config-utd-mt-webfltr-url)#`
		Specifies a URL profile for web filtering—*web-filter-profile-id*. Values: 1–255. After this command, you can configure alerts for blocked lists, allowed lists, and categories. For further information, see: Configure URL-based Web Filtering with an Inline Block Page.
		Note When configuring commands for multi-tenancy, compared to single-tenancy, you do not use an initial `utd` keyword.

Step 16	**blacklist** **Example:** `Device(config-utd-mt-webfltr-url)# blacklist`	Enters web filtering blocked list configuration mode.
Step 17	**parameter-map regex** *blacklist-name* **Example:** `Device(config-utd-mt-webf-url-bl)# parameter-map regex urlf-blacklist1`	Specifies a parameter-map regular expression using the blocked list that was defined earlier in step 1.

Step 18	**exit**
	Example:
	`Device(config-utd-mt-webf-url-bl)# exit`
	`Device(config-utd-mt-webfltr-url)#`
		Exits web filtering blocked list configuration mode.

Step 19	**whitelist**
	Example:
	`Device(config-utd-mt-webfltr-url)# whitelist`
	`Device(config-utd-mt-webf-url-wl)#`
		Enters web filtering allowed list configuration mode.

Step 20	**parameter-map regex** *whitelist-name* **Example:** `Device(config-utd-mt-webf-url-wl)# parameter-map regex urlf-list1`	Specifies a parameter-map regular expression using the allowed list that was defined earlier in step 3.

Step 21	**exit**
	Example:
	`Device(config-utd-mt-webf-url-wl)# exit`
	`Device(config-utd-mt-webfltr-url)#`
		Exits web filtering allowed list configuration mode.

Step 22	**exit**
	Example:
	`Device(config-utd-mt-webfltr-url)# exit`
	`Device(config-utd-multi-tenancy)#`
		Exits web filtering URL profile mode.

	Command or Action	Purpose
Step 23	**utd global** **Example:** `Device(config-utd-multi-tenancy)# utd global`	The commands entered for `utd global` apply to all tenants or policies e.g the commands shown below: `logging host syslog` and `threat inspection` for this Cisco CSR 1000v instance.

Step 24	**logging** {**host** *hostname* | **syslog**}
	Example:
	In this example, alerts are logged to a designated host log file.
	`Device(config-utd-mt-utd-global)# logging host systemlog1`
	Example:
	In this example, alerts are logged to IOS syslogs.
	`Device(config-utd-mt-utd-global)# logging syslog`
		The `logging` command specifies either a host name or IOS syslog, to which syslog messages are sent.

Step 25	**threat inspection** **Example:** `Device(config-utd-mt-utd-global)# threat inspection`	Enters global threat inspection mode.
Step 26	**signature update server** {**cisco** | **url** *url* } [**username** *username* [**password** *password*]] **Example:** `Device(config-utd-mt-utd-global-threat)# signature update server cisco username abcd password cisco123`	Configures the signature update server parameters. You must specify the signature update parameters with the server details. If you use www.cisco.com for signature updates, you must provide the username and password. If you use a local server for signature updates, based on the server settings you can provide the username and password. The router must be able to resolve the domain name by being connected to the internet.
Step 27	**signature update occur-at** {**daily** | **monthly** *day-of-month* | **weekly** *day-of-week*} *hour minute* **Example:** `Device(config-utd-mt-utd-global-threat)# signature update occur-at daily 0 0`	Configures the signature update interval parameters. This configuration will trigger the signature update to occur at midnight.
Step 28	**web-filter** **Example:** `Device(config-utd-mt-utd-global-threat)# web-filter`	This command, used in combination with the following `sourcedb` command, specifies the URL source database for web filtering.
Step 29	**sourcedb** *sourcedb-number* **Example:** `Device(config-utd-mt-utd-global-threat)# sourcedb 1`	Assigns a web filtering source database. Only one source database can be active.
Step 30	**exit** **Example:**	Exits threat inspection configuration mode.

	Command or Action	Purpose
	`Device(config-utd-mt-utd-global-threat)# exit`	
Step 31	**exit** **Example:** `Device(config-utd-mt-global)# exit`	Exits global update configuration mode.
Step 32	**threat-inspection whitelist profile** *policy-name* **Example:** `Device(config-utd-multi-tenancy)# threat-inspection whitelist profile wh101`	Associates an allowed list profile with the policy currently being configured. A similar command is used in single-tenancy, but with a `utd` keyword.
Step 33	**signature id** *id* **Example:** `Device(config-utd-mt-list)# signature id 101`	Specify the ID *id* that you have previously identified as a threat; for example, after observing the ID in an alert log file. Repeat this command for multiple signature IDs.
Step 34	**exit** **Example:** `Device(config-utd-mt-whitelist)# exit`	Exits an allowed list configuration mode.
Step 35	**threat-inspection profile** *profile-name* **Example:** `Device(config-utd-multi-tenancy)# threat-inspection profile 101`	Configures a threat inspection profile, which can be reused by multiple tenants. You can configure multiple threat-inspection profiles. Within a profile you can configure multiple allowed lists. `profile-name` is alphanumeric.

Step 36	**threat** {**detection** | **protection** }
	Example:
	`Device(config-utd-mt-threat)# threat protection`
		Specifies Intrusion Detection System (IDS) or Intrusion Prevention System (IPS) as the operating mode for the Snort engine.
		The default is **threat detection**

Step 37	**policy** {**balanced** | **connectivity** | **security**}
	Example:
	`Device(config-utd-mt-threat)# policy security`
		Configures the security policy for the Snort engine.
		• The default security policy type is **balanced**.

Step 38	**logging level**{**alert** | **crit** | **debug** | **emerg** | **err** | **info** | **notice** | **warning**}	Provides logs in one of these categories: • alert—provides alert level logs (severity=2) • crit—critical level logs (severity=3) • debug—all logs (severity=8) • emerg—emergency level logs (severity=1) • err—error level logs (severity=4) Default. • info—info level logs (severity=7)

	Command or Action	Purpose
		• notice—notice level logs (severity=6) • warning—warning level logs (severity=5)
Step 39	**whitelist profile** *profile-name* **Example:** `Device(config-utd-mt-threat)# whitelist profile wh101`	You can also specify allowed list profiles in a profile only for allowed lists in another place—the `threat-inspection whitelist profile` command above. (Optional) Enables allowed lists under the UTD engine.
Step 40	**exit** **Example:** `Device(config-utd-mt-threat)# exit`	Exits threat inspection mode.
Step 41	Repeat steps 35 to 40 to add additional threat-inspection profiles.	
Step 42	**policy** *policy-name* **Example:** `Device(config-utd-multi-tenancy)# policy pol101`	Defines the policy that will be associated with multiple tenants. A threat detection (IPS) and web filtering profile are added to the policy.
Step 43	**vrf** [*vrf-name* | global] **Example:** This example shows the configuration of two tenants (VRFs) and two policies. `Device(config-utd-mt-policy)# vrf vrf101`	Repeat the `vrf vrf-name` command for each of the VRFs (tenants) that will use the UTD policy. These VRFs previously defined, see: How to Configure VRFs for Multi-Tenancy, on page 97. Alternatively use `vrf global` to associate with the global (default) VRF and enables VRF under the interface.
Step 44	**all-interfaces** **Example:** `Device(config-utd-mt-policy)# all-interfaces`	(Optional) Associates all interfaces under the VRF with the policy.
Step 45	**threat-inspection profile** *profile-name* **Example:** `Device(config-utd-mt-policy)# threat-inspection profile 101`	(Optional) Associates the policy with a previously defined threat inspection profile, see Step 35.
Step 46	**web-filter url profile** *web-filter-profile-id* **Example:** `Device(config-utd-mt-policy)# web-filter url profile 1`	(Optional) Associates the policy with a previously defined web filtering profile, see step 15.
Step 47	**fail close** **Example:** `Device(config-utd-mt-policy)# fail close`	(Optional) Drops IPS/IDS packets on engine failure. Default is `fail open`.
Step 48	**exit**	Exits from policy configuration mode.

	Command or Action	Purpose
Step 49	Repeat steps 42 to 48 for each policy	
Step 50	**exit** **Example:** `Device(config-utd-multi-tenancy)# exit`	Exits the `utd engine standard multi-tenancy` mode. The policy configurations are applied, which may take a few minutes. During this time, further `utd engine standard multi-tenancy` configuration mode commands cannot be entered.

Step 51	**exit**
	Example:
	`Device(config)# exit`
	`Device#`	

Step 52	**show logging**
	Example:
	`Device(config)# show logging`
	`..UTD MT configuration download has started`
	`..UTD MT configuration download has completed`
		(Optional) Shows log messages that confirm whether policy configurations have been applied. Look for messages such as the following:
		`..UTD MT configuration download has started`
		`..UTD MT configuration download has completed`
		The message that includes "`download has completed`" shows that the policy configurations have been applied.

Step 53	**interface** *sub-interface* **Example:** `Device(config)# interface GigabitEthernet4.101`	Specify a sub-interface to be used for the tenant (VRF).
Step 54	**encapsulation dot1Q** *vlan-id* **Example:** `Device(config-if)# encapsulation dot1Q 101`	Applies a VLAN ID to the sub-interface.
Step 55	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vrf101`	Associates a VRF instance with the sub-interface.
Step 56	**ip address** *ip-address subnet-mask* **Example:** `Device(config-if)# ip address 111.0.0.1 255.255.255.0`	Specifies the sub-interface IP address of the VRF.
Step 57	**ip route** *ip-address subnet-mask sub-interface* **Example:** In this example, the VRF's subnet GigabitEthernet4.101 is linked to the global routing table using the static IP address 111.0.0.0 255.255.255.0. `Device(config-if)# ip route 111.0.0.0 255.255.255.0 GigabitEthernet4.101`	(Optional) This `ip route` command and the `ip route vrf` command in the following step are optional—you can use these steps if you want to configure route leaking using a static route between the VRF and the global routing table. This configures a static route to the VRF subnet from the VRF interface, so that the VRF subnet is accessible from the global routing table. For further information on

	Command or Action	Purpose
		configuring route leaking, see Route Leaking in MPLS/VPN Networks.
Step 58	**ip route vrf** *vrf-name ip-address subnet-mask* **global** **Example:** `Device(config-if)# ip route vrf vrf101 0.0.0.0 0.0.0.0 5.2.1.1 global`	(Optional) This step and the previous step are optional——you can use these steps if you want to configure route leaking using a static route between the VRF and the global routing table. For further information on configuring route leaking, see Route Leaking in MPLS/VPN Networks. Specifies the static VRF default route to the global routing table.
Step 59	**utd enable**	(Optional) Enables UTD on an interface. You can use this command if the `all-interfaces` command was not configured (in step 44).
Step 60	To configure a sub-interface for each tenant (VRF), repeat steps 53 to 59.	
Step 61	**exit**	Exits interface configuration mode.

The profiles for web filtering and threat inspection (IPS) have now been applied.

Example Configuration—Multi-Tenancy for Unified Threat Defense

This example shows a typical running configuration after configuring Multi-Tenancy for UTD for two tenants.

Note The following example mentions parameter maps `urlf-blacklist1` and `urlf-whitelist1`. The configuration of these parameter maps is not shown in the example. For further information on blocked list and approved list parameter-maps, see Configure URL-based Web Filtering with an Inline Block Page.

```
utd multi-tenancy
utd engine standard multi-tenancy
 web-filter block page profile 1
  text “This page is blocked"
 web-filter block page profile 2
  text “This page is blocked"
 web-filter url profile 1
  alert all
  blacklist
   parameter-map regex urlf-blacklist1
  whitelist
   parameter-map regex urlf-whitelist1
  categories block
   social-network
   sports
  block page-profile 1
  log level error
 web-filter url profile 2
  alert all
  blacklist
   parameter-map regex urlf-blacklist2
  categories block
   shopping
   news-and-media
   sports
   real-estate
   motor-vehicles
  block page-profile 2
  log level error
  reputation
   block-threshold low-risk
 web-filter sourcedb 1
  logging level error
 threat-inspection whitelist profile wh101
  signature id 101
 threat-inspection profile 101
  threat protection
  policy security
  logging level debug
  whitelist profile wh101
 threat-inspection profile 102
  threat detection
  policy security
  logging level debug
 utd global
  logging host 172.27.58.211
  logging host 172.27.58.212
  logging host 172.27.56.97
  threat-inspection
   signature update server cisco username abc password ]RDCe[B\^KFI_LgQgCFeBEKWP^SWZMZMb]KKAAB

   signature update occur-at daily 0 0
  web-filter
   sourcedb 1
 policy pol102
  vrf vrf102
  all-interfaces
  threat-inspection profile 102
  web-filter url profile 2
 policy pol101
  vrf vrf101
  all-interfaces
  threat-inspection profile 101
  web-filter url profile 1
  fail close
```

Verifying Unified Threat Defense Engine Standard Configuration

Use the following commands to verify your configuration.

SUMMARY STEPS

1. **enable**
2. **show utd multi-tenancy**
3. **show utd engine standard global**
4. **show utd engine standard status**
5. **show utd engine standard statistics**
6. **show utd engine standard statistics daq** [**dp** | **cp**]

7. **show utd engine standard statistics url-filtering** [**engine** | *no*]
8. **show utd engine standard statistics url-filtering vrf name** *vrf-name*
9. **show utd engine standard statistics internal**
10. **show utd engine standard logging event**
11. **show logging** | **include** CONFIG_DOWNLOAD
12. **show utd threat-inspection whitelist** [**profile** *profile-name*]
13. **show utd threat-inspection profile** *profile-name*
14. **show utd** [**policy** *profile-name*]
15. **show utd web-filter url** [**profile** *profile-name*]
16. **show utd web-filter block local-server** [**profile** *profile-name*]
17. **show utd web-filter sourcedb** [**profile** *profile-name*]
18. **show utd engine standard statistics daq dp** [**engine** *engine-num*] [**vrf** [**name** *vrf-name* | **global**]]
19. **show utd engine standard config threat-inspection whitelist** [**profile** *profile-name*]
20. **show utd engine standard config web-filter url profile** *profile-name*
21. **show utd engine standard config** [**vrf name** *vrf-name*]
22. **show utd engine standard config threat-inspection profile** *profile-name*
23. **show utd engine standard threat-inspection signature update status**
24. **show platform software qfp active feature utd config** [**vrf**[{**id** *vrf-id* | **name** *vrf-name* | **global** }]
25. **show platform software utd interfaces**
26. **show platform hardware qfp active feature utd config** [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }]
27. **show platform hardware qfp active feature utd stats** [**clear** | **divert** | **drop** | **general** | **summary**] [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }] [all] [**verbose**]
28. **show platform hardware qfp active feature utd stats summary** [**vrf name** *vrf-name* | **all**]
29. **show platform hardware qfp active feature utd stats drop all**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device# enable
```

Enables privileged EXEC mode. Enter your password if prompted.

Step 2 **show utd multi-tenancy**

Displays the current status of multi-tenancy.

Example:

```
Device# show utd multi-tenancy
Multitenancy is enabled
```

Step 3 **show utd engine standard global**

Displays the global settings for utd engine standard.

Example:

```
Device# show utd engine standard global
UTD Engine Standard Global: enabled
Threat-inspection: enabled
Web-filter: enabled
```

```
Logging:
```

Step 4 **show utd engine standard status**

Verify that the status of the UTD engine is Green.

Example:

```
Device# show utd eng standard status
Engine version       : 1.0.2_SV2983_XE_16_8

Profile              : Multi-tenancy
System memory        :
              Usage  : 3.50 %
              Status : Green
Number of engines    : 1

Engine        Running    CFT flows  Health     Reason
=======================================================
Engine(#1):   Yes        0          Green      None
=======================================================

Overall system status: Green

Signature update status:
=========================
Current signature package version: 29.0.c
Last update status: Failed
Last successful update time: None
Last failed update time: Thu Jan 11 13:34:36 2018 PST
Last failed update reason: [Errno 113] No route to host
Next update scheduled at: None
Current status: Idle
```

Step 5 **show utd engine standard statistics**

Example:

```
Device# show utd engine standard statistics
*************Engine #1*************
===============================================================================
Memory usage summary:
Total non-mmapped bytes (arena): 80125952
Bytes in mapped regions (hblkhd): 359546880
Total allocated space (uordblks): 68314032
Total free space (fordblks): 11811920
Topmost releasable block (keepcost): 112
===============================================================================
Packet I/O Totals:
Received: 49088
Analyzed: 49088 (100.000%)
Dropped: 0 ( 0.000%)
Filtered: 0 ( 0.000%)
Outstanding: 0 ( 0.000%)
Injected: 640
===============================================================================
Breakdown by protocol (includes rebuilt packets):
Eth: 49394 (100.000%)

<output removed for brevity>

Total: 49394
===============================================================================
Action Stats:
Alerts: 65 ( 0.132%)
Logged: 65 ( 0.132%)
Passed: 0 ( 0.000%)
```

Step 6 **show utd engine standard statistics daq** [**dp** | **cp**]

Show Snort DAQ statistics.

Example:

```
Device# show utd engine standard statistics daq dp
IOS-XE DAQ Counters(Engine #1):
---------------------------------
Frames received 654101
Bytes received 549106120
RX frames released 654101
Packets after vPath decap 654101
Bytes after vPath decap 516510928
Packets before vPath encap 651686
Bytes before vPath encap 514800669
Frames transmitted 651686
Bytes transmitted 544447557

<output removed for brevity>
```

Example:

```
Device# show utd engine standard statistics daq cp
IOS-XE DAQ CP Counters(Engine #1):
------------------------------------
Packets received :16353210
Bytes received :1112018252
Packets transmitted :16353210
Bytes transmitted :1700733776
Memory allocation :16353212
Memory free :16353210
CFT API error :0
VPL API error :0
Internal error :0
External error :0
Memory error :0
Timer error :0
RX ring full 0
CFT full 0
sPath lib flow handle exhausted 0
Memory status changed to yellow :1
Memory status changed to red :0
Process restart notifications :0
```

Step 7 **show utd engine standard statistics url-filtering** [**engine** | *no*]

Gives the URL statistics for all the tenants combined: the number of hits for sites on the blocked list, number of hits for sites on the allowed list, and the number of sites that are blocked by category block and reputation block.

Example:

```
Device# show utd engine standard statistics url-filtering
UTM Preprocessor Statistics
---------------------------
URL Filter Requests Sent:           377226166       379846771       381117940
URL Filter Response Received:       377009606       379622845       380892658
Blacklist Hit Count:                0               0               0
```

```
Whitelist Hit Count:                    0              0              0

Reputation Lookup Count:                376859139      379458008      380706804
Reputation Action Block:                0              0              0
Reputation Action Pass:                 307            280            102
Reputation Action Default Pass:         376858832      379457728      380706702
Reputation Score None:                  376858832      379457728      380706702
Reputation Score Out of Range:          0              0              0

Category Lookup Count:                  376859139      379458008      380706804
Category Action Block:                  0              0              0
Category Action Pass:                   307            280            102
Category Action Default Pass:           376858832      379457728      380706702
Category None:                          376858832      379457728      380706702


Device# show utd engine standard statistics url-filtering engine1
UTM Preprocessor Statistics
------------------------------------------------------
URL Filter Requests Sent:               377226166
URL Filter Response Received:           377009606
Blacklist Hit Count:                    0
Whitelist Hit Count:                    0

Reputation Lookup Count:                376859139
Reputation Action Block:                0
Reputation Action Pass:                 307
Reputation Action Default Pass:         376858832
Reputation Score None:                  376858832
Reputation Score Out of Range:          0

Category Lookup Count:                  376859139
Category Action Block:                  0
Category Action Pass:                   307
Category Action Default Pass:           376858832
Category None:                          376858832
```

Step 8 **show utd engine standard statistics url-filtering vrf name** *vrf-name*

Gives per-tenant URL statistics by using the additional parameters—**vrf name** *vrf-name* .

Example:

```
Device# show utd engine standard statistics url-filtering vrf name vrf101
UTM Preprocessor Statistics
---------------------------
URL Filter Requests Sent: 764
URL Filter Response Received: 764
Blacklist Hit Count: 3
Whitelist Hit Count: 44

Reputation Lookup Count: 764
Reputation Action Block: 0
Reputation Action Pass: 58
Reputation Action Default Pass: 706
Reputation Score None: 706
Reputation Score Out of Range: 0

Category Lookup Count: 764
Category Action Block: 5
Category Action Pass: 53
Category Action Default Pass: 706
Category None: 706
```

Step 9 **show utd engine standard statistics internal**

Example:

```
Device# show utd engine standard statistics internal
*************Engine #1*************
================================================================================
Memory usage summary:
Total non-mmapped bytes (arena): 80125952
Bytes in mapped regions (hblkhd): 359546880
Total allocated space (uordblks): 68314032
Total free space (fordblks): 11811920
Topmost releasable block (keepcost): 112
================================================================================
Packet I/O Totals:
Received: 49088
Analyzed: 49088 (100.000%)
Dropped: 0 ( 0.000%)
Filtered: 0 ( 0.000%)
Outstanding: 0 ( 0.000%)
Injected: 640
================================================================================
Breakdown by protocol (includes rebuilt packets):
Eth: 49394 (100.000%)
VLAN: 49394 (100.000%)
IP4: 49394 (100.000%)
Frag: 0 ( 0.000%)
ICMP: 5 ( 0.010%)
UDP: 2195 ( 4.444%)
TCP: 47194 ( 95.546%)

<output removed for brevity>
```

Step 10 **show utd engine standard logging event**

Displays the logs which contains alerts and URLs that are either on the blocked or allowed list per VRF.

Example:

```
Device# show utd engine standard logging event

2017/08/04-16:01:49.205959 UTC [**] [Instance_ID: 1] [**] Drop [**]
UTD WebFilter Category/Reputation [**] [URL: www.cricinfo.com] ** [Category: Sports]
** [Reputation: 96] [VRF: vrf101] {TCP} 23.72.180.26:80 -> 111.0.0.254:53509
2017/08/04-16:02:12.253330 UTC [**] [Instance_ID: 1] [**] Pass [**]
 UTD WebFilter Whitelist [**] [URL: www.espn.go.com/m]
[VRF: vrf101] {TCP} 111.0.0.254:53511 -> 199.181.133.61:80
```

Step 11 **show logging** | **include** CONFIG_DOWNLOAD

(Optional) Shows log messages that confirm whether policy configurations have been applied. Look for messages such as the following:

```
..UTD MT configuration download has started

..UTD MT configuration download has completed
```

The message `download has completed` shows that the policy configurations have been applied.

Example:

```
show# logging | include CONFIG_DOWNLOAD
Aug 23 11:34:21.250 PDT: %IOSXE_UTD-4-MT_CONFIG_DOWNLOAD: UTD MT configuration download has started
Aug 23 11:54:18.496 PDT: %IOSXE_UTD-4-MT_CONFIG_DOWNLOAD: UTD MT configuration download has completed
```

Step 12 **show utd threat-inspection whitelist** [**profile** *profile-name*]

Displays all allowed list profiles or a specific allowed list profile.

Example:

```
Device# show utd threat-inspection whitelist
Whitelist Profile: wh101
Signature ID: 101
```

Example:

```
Device# show utd threat-inspection whitelist profile wh101
Whitelist Profile: wh101
Signature ID: 101
```

Step 13 **show utd threat-inspection profile** *profile-name*

Displays the details of a threat-inspection profile specified by the *profile-name*.

Example:

```
Device# show utd threat-inspection profile 101
Threat-inspection Profile: 101
Operational Mode: Intrusion Protection
Operational Policy: Security
Logging Level: debug
Whitelist Profile: wh101
```

Step 14 **show utd** [**policy** *profile-name*]

Displays all UTD policies or a specific UTD policy.

Example:

```
Device# show utd policy pol101
Policy name: pol101
VRF name: vrf101, VRF ID: 1
Global Inspection (across above VRFs): Enabled
Threat-inspection profile: 101
Web-filter URL profile: 1
Fail Policy: Fail-open
```

Step 15 **show utd web-filter url** [**profile** *profile-name*]

Displays all URL profiles or a specific profile.

Example:

```
Device# show utd web-filter url profile 1
URL Profile: 1
Alert: all
Blacklist Parameter Map Regex: urlf-blacklist1
Whitelist Parameter Map Regex: urlf-whitelist1
Block Categories:
dating
sports
```

```
Block Page Profile 1
Log level error
reputation block-threshold high-risk
```

Step 16 **show utd web-filter block local-server** [**profile** *profile-name*]

Displays all block page profiles or a specific block page profile.

Example:

```
Device# show utd web-filter block local-server profile 2
Block Local Server Profile: 2
Content text: "Blocked by Web-Filter"
HTTP ports: 80
```

Step 17 **show utd web-filter sourcedb** [**profile** *profile-name*]

Displays all sourcedb profiles or a specific sourcedb profile.

Example:

```
Device# show utd web-filter sourcedb
SourceDB Profile: 1
database update server interval hour 0 minute 0
Fail open
Log level: error
Proxy host port 0

SourceDB Profile: 2
database update server interval hour 0 minute 0
Fail open
Log level: error
Proxy host port 0
```

Example:

```
Device# show utd web-filter sourcedb profile 1
SourceDB Profile: 1
database update server interval hour 0 minute 0
Fail open
Log level: error
Proxy host port 0
```

Step 18 **show utd engine standard statistics daq dp** [**engine** *engine-num*] [**vrf** [**name** *vrf-name* | **global**]]

Displays serviceplane data acquistion (DAQ) statistics for all VRFs or a specific VRF.

Example:

The following example shows the serviceplane data acquisition statistics for VRF vrf101.

```
Device# show utd engine standard statistics daq dp vrf name vrf101
IOS-XE DAQ Counters(Engine #1):
---------------------------------
Frames received 374509
Bytes received 303136342
RX frames released 374509
Packets after vPath decap 374509
Bytes after vPath decap 284405526
Packets before vPath encap 372883
Bytes before vPath encap 283234522
Frames transmitted 372883
```

```
Bytes transmitted 300202270

Memory allocation 781856
Memory free 749636
Memory free via timer 29420
Merged packet buffer allocation 0
Merged packet buffer free 0

VPL buffer allocation 0
VPL buffer free 0
VPL buffer expand 0
VPL buffer merge 0
VPL buffer split 0
VPL packet incomplete 0

VPL API error 0
CFT API error 0
Internal error 52
External error 0
Memory error 0
Timer error 0

Kernel frames received 373590
Kernel frames dropped 0

FO cached via timer 0
Cached fo used 0
Cached fo freed 0
FO not found 0
CFT full packets 0
```

Step 19 **show utd engine standard config threat-inspection whitelist** [**profile** *profile-name*]

Displays the details of a threat-inspection allowed list profile stored in a container.

Example:

```
Device# show utd engine standard config threat-inspection whitelist
UTD Engine Standard Configuration:

UTD threat-inspection whitelist profile table entries:
Whitelist profile: wh101
Entries: 1
```

Step 20 **show utd engine standard config web-filter url profile** *profile-name*

Displays the details of the web-filter profile stored in the container.

Example:

```
Device# show utd engine standard config web-filter url profile 1
UTD Engine Standard Configuration:

UTD web-filter profile table entries
Web-filter URL profile: 1
Whitelist:
www.espn.com
www.nbcsports.com
www.nfl.com
Blacklist:
www.cnn.com
Categories Action: Block
Categories:
Social Network
```

```
Sports
Block Profile: 1
Redirect URL: http://172.27.56.97/vrf101.html
Reputation Block Threshold: High risk
Alerts Enabled: Whitelist, Blacklist, Categories, Reputation
Debug level: Error
Conditional debug level: Error
```

Step 21 **show utd engine standard config** [**vrf name** *vrf-name*]

Displays the details of the UTD policy, threat-inspection profile and web-filter profile associated with a particular VRF.

Example:

```
Device# show utd engine standard config vrf name vrf101
UTD Engine Standard Configuration:

UTD VRF table entries:
VRF: vrf101 (1)
Policy: pol101
Threat Profile: 101
Webfilter Profile: 1
```

Step 22 **show utd engine standard config threat-inspection profile** *profile-name*

Displays the details of a specific threat-inspection profile.

Example:

```
Device# show utd engine standard config threat-inspection profile 101
UTD Engine Standard Configuration:

UTD threat-inspection profile table entries:
Threat profile: 101
Mode: Intrusion Prevention
Policy: Security
Logging level: Debug
Whitelist profile: wh101

Description:
Displays the details of a threat-inspection profile stored in the container.
```

Step 23 **show utd engine standard threat-inspection signature update status**

Shows the output of the current signature package version, previous signature package version, and last status update.

Example:

```
Device# show utd engine standard threat-inspection signature update status
Current signature package version: 29.0.c
Current signature package name: default
Previous signature package version: None
---------------------------------------
Last update status: Failed
---------------------------------------
Last successful update time: None
Last successful update method: None
Last successful update server: None
Last successful update speed: None
---------------------------------------
Last failed update time: Thu Jan 11 13:34:36 2018 PST
Last failed update method: Manual
Last failed update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
Last failed update reason: [Errno 113] No route to host
```

```
---------------------------------------
Last attempted update time: Thu Jan 11 13:34:36 2018 PST
Last attempted update method: Manual
Last attempted update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
---------------------------------------
Total num of updates successful: 0
Num of attempts successful: 0
Num of attempts failed: 1
Total num of attempts: 1
---------------------------------------
Next update scheduled at: None
---------------------------------------
Current status: Idle
```

Step 24 **show platform software qfp active feature utd config** [**vrf**[{**id** *vrf-id* | **name** *vrf-name* | **global** }]

Shows the service node statistics. The VRF information can only be shown in the case of multi-tenancy. Displays the data plane UTD configuration. In the following example the security context information is highlighted.

Example:

```
Device# Global configuration
  NAT64: disabled
  SN threads: 12
  CFT inst_id 0 feat id 0 fo id 0 chunk id 4
  Context Id: 0, Name: Base Security Ctx
   Ctx Flags: (0xf0000)
        Engine: Standard
        SN Redirect Mode : Fail-close, Divert
        Threat-inspection: Enabled, Mode: IPS
        Domain Filtering : Not Enabled
        URL Filtering    : Not Enabled
   SN Health: Green
```

Step 25 **show platform software utd interfaces**

Example:

```
Device# show platform software utd interfaces

UTD interfaces
All dataplane interfaces
```

Step 26 **show platform hardware qfp active feature utd config** [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }]

Show UTD datapath configuration and status.

Example:

```
Device# show platform hardware qfp active feature utd config vrf name vrf101
Global configuration
  NAT64: disabled
  Drop pkts: disabled
  Multi-tenancy: enabled
  Data plane initialized: yes
  SN threads: 12
  CFT inst_id 0 feat id 1 fo id 1 chunk id 8
  SN Health: Green
```

Step 27 **show platform hardware qfp active feature utd stats** [**clear** | **divert** | **drop** | **general** | **summary**] [**vrf** {**id** *vrf-id* | **name** *vrf-name* | **global** }] [all] [**verbose**]

Displays dataplane UTD statistics, including counts of zeros

clear—Clear Statistics

divert—Display AppNav Redirect Statistics

drop—Display Drop Statistics

general—Display General Statistics

summary—Display Summary Statistics

verbose—Display Verbose Statistics

vrf Display per VRF stats—The VRF information can only be entered if multi-tenancy is enabled.

id—display stats associated with the VRF id

name—display stats associated with the VRF with the provided name

global—display the stats associated with the global VRF (i.e vrf-id 0)

Example:

```
Device# show platform hardware qfp active feature utd stats

Summary Statistics:
TCP Connections Created 29893
UDP Connections Created 24402
ICMP Connections Created 796
Pkts dropped pkt 258
byt 66365
Pkts entered policy feature pkt 715602
byt 562095214
Pkts entered divert feature pkt 662014
byt 516226302
Pkts slow path pkt 55091
byt 4347864
Pkts Diverted pkt 662014
byt 516226302
Pkts Re-injected pkt 659094
byt 514305557

Would-Drop Statistics:

Service Node flagged flow for dropping 258

General Statistics:
Non Diverted Pkts to/from divert interface 1022186
Inspection skipped - UTD policy not applicable 1081563

<output removed for brevity>
```

Example:

Step 28 **show platform hardware qfp active feature utd stats summary** [**vrf name** *vrf-name* | **all**]

Displays information about all VRFs or a specific VRF, taken from the summary option of the **show platform hardware qfp active feature utd stats** command.

Example:

```
Device# show platform hardware qfp active feature utd stats vrf name vrf101
Security Context: Id:1 Name: 1 : vrf101

Summary Statistics:
TCP Connections Created 18428
UDP Connections Created 13737
ICMP Connections Created 503
Pkts dropped pkt 258
byt 66365
Pkts entered policy feature pkt 407148
byt 296496913
Pkts entered divert feature pkt 383176
byt 283158966
Pkts slow path pkt 32668
byt 2571632
Pkts Diverted pkt 383176
byt 283158966
Pkts Re-injected pkt 381016
byt 281761395


<output removed for brevity>
```

Step 29 **show platform hardware qfp active feature utd stats drop all**

Displays information from all the VRFs taken from the drop option of the **show platform** command.

Example:

```
Device# show platform hardware qfp active feature utd stats drop all

Would-Drop Statistics:

No diversion interface                                                     0
No egress interface                                                        0
Inspection service down                                                    0
Could not find divert interface                                            0
Could not find divert fib                                                  0
UTD FIB did not contain oce_chain                                          0
Invalid IP version                                                         0
IPS not supported                                                          0
Re-inject Error                                                            0
Service Node flagged flow for dropping                                  1225
Could not attach feature object                                            0
Could not allocate feature object                                          0
Error getting feature object                                               0
Policy: could not create connection                                        0
NAT64 Interface Look up Failed                                             0
Decaps: VPATH connection establishment error                               0
Decaps: VPATH could not find flow, no tuple                                0
Decaps: VPATH notification event error                                     0
Decaps: Could not delete flow                                              0
Decaps: VPATH connection classification error                              0
Encaps: Error retrieving feature object                                    0
Encaps: Flow not classified                                                0
Encaps: VPATH connection specification error                               0
Encaps: VPATH First packet meta-data failed                                0
Encaps: VPATH No memory for meta-data                                      0
Encaps: VPATH Could not add TLV                                            0
Encaps: VPATH Could not fit TLV into memory                                0
Service Node Divert Failed                                                 0
No feature object                                                          0
Service Node not healthy                                                 123
Could not allocate VRF meta-data                                           0
Could not allocate debug meta-data                                         0
Packet was virtually fragmented (VFR)                                      0
IPv6 Fragment                                                              0
IPv4 Fragment                                                              0
```

Troubleshooting Multi-Tenancy for Unified Threat Defense

Traffic is not Diverted

Problem Traffic is not diverted.

Possible Cause Vitual-service may not be activated.

Solution Check whether the virtual-service is activated by using the **show virtual-service list** command. The following is sample output from the command:

```
Device# show virtual-service list

Virtual Service List:


Name Status Package Name
------------------------------------------------------------------------------
snort Activated utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
```

Possible Cause Unified threat defense (UTD) may not be enabled for specified interface or interfaces.

Solution Use the **show platform software utd global** command to verify if UTD is enabled for the interface:

```
Device# show platform software utd global

UTD Global state
Engine               : Standard
Global Inspection    : Disabled
Operational Mode     : Intrusion Prevention
Fail Policy          : Fail-open
Container techonlogy : LXC
Redirect interface   : VirtualPortGroup1
UTD interfaces
GigabitEthernet0/0/0
```

Possible Cause The service node may not be working properly.

Solution Use the **show platform hardware qfp active feature utd config** command to verify if the health of the service node is green:

```
Device# show platform hardware qfp active feature utd config

Global configuration
NAT64: disabled
SN threads: 12
CFT inst_id 0 feat id 0 fo id 0 chunk id 4
Context Id: 0, Name: Base Security Ctx
Ctx Flags: (0x60000)
Engine: Standard
SN Redirect Mode : Fail-open, Divert
```

```
Threat-inspection: Enabled, Mode: IDS
Domain Filtering : Not Enabled
URL Filtering : Not Enabled
SN Health: Green
```

Solution Alternatively, in the case of multi-tenancy, you can use the **show platform hardware qfp active feature utd config vrf name** *vrf-name* command to verify if the health of the service node, for a specific VRF, is green:

```
Device# show platform hardware qfp active feature utd config vrf name vrf102
Global configuration
  NAT64: disabled
  Drop pkts: disabled
  Multi-tenancy: enabled
  Data plane initialized: yes
  SN threads: 12
  CFT inst_id 0 feat id 0 fo id 0 chunk id 4
  SN Health: Green
```

Possible Cause The Snort process may not be activated.

Solution Use the **show virtual-service detail** command to verify if the Snort process is up and running:

```
Device# show virtual-service detail

Virtual service UTDIPS detail
  State                 : Activated
  Owner                 : IOSd
  Package information
    Name                : utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Path                : bootflash:/utdsnort.1_0_1_SV2982_XE_16_3.20160701_131509.ova
    Application
      Name              : UTD-Snort-Feature
      Installed version : 1.0.1_SV2982_XE_16_3
      Description       : Unified Threat Defense
    Signing
      Key type          : Cisco development key
      Method            : SHA-1
    Licensing
      Name              : Not Available
      Version           : Not Available

  Detailed guest status

----------------------------------------------------------------------
Process               Status            Uptime           # of restarts
----------------------------------------------------------------------
climgr                 UP         0Y 0W 0D  0: 0:35        1
logger                 UP         0Y 0W 0D  0: 0: 4        0
snort_1                UP         0Y 0W 0D  0: 0: 4        0
Network stats:
eth0: RX  packets:43, TX  packets:6
eth1: RX  packets:8, TX  packets:6

Coredump file(s): lost+found

  Activated profile name: None
  Resource reservation
    Disk                : 736 MB
    Memory              : 1024 MB
    CPU                 : 25% system CPU

  Attached devices
    Type              Name        Alias
    ---------------------------------------------
    NIC               ieobc_1     ieobc
    NIC               dp_1_0      net2
    NIC               dp_1_1      net3
    NIC               mgmt_1      mgmt
    Disk              _rootfs
    Disk              /opt/var
    Disk              /opt/var/c
    Serial/shell                  serial0
    Serial/aux                    serial1
    Serial/Syslog                 serial2
    Serial/Trace                  serial3
    Watchdog          watchdog-2

  Network interfaces
    MAC address             Attached to interface
    ------------------------------------------------------
    54:0E:00:0B:0C:02       ieobc_1
    A4:4C:11:9E:13:8D       VirtualPortGroup0
    A4:4C:11:9E:13:8C       VirtualPortGroup1
    A4:4C:11:9E:13:8B       mgmt_1

  Guest interface
  ---
  Interface: eth2
  ip address: 48.0.0.2/24
Interface: eth1
  ip address: 47.0.0.2/24

  ---

  Guest routes
  ---
  Address/Mask                         Next Hop                          Intf.
-------------------------------------------------------------------------------
0.0.0.0/0                            48.0.0.1                          eth2
0.0.0.0/0                            47.0.0.1                          eth1

  ---

  Resource admission (without profile) : passed
    Disk space    : 710MB
    Memory        : 1024MB
    CPU           : 25% system CPU
    VCPUs         : Not specified
```

Possible Cause The AppNav tunnel may not be activated.

Solution Use the **show service-insertion type utd service-node-group** and **show service-insertion type utd service-context** commands to verify if the AppNav tunnel is activated.

Solution The following is sample output from the **show service-insertion type utd service-node-group** command:

```
Device# show service-insertion type utd service-node-group

Service Node Group name : utd_sng_1
Service Context : utd/1
Member Service Node count : 1


Service Node (SN) : 30.30.30.2
Auto discovered : No
SN belongs to SNG : utd_sng_1
```

```
Current status of SN : Alive
Time current status was reached : Tue Jul 26 11:57:48 2016

Cluster protocol VPATH version : 1
Cluster protocol incarnation number : 1
Cluster protocol last sent sequence number : 1469514497
Cluster protocol last received sequence number: 1464
Cluster protocol last received ack number : 1469514496
```

Solution The following is sample output from the **show service-insertion type utd service-context** command:

```
Device# show service-insertion type utd service-context

Service Context : utd/1
Cluster protocol VPATH version : 1
Time service context was enabled : Tue Jul 26 11:57:47 2016
Current FSM state : Operational
Time FSM entered current state : Tue Jul 26 11:57:58 2016
Last FSM state : Converging
Time FSM entered last state : Tue Jul 26 11:57:47 2016
Cluster operational state : Operational

Stable AppNav controller View:
30.30.30.1

Stable SN View:
30.30.30.2

Current AppNav Controller View:
30.30.30.1

Current SN View:
30.30.30.2
```

Possible Cause Check data plane UTD statistics for the status of the traffic. If the traffic is not diverted, the number of packets diverted and rejected will be zero. If the numbers are nonzero, then traffic diversion is happening, and the Snort sensor is resending packets back to the dataplane.

Solution Use the **show platform hardware qfp active feature utd stats** command to verify the status of the traffic.

```
Device# show platform hardware qfp active feature utd stats

Security Context:    Id:0    Name: Base Security Ctx

Summary Statistics:
Active Connections                                                     29
TCP Connections Created                                            712910
UDP Connections Created                                                80
Pkts entered policy feature                          pkt          3537977
                                                     byt        273232057
Pkts entered divert feature                          pkt          3229148
                                                     byt        249344841
Pkts slow path                                       pkt           712990
                                                     byt         45391747
Pkts Diverted                                        pkt          3224752
                                                     byt        249103697
Pkts Re-injected                                     pkt          3224746
                                                     byt        249103373
….
```

Solution Alternatively, in the case of multi-tenancy, you can use the **show platform hardware qfp active feature utd stats vrf name** *vrf-name* command to verify the status of the traffic, for a specific VRF.

```
Device# show platform hardware qfp active feature utd stats vrf name vrf 101

Security Context:    Id:1    Name: 1 : vrf101

Summary Statistics:
Active Connections                                                          2
TCP Connections Created                                                 34032
UDP Connections Created                                                 11448
ICMP Connections Created                                                   80
Pkts dropped                                        pkt                   626
                                                    byt                323842
Pkts entered policy feature                         pkt                995312
                                                    byt             813163885
Pkts entered divert feature                         pkt                639349
                                                    byt             420083106
Pkts slow path                                      pkt                 45560
                                                    byt               7103132
Pkts Diverted                                       pkt                638841
                                                    byt             419901335
Pkts Re-injected                                    pkt                630642
                                                    byt             412139098
….
```

Signature Update is not Working

Problem Signature update from Cisco Borderless Software Distribution (BSD) server is not working.

> **Possible Cause** Signature update may have failed due to various reasons. Check for the reason for the last failure to update the signatures.

Solution Use the **show utd engine standard threat-inspection signature update status** command to display the reason for the last failure to update the signatures:

```
Device# show utd eng standard threat-inspection signature update status
Current signature package version: 29.0.c
Current signature package name: default
Previous signature package version: None
---------------------------------------
Last update status: Failed
---------------------------------------
Last successful update time: None
Last successful update method: None
Last successful update server: None
Last successful update speed: None
---------------------------------------
Last failed update time: Thu Jan 11 13:34:36 2018 PST
Last failed update method: Manual
Last failed update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
Last failed update reason: [Errno 113] No route to host
---------------------------------------
Last attempted update time: Thu Jan 11 13:34:36 2018 PST
Last attempted update method: Manual
Last attempted update server: http://172.27.57.252/UTD-STD-SIGNATURE-2983-1-S.pkg
---------------------------------------
```

```
Total num of updates successful: 0
Num of attempts successful: 0
Num of attempts failed: 1
Total num of attempts: 1
---------------------------------------
Next update scheduled at: None
---------------------------------------
Current status: Idle
```

Possible Cause Domain Name System (DNS) is not configured correctly.

Solution Use the **show running-config | i name-server** command to display the name server details:

```
Device#  show run | i name-server

ip name-server 10.104.49.223
```

Possible Cause System error—Failed to process the username and password combination.

Solution Ensure that you have provided the correct credentials for signature package download.

Signature Update from the Local Server is not Working

Problem Signature update from the local server not working.

Possible Cause Last failure Reason: Invalid scheme—only HTTP/HTTPS supported.

Solution Ensure that you have provided the HTTP or secure HTTP (HTTPS) as the local download method.

Possible Cause Last failure Reason: Name or service not known.

Solution Ensure that the hostname or IP address provided for the local server is correct.

Possible Cause Last failure Reason: Credentials not supplied.

Solution Ensure that you have provided the credentials for local HTTP/HTTPS server.

Possible Cause Last failure Reason: File not found.

Solution Ensure that the signature file name or URL that you have provided is correct.

Possible Cause Last failure Reason: Download corrupted.

Solution

- Verify whether the retry signature update is corrupted as the previous signature download.
- Ensure that the correct signature package is available.

Logging to IOSd Syslog is not Working

Problem Logging to IOSd syslog is not working.

Possible Cause Logging to syslog may not be configured in the unified threat defense (UTD) configuration.

Solution Use the **show utd engine standard config** command to display the UTD configuration and to ensure that logging to syslog is configured.

```
Device# show utd engine standard config

UTD Engine Standard Configutation:
  Operation Mode : Intrusion Prevention
  Policy         : Security
```

```
Signature Update:
  Server    : cisco
  User Name : ccouser
  Password  : YEX^SH\fhdOeEGaOBIQAIcOVLgaVGf
  Occurs-at : weekly ;  Days:0 ; Hour: 23; Minute: 50

Logging:
  Server    :   IOS Syslog; 10.104.49.223
  Level     : debug

Whitelist Signature IDs:
  28878
```

Solution Use the following **show utd engine standard logging events** command to display the event logs for the UTD engine.

```
Device# show utd engine standard logging events

2016/06/13-14:32:09.524475 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected]
[Priority: 1] [VRF_ID: 2] {UDP} 11.1.1.10:58016 -> 21.1.1.10:53
2016/06/13-14:32:21.524988 IST [**] [Instance_ID: 1] [**] Drop [**] [1:30561:1]
BLACKLIST DNS request for known malware domain domai.ddns2.biz -
Win.Trojan.Beebone [**] [Classification: A Network Trojan was Detected] [Priority: 1]
[VRF_ID: 2] {UDP} a000:0:0:0:0:0:0:10:59964 -> b000:0:0:0:0:0:0:10:53
```

Logging to an External Server is not Working

Problem Logging to an external server is not working.

Possible Cause Syslog may not be running on the external server.

Solution Verify whether syslog server is running on the external server. Configure the following command on the external server to view its status:

```
ps -eaf | grep syslog

 root 2073 1 0 Apr12 ? 00:00:02 syslogd -r -m
```

Possible Cause Connectivity between unified threat defense (UTD) Linux Container (LXC) and external server may be lost.

Solution Verify the connectivity from the management interface to the external syslog server.

UTD Conditional Debugging

Conditional debugging is supported by multi-tenancy for Unified Threat Defense. For further details about how to configure conditional debugging, see:

http://www.cisco.com/c/en/us/td/docs/routers/asr1000/troubleshooting/guide/Tblshooting-xe-3s-asr-1000-book.html#task_AC969BB06B414DCBBDEF7ADD29EF8131