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.

- Restrictions for Snort IPS, on page 1
- Information About Snort IPS, on page 2
- How to Deploy Snort IPS, on page 8
- Configuration Examples for Snort IPS, on page 21
- Examples for Displaying Active Signatures, on page 27
- Verifying the Integrated Snort IPS Configuration, on page 28
- Deploying Snort IPS Using Cisco Prime CLI Templates, on page 35
- Migrating to IOx Container, on page 36
- Troubleshooting Snort IPS, on page 39
- Additional References for Snort IPS, on page 46
- Feature Information for Snort IPS, on page 46

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 1: 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)
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)

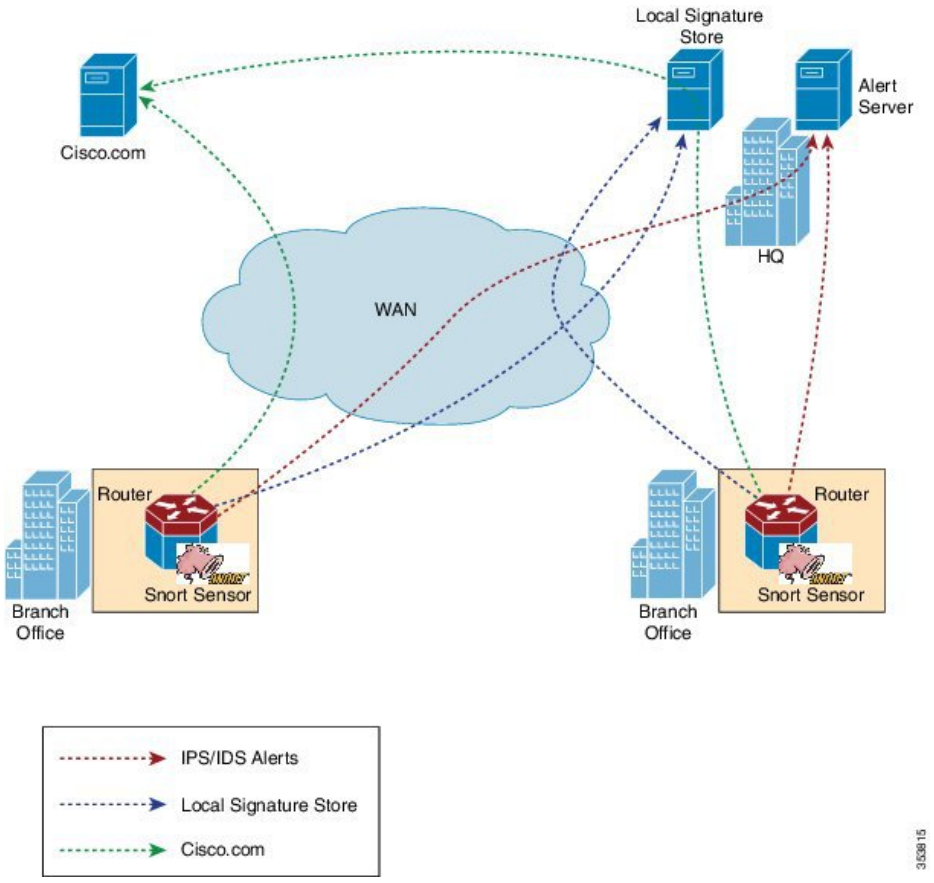
Platform	Profile	Virtual Service Resource Requirements		Platform Requirements
		System CPU	Memory	
Cisco 4351 ISR	Low (Default)	25%	Min: 1GB (RAM) Min: 750MB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	Medium	50%	Min: 2GB (RAM) Min: 1GB (Disk/Flash)	Min: 8GB (RAM) Min: 8GB(Disk/Flash)
	High	75%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

Platform	Profile	Virtual Service Resource Requirements		Platform Requirements
		System CPU	Memory	
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 1: 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.

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.
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.

	Command or Action	Purpose
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:** `Device(config-virt-serv)# vnic gateway VirtualPortGroup 0`	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. • 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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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**. • 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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.
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:**	Enables the logging of critical messages to the IOSd syslog.

	Command or Action	Purpose
	`Device(config-utd-eng-std)# logging 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:** `Device(config-utd-eng-std-insp)# exit`	Exits UTD standard engine configuration mode and returns to global configuration mode.
Step 20	**utd** **Example:** `Device(config)# utd`	Enables unified threat defense (UTD) and enters UTD configuration mode.

	Command or Action	Purpose
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:** `Device(config-cmap)# exit`	Exits class-map configuration mode and returns to global configuration mode.
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.

	Command or Action	Purpose
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 2: 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.