Get Started with Snort 3 Network Analysis Policies

This chapter provides an insight into network analysis policy basics, perquisites, and how to manage network analysis policies. It also provides information on custom network analysis policy creation and network analysis policy settings.

Overview of Network Analysis Policies

Network analysis policies govern many traffic preprocessing options, and are invoked by advanced settings in your access control policy. Network analysis-related preprocessing occurs after Security Intelligence matching and SSL decryption, but before intrusion or file inspection begins.

By default, the system uses the *Balanced Security and Connectivity* network analysis policy to preprocess all traffic handled by an access control policy. However, you can choose a different default network analysis policy to perform this preprocessing. For your convenience, the system provides a choice of several non-modifiable network analysis policies, which are tuned for a specific balance of security and connectivity by the Cisco Talos Intelligence Group (Talos). You can also create a custom network analysis policy with custom preprocessing settings.

Tip System-provided intrusion and network analysis policies are similarly named but contain different configurations. For example, the Balanced Security and Connectivity network analysis policy and the Balanced Security and Connectivity intrusion policy work together and can both be updated in intrusion rule updates. However, the network analysis policy governs mostly preprocessing options, whereas the intrusion policy governs mostly intrusion rules. Network analysis and intrusion policies work together to examine your traffic.

You can also tailor traffic preprocessing options to specific security zones, networks, and VLANs by creating multiple custom network analysis policies, then assigning them to preprocess different traffic. (Note that ASA FirePOWER cannot restrict preprocessing by VLAN.)

Manage Network Analysis Policies

In a multidomain deployment, the system displays policies created in the current domain, which you can edit. It also displays policies created in ancestor domains, which you cannot edit. To view and edit policies created in a lower domain, switch to that domain.

Under your user name in the toolbar, the system displays a tree of available domains. To switch domains, choose the domain you want to access.

Procedure

Step 1 Choose one of the following paths to access the network analysis policy.

- **Policies** > **Access Control**, then click **Network Analysis Policy**
- **Policies** > **Access Control** > **Intrusion**, then click **Network Analysis Policies**
- **Policies** > **Intrusion** > **Network Analysis Policies**

Note If your custom user role limits access to the first path listed here, use the second path to access the policy.

Step 2 Manage your network analysis policy:

- Compare—Click **Compare Policies**; see *Comparing Policies* in the *Cisco Secure Firewall Management Center Configuration Guide*.

  Note You can compare Snort 2 policies only.

- Create—If you want to create a new network analysis policy, click **Create Policy**.

  Two versions of the network analysis policy are created, a **Snort 2 Version** and a **Snort 3 Version**.

  - For the Snort 2 version, see *Custom Network Analysis Policy Creation for Snort 2* in the *Cisco Secure Firewall Management Center Configuration Guide*.
  - For the Snort 3 version, see Custom Network Analysis Policy Creation for Snort 3, on page 5.

- Delete—If you want to delete a network analysis policy, click the **Delete** icon, then confirm that you want to delete the policy. You cannot delete a network analysis policy if an access control policy references it.

  If the controls are dimmed, the configuration belongs to an ancestor domain, or you do not have permission to modify the configuration.

- Edit—If you want to edit an existing network analysis policy, click the **Edit** icon.

  If **View** (👁) appears instead, the configuration belongs to an ancestor domain, or you do not have permission to modify the configuration.

- Report—Click the **Report** icon; see *Generating Current Policy Reports* in the *Cisco Secure Firewall Management Center Configuration Guide*.

Snort 3 Definitions and Terminologies for Network Analysis Policy

The following table lists the Snort 3 concepts and terms used in the Network Analysis Policy.

Table 1: Snort 3 Definitions and Terminologies for Network Analysis Policy

Term	Description
Inspectors	Inspectors are plugins that process packets (similar to the Snort 2 preprocessor).
Binder inspector	Binder inspector defines the flow when a particular inspector has to be accessed and taken into consideration. When the traffic matches the conditions defined in the binder inspector, only then do the values/configurations for that inspector come into effect. For more information, see *Binder Inspector* in Custom Network Analysis Policy Creation for Snort 3, on page 5.
Singleton inspectors	Singleton inspectors contain one instance. These inspectors do not support adding more instances like multiton inspectors. Settings of singleton inspector are applied to the entire traffic matching that inspector and not to a specific traffic segment. For more information, see *Singleton Inspectors* in Custom Network Analysis Policy Creation for Snort 3, on page 5.

Term	Description
Multiton inspectors	Multiton inspectors contain multiple instances which you can configure as needed. These inspectors support configuring settings based on specific conditions, such as network, port, and VLAN. One set of supported settings is called an instance. For more information, see *Multiton Inspectors* in Custom Network Analysis Policy Creation for Snort 3, on page 5.
Schema	The schema file is based on the OpenAPI JSON specification, and it validates the content that you upload or download. You can download the schema file and open it using any third-party JSON editor, such as Swagger editor. The schema file helps you to identify what parameters can be configured for inspectors with their corresponding allowed values, range, and accepted patterns to be used. For more information, see Customize the Network Analysis Policy, on page 13.
Sample file	It is a pre-existing template that contains example configurations to help you with configuring the inspectors. You can refer to the example configurations included in the sample file and make any changes that you may require. For more information, see Customize the Network Analysis Policy, on page 13.
Full configuration	You can download the entire inspector configurations in a single file. All information regarding the inspector configuration is available in this file. The full configuration is a merged configuration of the default configuration (rolled out as a part of the LSP updates by Cisco Talos) and the custom NAP inspector configurations. For more information, see Customize the Network Analysis Policy, on page 13.

Term	Description
Overridden configuration	In the **Snort 3 Version** of the network analysis policy page: • Under **Actions** > **Upload**, you can click **Overridden Configuration** to upload the JSON file that contains the overridden configuration. • Under **Actions** > **Download**, you can click **Overridden Configuration** to download the inspector configuration that has been overridden. If you have not overridden any inspector configuration, then this option is disabled. When you override the inspector configuration, then this option is enabled automatically to allow you to download. For more information, see Customize the Network Analysis Policy, on page 13.

Related Topics

Custom Network Analysis Policy Creation for Snort 3, on page 5

Customize the Network Analysis Policy, on page 13

Network Analysis Policy Mapping, on page 10

Prerequisites for Network Analysis and Intrusion Policies

To allow the Snort inspection engine to process traffic for intrusion and malware analysis, you must have the IPS license enabled for the Threat Defense device.

You must be an Admin user to manage network analysis, intrusion policies, and perform migration tasks.

Custom Network Analysis Policy Creation for Snort 3

The default network analysis policy is tuned for typical network requirements and optimal performance. Usually, the default network analysis policy suffices most network requirements and you might not need to customize the policy. However, when you have a specific network requirement or when you are facing performance issues, the default network analysis policy can be customized. Note that customizing the network analysis policy is an advanced configuration that should be done only by advanced users or Cisco support.

Network analysis policy configuration for Snort 3 is a data-driven model, which is based on JSON and JSON Schema. Schema is based on the OpenAPI specification, and it helps you get a view of the supported inspectors, settings, settings type, and valid values. The Snort 3 inspectors are plugins that process packets (similar to the Snort 2 preprocessor). Network analysis policy configuration is available to download in the JSON format.

In Snort 3, the list of inspectors and settings are not in a one-to-one mapping with the Snort 2 list of preprocessors and settings. Also, the number of inspectors and settings available in management center is a subset of the inspectors and settings that Snort 3 supports. See https://snort.org/snort3 for more information on Snort 3. See https://www.cisco.com/go/snort3-inspectors for more information on the inspectors available in management center.

Note

- While upgrading the management center to the 7.0 release, the changes that were done in the Snort 2 version of the network analysis policy are not migrated to Snort 3 after the upgrade.
- Unlike the intrusion policy, there is no option to synchronize Snort 2 network analysis policy settings to Snort 3.

Default Inspector Updates

Lightweight Security Package (LSP) updates may contain new inspectors or modifications to integer ranges for existing inspector configurations. Following the installation of an LSP, new inspectors and/or updated ranges will be available under **Inspectors** in the **Snort 3 Version** of your network analysis policy.

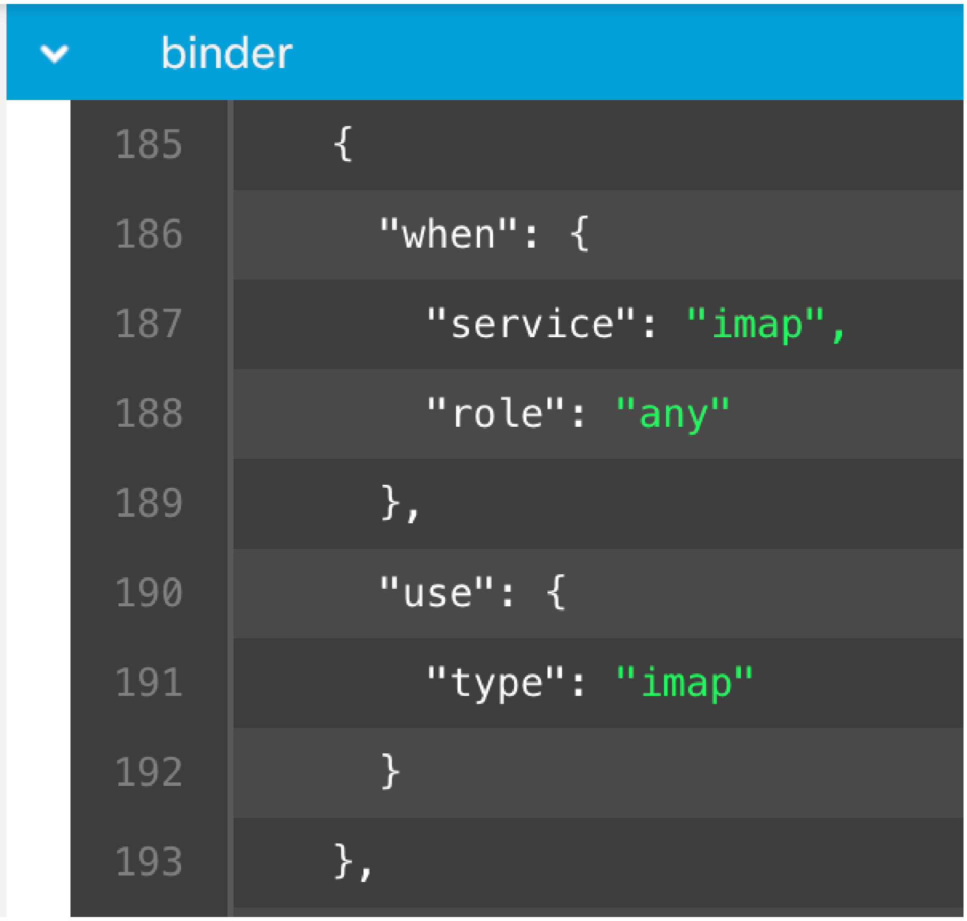
Binder Inspector

Binder inspector defines the flow when a particular inspector has to be accessed and taken into consideration. When the traffic matches the conditions defined in the binder inspector, only then the values/configurations for that inspector come into effect. For example:

For the *imap* inspector, the binder defines the following condition when it has to be accessed. That is when:

- Service is equal to imap.
- Role is equal to any.

If these conditions are met, then use the type imap.



```
    {
      "when": {
        "service": "imap",
        "role": "any"
      },
      "use": {
        "type": "imap"
      }
    },
```

Singleton Inspectors

Singleton inspectors contain a single instance. These inspectors do not support adding more instances like multiton inspectors. Settings of singleton inspector are applied to the entire traffic and not to a specific traffic segment.

For example:

```
{
   "normalizer":{
      "enabled":true,
      "type":"singleton",
      "data":{
         "ip4":{
            "df":true
         }
      }
   }
}
```

Multiton Inspectors

Multiton inspectors contain multiple instances which you can configure as needed. These inspectors support configuring settings based on specific conditions, such as network, port, and VLAN. One set of supported settings is called an instance. There is a default instance, and you can also add additional instances based on specific conditions. If the traffic matches that condition, the settings from that instance are applied. Otherwise, the settings from the default instance are applied. Also, the name of the default instance is the same as the inspector's name.

For a multiton inspector, when you upload the overridden inspector configuration, you also need to include/define a matching binder condition (conditions under when the inspector has to be accessed or used) for each instance in the JSON file, otherwise, the upload will result in an error. You can also create new instances, but make sure that you include a binder condition for every new instance that you create to avoid errors.

For example:

- Multiton inspector where the default instance is modified.

```
{
   "http_inspect":{
      "enabled":true,
      "type":"multiton",
      "instances":[
         {
            "name":"http_inspect",
            "data":{
               "response_depth":5000
            }
         }
      ]
   }
}
```

- Multiton inspector where the default instance and default binder is modified.

```
{
   "http_inspect":{
      "enabled":true,
      "type":"multiton",
      "instances":[
         {
            "name":"http_inspect",
            "data":{
               "response_depth":5000
            }
         }
      ]
   },
   "binder":{
      "type":"binder",
      "enabled":true,
      "rules":[
         {
            "use":{
               "type":"http_inspect"
            },
            "when":{
               "role":"any",
               "ports":"8080",
               "proto":"tcp",
               "service":"http"
            }
         }
      ]
   }
}
```

- Multiton inspector where a custom instance and a custom binder is added.

```
{
   "http_inspect":{
      "enabled":true,
      "type":"multiton",
      "instances":[
         {
            "name":"http_inspect1",
            "data":{
               "response_depth":5000
            }
         }
      ]
   },
   "binder":{
      "type":"binder",
      "enabled":true,
      "rules":[
         {
            "use":{
               "type":"http_inspect",
               "name":"http_inspect1"
            },
            "when":{
               "role":"any",
               "ports":"8080",
               "proto":"tcp",
               "service":"http"
            }
         }
      ]
   }
}
```

Common Industrial Protocol Safety

Common Industrial Protocol (CIP) Safety is a set of extensions to the CIP that enables the safe operation of devices. It also provides fail-safe communication between different nodes on a CIP network.

The CIP Safety protocol comprises two main components:

- CIP Safety segments—Used in Forward Open messages to exchange safety parameters for the subsequent safety session.
- CIP Safety messages—Used to exchange actual safety information.

The CIP inspector detects and identifies:

- CIP as a service and client
- Payloads, such as CIP Read, CIP Admin, CIP Infrastructure, and CIP Write

  The CIP inspector can parse the CIP segments and detect the CIP Safety segments in the Forward Open requests.

To test the CIP Safety feature, you must enable the CIP inspector. See Detect and Block Safety Segments in CIP Packets, on page 10.

Detect and Block Safety Segments in CIP Packets

Use case: To detect and block CIP safety segments while allowing other CIP packets:

- Create a custom network analysis policy called **cip_safety**.
- Create access control rules in your access control policy to block CIP Safety and to allow all other packets.

To test the CIP Safety feature, enable the CIP inspector in the management center and assign it to an access control policy.

Procedure

Step 1 Go to **Policies** > **Intrusion** > **Network Analysis Policies**.

Step 2 Click the **Snort 3 Version** of the network analysis policy **cip_safety** that you created.

Step 3 Under **Inspectors**, click **cip** to expand it.

The default configuration appears in the left column and the overridden configuration appears in the right column under the inspector.

Step 4 Under **Overridden Configuration** on the right column, click the **Edit Inspector** icon and change the "enabled" field in **cip** from false (default) to true.

Step 5 Click **OK**.

Step 6 Click **Save**.

Step 7 To assign the **cip** inspector to the access control policy, choose **Policies** > **Access Control** > **Edit** and choose the **Advanced Settings** option from the **More** drop-down arrow at the end of the packet flow line.

Step 8 Click **Edit** (✏) next to **Network Analysis and Intrusion Policies**.

Step 9 In the **Network Analysis and Intrusion Policies** window, choose the access control policy **cip_safety** that you created from the **Default Network Analysis Policy** drop-down list.

The CIP inspector is now enabled in the management center and you can create the custom access control rules to block CIP Safety and to allow all other CIP packets.

Step 10 After you send live traffic containing CIP Safety packet flows, go to **Connection Events** to verify that the payload is the expected payload that contains CIP Safety packet logs for the detection and block use case as mentioned in this procedure. **CIP** is detected as an application protocol and client (see the **Application Protocol** and **Client** fields), and **CIP Safety** is shown under the **Web Application** field.

Network Analysis Policy Mapping

For network analysis policies, Cisco Talos provides mapping information, which is used to find the corresponding Snort 2 version of the policies for the Snort 3 version.

This mapping ensures that the Snort 3 version of policies has its equivalent Snort 2 version.

View Network Analysis Policy Mapping

Procedure

Step 1 Go to **Policies** > **Intrusion** > **Network Analysis Policies**.

Step 2 Click **NAP Mapping**.

Step 3 Expand the arrow for **View Mappings**.

The Snort 3 network analysis policies that are automatically mapped to a Snort 2 equivalent policy are displayed.

Step 4 Click **OK**.

Create a Network Analysis Policy

All the existing network analysis policies are available in management center with their corresponding Snort 2 and Snort 3 versions. When you create a new network analysis policy, it is created with both the Snort 2 version and the Snort 3 version.

Procedure

Step 1 Go to **Policies** > **Intrusion** > **Network Analysis Policies**.

Step 2 Click **Create Policy**.

Step 3 Enter the **Name** and **Description**.

Step 4 Select a **Base Policy** and click **Save**.

The new network analysis policy is created with its corresponding **Snort 2 Version** and **Snort 3 Version**.

Modify the Network Analysis Policy

You can modify the network analysis policy to change its name, description, or the base policy.

Procedure

Step 1 Go to **Policies** > **Intrusion** > **Network Analysis Policies**.

Step 2 Click **Edit** to change the name, description, inspection mode, or the base policy.

Note If you edit the network analysis policy name, description, base policy, and inspection mode, the edits are applied to both the Snort 2 and Snort 3 versions. If you want to change the inspection mode for a specific version, then you can do that from within the network analysis policy page for that respective version.

Step 3 Click **Save**.

Search for an Inspector on the Network Analysis Policy Page

On the Snort 3 version of the network analysis policy page, you may need to search for an inspector by entering any relevant text in the search bar.

Procedure

Step 1 Go to **Policies** > **Intrusion** > **Network Analysis Policies**.

Step 2 Go to the **Snort 3 Version** of the network analysis policy.

Step 3 Enter an inspector's name or any relevant text to search for in the **Search** bar.

All the inspectors matching the text you search for are displayed.

For example, if you enter `pop`, then the pop inspector and the binder inspector are shown as matching results on the screen.

Related Topics

Examples of Custom Network Analysis Policy Configuration, on page 21
View the List of Inspectors with Overrides, on page 18
Snort 3 Definitions and Terminologies for Network Analysis Policy, on page 3
Customize the Network Analysis Policy, on page 13
Make Inline Edit for an Inspector to Override Configuration, on page 16

Copy the Inspector Configuration

You can copy the inspector configuration for the Snort 3 version of the network analysis policy according to your requirements.

Procedure

Step 1 Go to **Policies** > **Intrusion** > **Network Analysis Policies**.

Step 2 Go to the **Snort 3 Version** of the network analysis policy.

Step 3 Under **Inspectors**, expand the required inspector for which you want to copy the configuration.

The default configuration is displayed on the left column and the overridden configuration is displayed on the right column under the inspector.

Step 4 Click the **Copy to clipboard** icon to copy the inspector configuration to the clipboard for one or both of the following.

- **Default Configuration** in the left column
- **Overridden Configuration** in the right column

Step 5 Paste the copied inspector configuration to a JSON editor to make any edits you may require.

Related Topics

Customize the Network Analysis Policy, on page 13

Customize the Network Analysis Policy

You can customize the Snort 3 version of the network analysis policy according to your requirements.

Procedure

Step 1 Go to **Policies** > **Intrusion** > **Network Analysis Policies**.

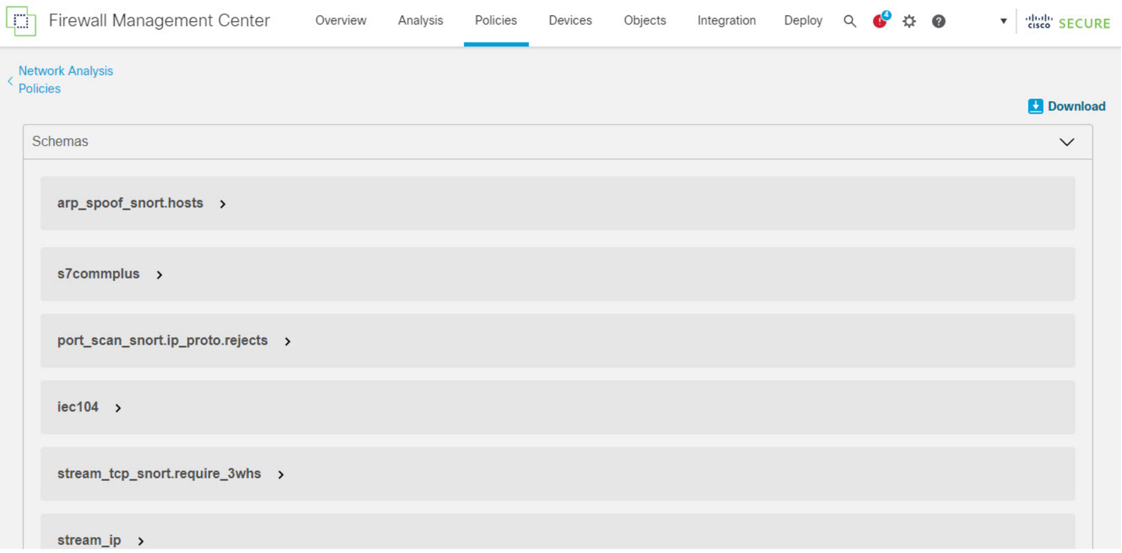
Step 2 Go to the **Snort 3 Version** of the network analysis policy.

Step 3 Click the **Actions** drop-down menu.

The following options are displayed:

- View Schema
- Download Schema / Download Sample File / Template
- Download Full Configuration
- Download Overridden Configuration
- Upload Overridden Configuration

Step 4 Click **View Schema** to open the schema file directly in a browser.



arp_spoof_snort.hosts ›

s7commplus ›

port_scan_snort.ip_proto.rejects ›

iec104 ›

stream_tcp_snort.require_3whs ›

stream_ip ›

Step 5 You can download the schema file, sample file / template, full configuration, or overridden configuration as needed.

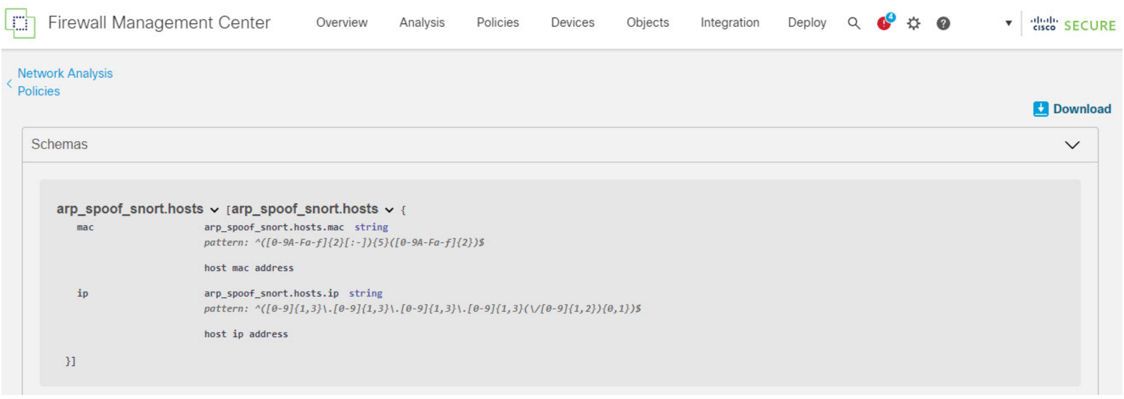
These options provide you an insight about the allowed values, range, and patterns, existing and default inspector configurations, and overridden inspector configurations.

a) Click **Download Schema** to download the schema file.

The schema file validates the content that you upload or download. You can download the schema file and open it using any third-party JSON editor. The schema file helps you to identify what parameters can be configured for inspectors with their corresponding allowed values, range, and accepted patterns to be used.

For example, for the *arp_spoof_snort* inspector, you can configure the hosts. The hosts include the *mac* and *ip* address values. The schema file shows the following accepted pattern for these values.

- **mac** – **`pattern: ^([0-9A-Fa-f]{2}[:-]){5}([0-9A-Fa-f]{2})$`**
- **ip** – **`pattern: ^([0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}(/[0-9]{1,2}){0,1})$`**



```
arp_spoof_snort.hosts ⌄ [arp_spoof_snort.hosts ⌄ {
   mac                 arp_spoof_snort.hosts.mac  string
                       pattern: ^([0-9A-Fa-f]{2}[:-]){5}([0-9A-Fa-f]{2})$

                       host mac address

   ip                  arp_spoof_snort.hosts.ip  string
                       pattern: ^([0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}(\/[0-9]{1,2}){0,1})$

                       host ip address

  }]
```

You must provide the values, range, patterns according to the accepted ones in the schema file to be able to successfully override the inspector configuration, otherwise, you get an error message.

b) Click **Download Sample File / Template** to use a pre-existing template that contains example configurations to help you with configuring the inspectors.

You can refer to the example configurations included in the sample file and make any changes that you may require.

c) Click **Download Full Configuration** to download the entire inspector configurations in a single JSON file.

Instead of expanding the inspectors separately, you can download the full configuration to look out for the information you need. All information regarding the inspector configuration is available in this file.

d) Click **Download Overridden Configuration** to download the inspector configuration that has been overridden.

Step 6 To override the existing configuration, follow the steps.

You can choose to override an inspector configuration using the following ways.

- Make inline edits for an inspector directly on the management center. See the topic **Make Inline Edit for an Inspector to Override Configuration** in the **Getting Started with Network Analysis Policies** chapter of the *Cisco Secure Firewall Management Center Snort 3 Configuration Guide*.
- Continue to follow the current procedure of using the **Actions** drop-down menu to upload the overridden configuration file.

If you chose to make inline edits directly in the management center, then you don't need to follow the current procedure further. Otherwise, you must follow this procedure completely.

a) Under **Inspectors**, expand the required inspector for which you want to override the default configuration.

   The default configuration is displayed on the left column and the overridden configuration is displayed on the right column under the inspector.

   You may need to search for an inspector by entering any relevant text in in the search bar.

b) Click the **Copy to clipboard** icon to copy the default inspector configuration to the clipboard.
c) Create a JSON file and paste the default configuration in it.
d) Keep the inspector configuration that you want to override, and remove all the other configuration and instances from the JSON file.

   You can also use the **Sample File / Template** to understand how to override the default configuration. This is a sample file that includes JSON snippets explaining how you can customize the network analysis policy for Snort 3.

e) Make changes to the inspector configuration as needed.

   Validate the changes and make sure they conform to the schema file. For multiton inspectors, make sure that the binder conditions for all instances are included in the JSON file. See *Multiton Inspectors* in the topic **Custom Network Analysis Policy Creation for Snort 3** in the *Cisco Secure Firewall Management Center Snort 3 Configuration Guide* for more information.

f) If you are copying any further default inspector configurations, append that inspector configuration to the existing file that contains the overridden configuration.

   Note The copied inspector configuration must comply with the JSON standards.

g) Save the overridden configuration file to your system.

Step 7 From the **Actions** drop-down menu, choose Upload Overridden Configuration to upload the JSON file that contains the overridden configuration.

Caution Upload only the changes that you require. You should not upload the entire configuration as it makes the overrides sticky in nature and therefore, any subsequent changes to the default configuration as part of the LSP updates would not be applied.

You can drag and drop a file or click to browse to the JSON file saved in your system that contains the overridden inspector configuration.

- **Merge inspector overrides** – Content in the uploaded file is merged with the existing configuration if there is no common inspector. If there are common inspectors, then the content in the uploaded file (for common inspectors) takes precedence over the previous content, and it replaces the previous configuration for those inspectors.
- **Replace inspector overrides** – Removes all previous overrides and replaces them with the new content in the uploaded file.

  Attention Choosing this option deletes all the previous overrides. Make an informed decision before you override the configuration using this option.

If any error occurs while uploading the overridden inspectors, you see the error in the **Upload Overridden Configuration File** pop-up window. You can also download the file with the error, fix the error, and reupload the file.

Step 8 In the **Upload Overridden Configuration File** pop-up window, click **Import** to upload the overridden inspector configuration.

After you upload the overridden inspector configuration, you will see an orange icon next to the inspector that signifies that it is an overridden inspector.

Also, the **Overridden Configuration** column under the inspector shows the overridden value.

You can also view all the overridden inspectors using the **Show Overrides Only** checkbox adjacent to the Search bar.

Note Make sure that you always download the overridden configuration, open the JSON file, and append any new changes/overrides to the inspector configurations to this file. This action is needed so that you do not lose the old overridden configurations.

Step 9 (Optional) Take a backup of the overridden configuration file on your system before making any new inspector configuration changes.

Tip We recommend that you take the backup from time to time as you override the inspector configuration.

Related Topics

Revert Overridden Configuration to Default Configuration, on page 18
View the List of Inspectors with Overrides, on page 18
Search for an Inspector on the Network Analysis Policy Page, on page 12
Copy the Inspector Configuration , on page 12

Make Inline Edit for an Inspector to Override Configuration

For the Snort 3 version of the network analysis policy, you can make an inline edit for the inspector configuration to override the configuration according to your requirements.

Alternatively, you can also use the **Actions** drop-down menu to upload the overridden configuration file. See Customize the Network Analysis Policy, on page 13 for more information.

Procedure

Step 1 Go to **Policies** > **Intrusion** > **Network Analysis Policies**.

Step 2 Go to the **Snort 3 Version** of the network analysis policy.

Step 3 Under **Inspectors**, expand the required inspector for which you want to override the default setting.

The default configuration is displayed on the left column and the overridden configuration is displayed on the right column under the inspector.

Step 4 Under the **Overridden Configuration** in the right column, click **Edit Inspector** (Pencil) icon to make changes to the inspector configuration.

The Override Configuration pop-up appears where you can make the required edits.

Note

- Make sure that you keep only those settings that you want to override. If you leave a setting with the same value, that field becomes sticky. This means if that setting is changed in the future by Talos, the current value will be retained.
- If you are adding or deleting any custom instance, make sure that you add or delete a binder rule for that instance in the binder inspector as well.

Step 5 Click **OK**.

If there are any errors according to the JSON standards, it shows you an error message.

Step 6 Click **Save** to save the changes.

If the changes conform to the OpenAPI schema specification, the management center allows you to save the configuration, otherwise, the **Error saving overridden configuration** pop-up appears that shows the errors. You can also download the file with the errors.

Related Topics

Customize the Network Analysis Policy, on page 13
Revert Unsaved Changes during Inline Edits , on page 17
Revert Overridden Configuration to Default Configuration, on page 18
Examples of Custom Network Analysis Policy Configuration, on page 21

Revert Unsaved Changes during Inline Edits

While making inline edits to override the configuration for an inspector , you can revert any unsaved changes. Note that this action reverts all unsaved changes to the most recently saved value, but does not revert the configuration to the default configuration for an inspector.

Procedure

Step 1 Go to **Policies** > **Intrusion** > **Network Analysis Policies**.

Step 2 Go to the **Snort 3 Version** of the network analysis policy.

Step 3 Under **Inspectors**, expand the required inspector for which you want to revert the unsaved changes.

The default configuration is displayed in the left column and the overridden configuration is displayed in the right column under the inspector.

Step 4 Under the **Overridden Configuration** on the right column, click the **Cross** (X) icon to revert any unsaved changes for the inspector.

Alternatively, you can click **Cancel** to cancel the changes.

If you do not have any unsaved changes to the inspector configuration, then this option is not visible.

Related Topics

Revert Overridden Configuration to Default Configuration, on page 18
Make Inline Edit for an Inspector to Override Configuration, on page 16

View the List of Inspectors with Overrides

You can view a list of all the overridden inspectors.

Procedure

Step 1 Go to **Policies** > **Intrusion** > **Network Analysis Policies**.

Step 2 Go to the **Snort 3 Version** of the network analysis policy.

Step 3 Check the **Show Overrides Only** checkbox adjacent to the Search bar to view the list of overridden inspectors.

All the overridden inspectors are shown with an orange icon next to their names to help you identify them.

Related Topics

- Search for an Inspector on the Network Analysis Policy Page, on page 12
- Make Inline Edit for an Inspector to Override Configuration, on page 16
- Customize the Network Analysis Policy, on page 13

Revert Overridden Configuration to Default Configuration

You can revert any changes that you made to override the default configuration for an inspector. This action reverts the overridden configuration to the default configuration for an inspector.

Procedure

Step 1 Go to **Policies** > **Intrusion** > **Network Analysis Policies**.

Step 2 Go to the **Snort 3 Version** of the network analysis policy.

Step 3 Under **Inspectors**, expand the required inspector for which you want to revert the overridden configuration.

The overridden inspectors are shown with the orange icon next to their name.

The default configuration is displayed on the left column and the overridden configuration is displayed on the right column under the inspector. Under the **Overridden Configuration** on the right column, click **Revert to default configuration** (back arrow) icon to revert the overridden configuration for the inspector to the default configuration.

If you did not make any changes to the default configuration for the inspector, then this option is disabled.

Step 4 Click **Revert** to confirm the decision.

Step 5 Click **Save** to save the changes.

If you do not want to save the changes, you can click **Cancel** or the **Cross** (X) icon.

Related Topics

- Revert Unsaved Changes during Inline Edits , on page 17
- Customize the Network Analysis Policy, on page 13
- Make Inline Edit for an Inspector to Override Configuration, on page 16

Examples of Custom Network Analysis Policy Configuration, on page 21

Validate Snort 3 Policies

To validate the Snort 3 policies, here is a list of basic information that user can make note of:

- Current version of the management center can manage multiple threat defense versions.
- Current version of management center supports NAP configurations which are not applicable to previous version of threat defense devices.
- Current NAP Policy and validations will work based on the current version support.
- Changes may include content which is not valid for previous versions of threat defenses.
- Policy configuration changes are accepted if they are valid configuration for the current version and which is performed using current Snort 3 binary and NAP schema.
- For previous version threat defenses, validation is performed during deployment using NAP schema and Snort 3 binary for that specific version. If there is any configuration which is not applicable for the given version, user is provided information or warning that we will not deploy the configuration which is not supported on the given version and remaining configuration will get deployed.

In this procedure, when we associate the NAP policy to an Access Control Policy and deploy it on a device, for example any inspector like rate filter configuration is applied to validate the Snort 3 policies.

Procedure

Step 1 **Steps to Override NAP Policy Configuration:** Under **Inspectors** in the **Snort 3 Version** of the network analysis policy, expand the required inspector for which you want to override the default setting.

The default configuration is displayed on the left column and the overridden configuration is displayed on the right column under the inspector.

Step 2 Under the **Overridden Configuration** on the right column, click **Edit Inspector** (Pencil) icon to make changes to any inspector like rate_filter.

The Override Configuration pop-up appears where you can make the required edits to the rate_filter inspector.

Step 3 Click **OK**.

Step 4 Click **Save** to save the changes.

Alternatively, you can also use the **Actions** drop-down menu to upload the overridden configuration file.

Step 5 Click the **Actions** drop-down menu in the **Snort 3 Version** of the network analysis policy.

Step 6 Under **Upload** you can click **Overridden Configuration** to upload the JSON file that contains the overridden configuration.

Caution Upload only the changes that you require. You should not upload the entire configuration as it makes the overrides sticky in nature and therefore, any subsequent changes to the default configuration as part of the LSP updates will not be applied.

You can drag and drop a file or click to browse to the JSON file saved in your system that contains the overridden inspector configuration.

- **Merge inspector overrides** – Content in the uploaded file is merged with the existing configuration if there is no common inspector. If there are common inspectors, then the content in the uploaded file (for common inspectors) takes precedence over the previous content, and it replaces the previous configuration for those inspectors.
- **Replace inspector overrides** – Removes all previous overrides and replaces them with the new content in the uploaded file.

  Attention As choosing this option deletes all the previous overrides, make an informed decision before you override the configuration using this option.

If any error occurs while uploading the overridden inspectors, you see the error on the **Upload Overridden Configuration File** pop-up window. You can also download the file with the error, then fix the error and reupload the file.

Step 7 **Steps to Associate NAP Policy to Access Control Policy:** In the access control policy editor, click **Advanced**, then click **Edit** next to the Network Analysis and Intrusion Policies section.

Step 8 From the **Default Network Analysis Policy** drop-down list, select a default network analysis policy.

If you choose a user-created policy, you can click **Edit** to edit the policy in a new window. You cannot edit system-provided policies.

Step 9 Click **OK**.

Step 10 Click **Save** to save the policy.

Step 11 Alternatively, in the access control policy editor, click **Advanced**, then click **Edit** next to the Network Analysis and Intrusion Policies section.

Step 12 Click **Add Rule**.

Step 13 Configure the rule's conditions by clicking the conditions you want to add.

Step 14 Click **Network Analysis** and choose the **Network Analysis Policy** you want to use to preprocess the traffic matching this rule.

Step 15 Click **Add**.

Step 16 **Deployment:** On the management center menu bar, click **Deploy** and then select **Deployment**.

Step 17 Identify and choose the devices on which you want to deploy configuration changes.

- Search—Search for the device name, type, domain, group, or status in the search box.
- Expand—Click **Expand Arrow** to view device-specific configuration changes to be deployed.

  By selecting the device check box, all the changes for the device, which are listed under the device, are pushed for deployment. However, you can use the **Policy Selection** to select individual policies or configurations to deploy while withholding the remaining changes without deploying them.

  Optionally, use **Show or Hide Policy** to selectively view or hide the associated unmodified policies.

Step 18 Click **Deploy**.

Step 19 If the system identifies errors or warnings in the changes to be deployed, it displays them in the **Validation Messages** window. To view complete details, click the arrow icon before the warnings or errors.

Note It shows a warning that Snort 3 Network analysis policy contains inspectors or attributes that are not valid for this threat defense version, following the invalid settings will be skipped in deployment: Invalid inspectors are : [“rate_filter”] only for devices lower than 7.1 version.

Examples of Custom Network Analysis Policy Configuration

This is a sample file that includes JSON snippets explaining how you can customize the network analysis policy for Snort 3. You can choose to override an inspector configuration using the following ways:

- Make inline edits for an inspector directly on the management center. See Make Inline Edit for an Inspector to Override Configuration, on page 16.
- Use the **Actions** drop-down menu to upload the overridden configuration file. See Customize the Network Analysis Policy, on page 13.

Before you choose any of these options, review all the following details and examples that will help you in defining the network analysis policy overrides successfully. You must read and understand the examples for various scenarios explained here to avoid any risks and errors.

If you choose to override an inspector configuration from the **Actions** drop-down menu, you need to construct a JSON file for the network analysis policy overrides and upload the file.

For overriding an inspector configuration in the network analysis policy, you must upload only the changes that you require. You should not upload the entire configuration because it makes the overrides sticky in nature and therefore, any subsequent changes to the default values or configuration as part of the LSP updates would not be applied.

Here are the examples for various scenarios:

Enabling a Singleton Inspector when the Default State in the Base Policy is Disabled

```
{
  "rate_filter": {
    "enabled": true,
    "type": "singleton",
    "data": []
  }
}
```

Disabling a Singleton Inspector when the Default State in the Base Policy is Enabled

```
{
  "rate_filter": {
    "enabled": false,
    "type": "singleton",
    "data": []
  }
}
```

Enabling a Multiton Inspector when the Default State in the Base Policy is Disabled

```
{
  "ssh": {
    "enabled": true,
    "type": "multiton",
    "instances": []
  }
}
```

Disabling a Multiton Inspector when the Default State in the Base Policy is Enabled

```
{
  "ssh": {
    "enabled": false,
    "type": "multiton",
    "instances": []
  },
  "iec104": {
    "type": "multiton",
    "enabled": false,
    "instances": []
  }
}
```

Overriding the Default Value of Specific Setting(s) for Singleton Inspector

```
{
  "normalizer": {
    "enabled": true,
    "type": "singleton",
    "data": {
      "tcp": {
        "block": true
      },
      "ip6": true
    }
  }
}
```

Overriding Specific Setting(s) of a Default Instance (where Instance Name Matches with Inspector Type) in Multiton Inspector

```
{
  "http_inspect": {
    "enabled": true,
    "type": "multiton",
    "instances": [
      {
        "data": {
          "unzip": false
        },
        "name": "http_inspect"
      }
    ]
  }
}
```

Adding Binder Rule for a Default Instance with Required Changes

Note Default binder rules can't be edited, they are always appended at the end.

```
{
  "binder": {
    "enabled": true,
    "type": "binder",
    "rules": [
      {
        "use": {
          "type": "http_inspect"
        },
        "when": {
          "role": "server",
          "service": "http",
          "dst_nets": "10.1.1.0/24"
        }
      }
    ]
  }
}
```

Adding a New Custom Instance

Note Corresponding binder rule entry must be defined in the binder inspector.

```
{
  "telnet": {
    "enabled": true,
    "type": "multiton",
    "instances": [
      {
        "name": "telnet_my_instance",
        "data": {
          "encrypted_traffic": true
        }
      }
    ]
  },
  "binder": {
    "enabled": true,
    "type": "binder",
    "rules": [
      {
        "when": {
          "role": "any",
          "service": "telnet"
        },
        "use": {
          "type": "telnet",
          "name": "telnet_my_instance"
        }
      }
    ]
  }
}
```

Overriding a Singleton Instance, Multiton Default Instance, and Creating a New Multiton Instance in a Single JSON Override

Example to show the following in a single JSON override:

- Overriding a Singleton instance (**`normalizer`** inspector)
- Overriding a Multiton default instance (**`http_inspect`** inspector)
- Creating a new Multiton instance (**`telnet`** inspector)

```
{
  "normalizer": {
    "enabled": true,
    "type": "singleton",
    "data": {
      "tcp": {
        "block": true
      },
      "ip6": true
    }
  },
  "http_inspect": {
    "enabled": true,
    "type": "multiton",
    "instances": [
      {
        "data": {
          "unzip": false,
          "xff_headers": "x-forwarded-for true-client-ip x-another-forwarding-header"
        },
        "name": "http_inspect"
      }
    ]
  },
  "telnet": {
    "enabled": true,
    "type": "multiton",
    "instances": [
      {
        "name": "telnet_my_instance",
        "data": {
          "encrypted_traffic": true
        }
      }
    ]
  },
  "binder": {
    "enabled": true,
    "type": "binder",
    "rules": [
      {
        "when": {
          "role": "any",
          "service": "telnet"
        },
        "use": {
          "type": "telnet",
          "name": "telnet_my_instance"
        }
      },
      {
        "use": {
          "type": "http_inspect"
        },
        "when": {
          "role": "server",
          "service": "http",
          "dst_nets": "10.1.1.0/24"
        }
      }
    ]
  }
}
```

Note You don't need to give the **`name`** attribute for the default instance in binder rules.

Configuring arp_spoof

Example for configuring **`arp_spoof`**:

Ther **`arp_spoof`** inspector does not have any default configurations for any attributes. This demonstrates the case where you can provide the overrides.

```
{
  "arp_spoof": {
    "type": "singleton",
    "data": {
      "hosts": [
        {
          "ip": "1.1.1.1",
          "mac": "ff:0f:f1:0f:0f:ff"
        },
        {
          "ip": "2.2.2.2",
          "mac": "ff:0f:f2:0f:0f:ff"
        }
      ]
    },
    "enabled": true
  }
}
```

Configuring rate_filter

```
{
  "rate_filter": {
    "data": [
      {
        "apply_to": "[10.1.2.100, 10.1.2.101]",
        "count": 5,
        "gid": 135,
        "new_action": "alert",
        "seconds": 1,
        "sid": 1,
        "timeout": 5,
        "track": "by_src"
      }
    ],
    "enabled": true,
    "type": "singleton"
  }
}
```

Configuring Binder Rules when Multi-Hierarchy Network Analysis Policy is Used

This example illustrates adding a new custom instance in child policy and the way binder rules should be written. Binder rules are defined as a list and therefore, it is important to pick up the rules defined in the parent policy and build the new rules on top of it as rules will not be merged automatically. The binder rules available in child policy are a source of truth in totality.

On the threat defense, the default Cisco Talos policy rules are appended on these user-defined overrides.

Parent Policy:

We have defined a custom instance by the name **`telnet_parent_instance`** and the corresponding binder rule.

```
{
  "telnet": {
    "type": "multiton",
    "instances": [
      {
        "data": {
          "normalize": true,
          "encrypted_traffic": true
        },
        "name": "telnet_parent_instance"
      }
    ],
    "enabled": true
  },
  "binder": {
    "enabled": true,
    "type": "binder",
    "rules": [
      {
        "when": {
          "role": "any",
          "service": "telnet"
        },
        "use": {
          "type": "telnet",
          "name": "telnet_parent_instance"
        }
      }
    ]
  }
}
```

Child Policy:

This network analysis policy has the aforementioned policy as its base policy. We have defined a custom instance by the name **`telnet_child_instance`** and have also defined the binder rules for this instance. The binder rules from parent policy need to be copied here, and then child policy binder rules can be prepended or appended on top of it based on the nature of the rule.

```
{
  "telnet": {
    "type": "multiton",
    "instances": [
      {
        "data": {
          "normalize": true,
          "encrypted_traffic": false
        },
        "name": "telnet_child_instance"
      }
    ],
    "enabled": true
  },
  "binder": {
    "enabled": true,
    "type": "binder",
    "rules": [
      {
        "when": {
          "role": "any",
          "service": "telnet",
          "nets": "10.2.2.0/24"
        },
        "use": {
          "type": "telnet",
          "name": "telnet_child_instance"
        }
      },
      {
        "when": {
          "role": "any",
          "service": "telnet"
        },
        "use": {
          "type": "telnet",
          "name": "telnet_parent_instance"
        }
      }
    ]
  }
}
```

Configuring List Inspector Attribute in General

While changing overrides for any attribute of type list, it is important to pass the full contents rather than partial override. This means if a base policy attributes are defined as:

```
{
  "list-attribute": [
    {
      "entry1": {
        "key1": "value1"
      }
    },
    {
      "entry2": {
        "key2": "value2"
      }
    }
  ]
}
```

If you want to modify **`value1`** to **`value1-new`**, the override payload must look like the following:

Correct Way:

```
{
  "list-attribute": [
    {
      "entry1": {
        "key1": "value1-new"
      }
    },
    {
      "entry2": {
        "key2": "value2"
      }
    }
  ]
}
```

Incorrect Way:

```
{
  "list-attribute": [
    {
      "entry1": {
        "key1": "value1-new"
      }
    }
  ]
}
```

You can understand this configuration by taking the trimmed values of the **`alt_max_command_line_len`** attribute in the **`smtp`** inspector. Suppose the default (base) policy configuration for **`smtp`** inspector is as follows:

```
{
  "smtp": {
    "type": "multiton",
    "instances": [
      {
        "name": "smtp",
        "data": {
          "decompress_zip": false,
          "normalize_cmds": "ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO
           EMAL ESAM ESND ESOM ETRN EVFY EXPN HELO HELP IDENT MAIL
           NOOP ONEX QUEU QUIT RCPT RSET SAML SEND SOML STARTTLS TICK
           TIME TURN TURNME VERB VRFY X-ADAT XADR XAUTH XCIR X-DRCP X-
           ERCP XEXCH50 X-EXCH50 X-EXPS XGEN XLICENSE X-LINK2STATE XQUE
           XSTA XTRN XUSR",
          "ignore_data": false,
          "max_command_line_len": 512,
          "max_header_line_len": 1000,
          "log_rcptto": false,
          "decompress_swf": false,
          "max_response_line_len": 512,
          "b64_decode_depth": -1,
          "max_auth_command_line_len": 1000,
          "log_email_hdrs": false,
          "xlink2state": "alert",
          "binary_data_cmds": "BDAT XEXCH50",
          "auth_cmds": "AUTH XAUTH X-EXPS",
          "log_filename": false,
          "uu_decode_depth": -1,
          "ignore_tls_data": false,
          "data_cmds": "DATA",
          "bitenc_decode_depth": -1,
          "alt_max_command_line_len": [
            {
              "length": 255,
              "command": "ATRN"
            },
            {
              "command": "AUTH",
              "length": 246
            },
            {
              "length": 255,
              "command": "BDAT"
            },
            {
              "length": 246,
              "command": "DATA"
            }
          ],
          "log_mailfrom": false,
          "decompress_pdf": false,
          "normalize": "none",
          "email_hdrs_log_depth": 1464,
          "valid_cmds": "ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO
           EMAL ESAM ESND ESOM ETRN EVFY EXPN HELO HELP IDENT MAIL
           NOOP ONEX QUEU QUIT RCPT RSET SAML SEND SOML STARTTLS TICK
           TIME TURN TURNME VERB VRFY X-ADAT XADR XAUTH XCIR X-DRCP X-
```

```
             ERCP XEXCH50 X-EXCH50 X-EXPS XGEN XLICENSE X-LINK2STATE XQUE
             XSTA XTRN XUSR",
            "qp_decode_depth": -1
          }
        }
      ],
      "enabled": true
    }
  }
```

Now, if you want to add two more objects to the **`alt_max_command_line_len`** list:

```
{
    "length": 246,
    "command": "XEXCH50"
},
{
    "length": 246,
    "command": "X-EXPS"
}
```

Then the custom network analysis policy override JSON would look like the following:

```
{
  "smtp": {
    "type": "multiton",
    "instances": [
      {
        "name": "smtp",
        "data": {
          "alt_max_command_line_len": [
            {
              "length": 255,
              "command": "ATRN"
            },
            {
              "command": "AUTH",
              "length": 246
            },
            {
              "length": 255,
              "command": "BDAT"
            },
            {
              "length": 246,
              "command": "DATA"
            },
            {
              "length": 246,
              "command": "XEXCH50"
            },
            {
              "length": 246,
              "command": "X-EXPS"
            }
          ]
        }
      }
    ],
    "enabled": true
  }
}
```

Configuring Overrides when Multi-Hierarchy Network Analysis Policy is used in Multiton Inspector

This example illustrates overriding attributes in child policy and how the merged configuration will be used in the child policy for any instance. Any overrides defined in the child policy will be merged with the parent policy. Thus, if attribute1 and attribute2 are overridden in parent policy and attribute2 and attribute3 are overridden in the child policy, the merged configurations are for child policy. This means that attribute1 (defined in parent policy), attribute2 (defined in child policy), and attribute3 (defined in child policy) will be configured on the device.

Parent Policy:

Here we have defined a custom instance by the name **`telnet_parent_instance`** and overridden 2 attributes namely, **`normalize`** and **`encrypted_traffic`** in the custom instance.

```
{
  "telnet": {
    "type": "multiton",
    "instances": [
      {
        "data": {
          "normalize": true,
          "encrypted_traffic": false
        },
        "name": "telnet_parent_instance"
      }
    ],
    "enabled": true
  },
  "binder": {
    "enabled": true,
    "type": "binder",
    "rules": [
      {
        "when": {
          "role": "any",
          "service": "telnet"
        },
        "use": {
          "type": "telnet",
          "name": "telnet_parent_instance"
        }
      }
    ]
  }
}
```

Child Policy:

This network analysis policy has the aforementioned policy as its base policy. We have overridden attribute **`encrypted_traffic`** from parent policy and also overridden new attribute **`ayt_attack_thresh`**.

```
{
  "telnet": {
    "type": "multiton",
    "instances": [
      {
        "data": {
          "encrypted_traffic": true,
          "ayt_attack_thresh": 1
        },
        "name": "telnet_parent_instance"
      }
    ],
    "enabled": true
  }
}
```

With the above policy JSON, when you deploy the network analysis policy the following merged JSON will be configured on the device.

```
{
  "telnet": {
    "type": "multiton",
    "instances": [
      {
        "data": {
          "normalize": true,
          "encrypted_traffic": true,
          "ayt_attack_thresh": 1
        },
        "name": "telnet_parent_instance"
      }
    ],
    "enabled": true
  },
  "binder": {
    "enabled": true,
    "type": "binder",
    "rules": [
      {
        "when": {
          "role": "any",
          "service": "telnet"
        },
        "use": {
          "type": "telnet",
          "name": "telnet_parent_instance"
        }
      }
    ]
  }
}
```

This example illustrates details for the custom network analysis policy. The same behavior is also exhibited in the default instance. Also, a similar merging would be done for Singleton inspectors.

Removing all the Inspector Overrides for the Network Analysis Policy:

Whenever you want to remove all the overrides for a specific network analysis policy, you can upload an empty JSON. While uploading the overrides, choose the option **Replace inspector overrides**.

```
{
}
```

Related Topics

Snort 3 Definitions and Terminologies for Network Analysis Policy, on page 3

Network Analysis Policy Mapping, on page 10

Custom Network Analysis Policy Creation for Snort 3, on page 5

Search for an Inspector on the Network Analysis Policy Page, on page 12

Copy the Inspector Configuration , on page 12

Customize the Network Analysis Policy, on page 13

View the List of Inspectors with Overrides, on page 18

Network Analysis Policy Settings and Cached Changes

When you create a new network analysis policy, it has the same settings as its base policy.

When tailoring a network analysis policy, especially when disabling inspectors, keep in mind that some inspectors and intrusion rules require that traffic first be decoded or preprocessed in a certain way. If you disable a required inspector, the system automatically uses it with its current settings, although the inspector remains disabled in the network analysis policy web interface.

Note Because preprocessing and intrusion inspection are so closely related, the network analysis and intrusion policies examining a single packet **must** complement each other. Tailoring preprocessing, especially using multiple custom network analysis policies, is an **advanced** task.

The system caches one network analysis policy per user. While editing a network analysis policy, if you select any menu or other path to another page, your changes stay in the system cache even if you leave the page.

Custom Rules in Snort 3

You can create a custom intrusion rule by importing a local rule file. The rule file can either have a `.txt` or `.rules` extension. The system saves the custom rule in the local rule category, regardless of the method you used to create it. A custom rule must belong to a rule group. However, a custom rule can be a part of two or more groups as well.

When you create a custom intrusion rule, the system assigns it a unique rule number, which has the format `GID:SID:Rev`. The elements of this number are:

- **GID**—Generator ID. For custom rules, it is not necessary to specify the GID. The system automatically generates the GID based on whether you are in the Global domain or a sub-domain while uploading the rules. For all standard text rules, this value is 2000 for a Global domain.
- **SID**—Snort ID. Indicates whether the rule is a local rule of a system rule. When you create a new rule, assign a unique SID to the rule.

  SID numbers for local rules start at 1000000, and the SID for each new local rule is incremented by one.
- **Rev**—The revision number. For a new rule, the revision number is one. Each time you modify a custom rule the revision number should be incremented by one.

In a custom standard text rule, you set the rule header settings and the rule keywords and arguments. You can use the rule header settings to focus the rule to only match traffic using a specific protocol and traveling to or from specific IP addresses or ports.

Note Snort 3 custom rules cannot be edited. Ensure custom rules have a valid classification message for `classtype` within the rule text. If you import a rule without a classification or wrong classification, then delete and recreate the rule.

Sensitive Data Detection in Snort 3

Sensitive data such as social security numbers, credit card numbers, emails, and so on may be leaked onto the internet, intentionally or accidentally. Sensitive data detection is used to detect and generate events on possible sensitive data leakage. Events are generated only if there is a transfer of significant amount of Personally Identifiable Information (PII) data. Sensitive data detection can mask PII in the output of events.

sd_pattern Option

Use the `sd_pattern` IPS option to detect and filter PII. This information includes credit card numbers, U.S. Social Security numbers, phone numbers, and email addresses. A regular expression (regex) syntax is available for defining your own PII.

The sd_pattern option has the following settings:

- Pattern—An implicit, required setting that specifies the regular expression to look for in the PDU. The regex must be written in PCRE syntax.
- Threshold—An explicit, optional setting that specifies the number of matches in the PDU required to generate an event.

  The `sd_pattern` as IPS rule option is available in Snort with no requirements for additional inspectors. The rule option’s syntax is:

  ```
  sd_pattern: "<pattern>"[, threshold <count>];
  ```

  For example:

  ```
  sd_pattern:"credit_card", threshold 2;
  ```

Built-in Patterns

There are five built-in patterns for sensitive data. To use the built-in patterns in the "pattern" setting, you must specify the name of the PII type that needs to be matched and the necessary regex is substituted for it. The PII name and regex mappings or patterns are described as follows:

- credit_card—

  ```
  \d{4}\D?\d{4}\D?\d{2}\D?\d{2}\D?\d{3,4}
  ```

- us_social—

  ```
  [0-8]\d{2}-\d{2}-\d{4}
  ```

- us_social_nodashes—

  ```
  [0-8]\d{8}
  ```

- Email—

  ```
  [a-zA-Z0-9!#$%&'*+\/=?^_`{|}~-]+(?:\.[a-zA-Z0-9!#$%&'*+\/=?^_`{|}~-]+)*@(?:[a-zA-Z0-9]
  (?:[a-zA-Z0-9-]*[a-zA-Z0-9])?\.)+[a-zA-Z0-9](?:[a-zA-Z0-9-]*[a-zA-Z0-9])?
  ```

- us_phone—

  ```
  (?:\+?1[-\.\s]?)?\(?([2-9][0-8]\d)\)?[-\.\s]([2-9]\d{2})[-\.\s](\d{4})
  ```

PII Name	Pattern
credit_card	\d{4}\D?\d{4}\D?\d{2}\D?\d{2}\D?\d{3,4}

PII Name	Pattern
us_social	`[0-8]\d{2}-\d{2}-\d{4}`
us_social_nodashes	`[0-8]\d{8}`
Email	`[a-zA-Z0-9!#$%&'*+\/=?^_`{\|}~-]+(?:\.[a-zA-Z0-9!#$%&'*+\/=?^_`{|}~-]+)*@(?:[a-zA-Z0-9](?:[a-zA-Z0-9-]*[a-zA-Z0-9])?\.)+[a-zA-Z0-9](?:[a-zA-Z0-9-]*[a-zA-Z0-9])?`
us_phone	`(?:\+?1[-\.\s]?)?\(?([2-9][0-8]\d)\)?[-\.\s]([2-9]\d{2})[-\.\s](\d{4})`

Masking for data matching these patterns only work with system-provided rules or built-in patterns for Credit Cards, U.S. Social Security numbers, emails, and U.S. phone numbers. Masking does not work for custom rules or user-defined PII patterns. Rules are available in the Lightweight Security Package (LSP) for sensitive data, gid:13. By default, they are not enabled in any system-provided policy.

The sensitive data rules in LSP cover all built-in patterns and have the following threshold values:

- credit_card: 2
- us_social: 2
- us_social_nodashes: 20
- email: 20
- us_phone: 20

You can use the sd_pattern option to create custom rules and modify existing rules. To do this, use the Snort 3 intrusion policy interface.

An example of a rule with sd_pattern with a custom pattern and threshold:

alert tcp (sid: 1000000001; sd_pattern:"[\w-\.]+@([\w-]+\.)+[\w-]{2,4}",threshold 4; msg: "email, threshold 4")

Examples

An example of custom rules using sensitive data detection:

Rule with built-in pattern:

```
alert tcp (
        msg:"SENSITIVE-DATA Email";
        flow:only_stream;
        pkt_data;
        sd_pattern:"email", threshold 5;
        service:http, smtp, ftp-data, imap, pop3;
        gid:2000;
        sid:1000001;
)
```

Rule with custom pattern

```
alert tcp (
        msg:"SENSITIVE-DATA US phone numbers";
        flow:only_stream;
        file_data;
        sd_pattern:"+?3?8?(0[\s\.-]\d{2}[\s\.-]\d{3}[\s\.-]\d{2}[\s\.-]\d{2})", threshold
2;
        service:http, smtp, ftp-data, imap, pop3;
        gid:2000;
        sid:1000002;
)
```

Here are some more examples of complete Snort IPS rules with built-in sensitive data patterns:

- alert tcp (sid:1; msg:"Credit Card"; sd_pattern:"credit_card", threshold 2;)
- alert tcp (sid:2; msg:"US Social Number"; sd_pattern:"us_social", threshold 2;)
- alert tcp (sid:3; msg:"US Social Number No Dashes"; sd_pattern:"us_social_nodashes", threshold 2;)
- alert tcp (sid:4; msg:"US Phone Number"; sd_pattern:"us_phone", threshold 2;)
- alert tcp (sid:5; msg:"Email"; sd_pattern:"email", threshold 2;)

Disabling data masking is not supported in the Secure Firewall Management Center and Secure Firewall Device Manager.

Overview of Encrypted Visibility Engine

The encrypted visibility engine (EVE) is used to provide more visibility into the encrypted sessions without the need to decrypt them. These insights into encrypted sessions are obtained by Cisco's open-source library that is packaged in Cisco's vulnerability database (VDB). The library fingerprints and analyzes incoming encrypted sessions and matches it against a set of known fingerprints. This database of known fingerprints is also available in the Cisco VDB.

Note The encrypted visibility engine feature is supported only on management center-managed devices running Snort 3. This feature is not supported on Snort 2 devices, device manager-managed devices, or CDO.

Some of the important features of EVE are the following:

- You can take access control policy actions on the traffic using information derived from EVE.
- The VDB included in Cisco Secure Firewall has the ability to assign applications to some processes detected by EVE with a high confidence value. Alternatively, you can create custom application detectors to:
    - Map EVE-detected processes to new user-defined applications.
    - Override the built-in value of process confidence that is used to assign applications to EVE-detected processes.

      See the **Configuring Custom Application Detectors** and **Specifying EVE Process Assignments** sections in the **Application Detection** chapter of the Cisco Secure Firewall Management Center Device Configuration Guide.
- EVE can detect the operating system type and version of the client that created a Client Hello packet in the encrypted traffic.
- EVE supports fingerprinting and analysis of Quick UDP Internet Connections (QUIC) traffic too. The server name from the Client Hello packet is displayed in the URL field of the **Connection Events** page.

Attention To use EVE on management center, you must have a valid IPS license on your device. In the absence of a IPS license, the policy displays a warning and deployment is not allowed.

Note EVE can detect the operating system type and version of SSL sessions. Normal usage of the operating system, such as running applications and package management software, can trigger OS detection. To view client OS detection, in addition to enabling the EVE toggle button, you must enable **Hosts** under **Policies** > **Network Discovery**. To view a list of possible operating systems on the host IP address, click **Analysis** > **Hosts** > **Network Map**, and then choose the required host.

Related Links

Configure EVE, on page 37

How EVE Works

The encrypted visibility engine (EVE) inspects the Client Hello portion of the TLS handshake to identify client processes. The Client Hello is the initial data packet that is sent to the server. This gives a good indication of the client process on the host. This fingerprint, combined with other data such as destination IP address, provides the basis for EVE's application identification. By identifying specific application fingerprints in the TLS session establishment, the system can identify the client process and take appropriate action (allow/block).

EVE can identify over 5,000 client processes. The system maps a number of these processes to client applications for use as criteria in access control rules. This gives the system the ability to identify and control these applications without enabling TLS decryption. By using fingerprints of known malicious processes, EVE technology can also be used to identify and block encrypted malicious traffic without outbound decryption.

Through machine learning (ML) technology, Cisco processes over one billion TLS fingerprints and over 10000 malware samples daily to create and update EVE fingerprints. These updates are then delivered to customers using Cisco's Vulnerability Database (VDB) packages.

Indications of Compromise Events

The host's Indications of Compromise (IoC) events for encrypted visibility engine detection allows you to check connection events with a very high malware confidence level, as reported by EVE. IoC events are triggered for encrypted sessions generated from a host using a malicious client. You can view information, such as IP address, MAC address, and OS information of the malicious host, and the timestamp of the suspicious activity.

A session with Encrypted Visibility Threat Confidence score 'Very High' as seen in connection events genreates an IoC event. You must enable **Hosts** from **Policies** > **Network Discovery**. In the management center, you can view the IoC event existence from:

- **Analysis** > **Indications of Compromise**.
- **Analysis** > **Network Map** > **Indications of Compromise** > Choose the host that must be checked.

  You can view the process information of the session that generated the IoC from:

Analysis > **Connection Events** > **Table View of Connection Events** > **IoC** column. Note that you must manually select the Encrypted Visbility fields and IoC field.

QUIC Fingerprinting in EVE

Snort can identify client applications in Quick UDP Internet Connections (QUIC sessions) based on EVE. QUIC fingerprinting can:

- Detect applications over QUIC without enabling decryption.
- Identify malware without enabling decryption.
- Detect service applications. You can assign access control rules based on the service detected over the QUIC protocol.

Configure EVE

Procedure

Step 1 Choose **Policies** > **Access Control**.

Step 2 Click **Edit** (✎) next to the access control policy you want to edit.

Step 3 Choose **Advanced Settings** from the **More** drop-down arrow at the end of the packet flow line.

Step 4 Click **Edit** (✎) next to **Encrypted Visibility Engine**.

Step 5 In the **Encrypted Visibility Engine** page, enable the **Encrypted Visibility Engine (EVE)** toggle button.

Step 6 **Use EVE for Application Detection**—This toggle button is enabled by default, which means that EVE is allowed to assign client applications to processes.

EVE's fingerprint information is added in the **Encrypted Visibility Fingerprint** column header of the connection events or unified events. For further analysis of the EVE data collected, you can right-click the fingerprint information to open a dropdown menu. In the menu, click **View Encrypted Visibility Engine Process Analysis** to go to appid.cisco.com and view details, such as the fingerprint, VDB version, and so on. Different rows with the same fingerprint string and potential process names associated with them and their prevalence are displayed. Prevalence indicates the frequency of a process associated with a particular fingerprint in the data collection system. You can choose the process names and click **Submit Request** to give feedback about any discrepancy in EVE's process detection. For example, you can submit requests if the process name that is detected does not match with the traffic that is being sent or if the process name is not detected at all for a particular fingerprint.

If you disable the **Use EVE for Application Detection** toggle button:

- AppID-identified clients are assigned to processes and you can see the EVE process and score, but there is no mapping of EVE-detected processes to applications and no action is taken. You can see the details of the events under **Connection Events** or **Unified Events**. To see the difference in connection events (with and without application assignment), see the **Client Application** column header.
- The **Encrypted Visibility Fingerprint** field in the connection events or unified events is empty.

Step 7 Enable the **Block Traffic Based on EVE Score** toggle button to block traffic based on EVE's threat confidence score. Any incoming traffic that is a potential threat is blocked by default.

The default block threshold is 99 percent, which means:

- If EVE detects the traffic to be malware with 99 percent confidence or more, the traffic is blocked.
- If EVE detects the traffic to be malware with less than 99 percent confidence, EVE takes no action.

Note If EVE has blocked the traffic, in the **Connection Events** page, the **Reason** column header displays **Encrypted Visiblity Block**.

Step 8 Use the slider to adjust the threshold for blocking based on EVE's threat confidence, which ranges from **Very Low** to **Very High**.

Step 9 For further granular control, enable the **Advanced Mode** toggle button. Now, you can assign a specific EVE Threat Confidence Score for blocking traffic. The default block threshold is 99 percent.

Caution We recommend that you do not set a threshold below 50 percent to ensure optimal performance.

Step 10 Click **OK**.

Step 11 Click **Save**.

What to do next

Deploy configuration changes.

View EVE Events

After enabling the **Encrypted Visibility Engine** and deploying your access control policy, you can start sending live traffic through your system. You can view the logged connection events in the **Connection Events** page. To access the connection events, in the management center:

Procedure

Step 1 Click **Analysis** > **Connections** > **Events**.

Step 2 Click the **Table View of Connection Events** tab.

You can also view the connection event fields in the **Unified Events** viewer, which is under the **Analysis** menu.

Encrypted Visibility Engine can identify the client process that initiated a connection, the OS on the client, and if the process contains malware or not.

Step 3 In the **Connection Events** page, view the following columns that are added for Encrypted Visibility Engine. Note that you must explicitly enable the mentioned columns.

- Encrypted Visibility Process Name
- Encrypted Visibility Process Confidence Score
- Encrypted Visibility Threat Confidence

- Encrypted Visibility Threat Confidence Score
- Detection Type

For information about these fields, see the section **Connection and Security Intelligence Event Fields** in the **Connection and Security-Related Connection Events** chapter of the Cisco Secure Firewall Management Center Administration Guide.

Note In the **Connection Events** page, if processes are assigned applications, the **Detection Type** column displays **Encrypted Visibility Engine** indicating that the client application was identified by EVE. Without application assignments to process names, the **Detection Type** column displays **AppID** indicating that the engine that identified the client application was AppID.

View EVE Dashboard

You can view the EVE analysis information in two dashboards. To access the dashboards:

Procedure

Step 1 Under **Overview** > **Dashboards**, click **Dashboard**.

Step 2 In the **Summary Dashboard** window, click the **switch dashboard** link and choose **Application Statistics** from the dropdown box.

Step 3 Choose the **Encrypted Visibility Engine** tab to view the following two dashboards:

- **Top Encrypted Visibility Engine Discovered Processes**—Displays the top TLS process names being used in your network and the connection count. You can click the process name in the table to see the filtered view of the **Connection Events** page, which is filtered by the process name.
- **Connections by Encrypted Visibility Engine Threat Confidence**—Displays the connections by the confidence levels (Very High, Very Low, and so on). You can click the Threat confidence level in the table to see the filtered view of the **Connection Events** page, which is filtered by the confidence level.

About Elephant Flow Detection and Remediation

You can use the elephant flow detection feature to detect and remediate elephant flows. The following remediation actions can be applied:

- **Bypass elephant flow**–You can configure elephant flow to bypass Snort inspection. If this is configured, Snort does not receive any packet from that flow.
- **Throttle elephant flow**–You can apply rate-limit to the flow and continue to inspect flows. The flow rate is calculated dynamically and 10% of the flow rate is reduced. Snort sends the verdict (QoS flow with 10% less flow rate) to the firewall engine. If you choose to bypass all applications including unidentified applications, you cannot configure the throttle action (rate-limit) for any flow.

Note For the elephant flow detection to work, Snort 3 must be the detection engine.

Elephant Flow Upgrade from Intelligent Application Bypass

Intelligent Application Bypass (IAB) is deprecated from version 7.2.0 onwards for Snort 3 devices.

For devices running 7.2.0 or later, you must configure elephant flow settings under the **Elephant Flow Settings** section in the AC policy (Advanced settings tab).

Post-upgrade to 7.2.0 (or later), if you are using a Snort 3 device, the elephant flow configuration settings will be picked and deployed from the **Elephant Flow Settings** section and not from the **Intelligent Application Bypass Settings** section, so if you have not migrated to Elephant Flow configuration settings, your device will lose the elephant flow configuration upon the next deployment.

The following table shows the IAB or elephant flow configurations that can be applied to version 7.2.0 or later and to version 7.1.0 or earlier that are running Snort 3 or Snort 2 engines.

Management Center	Threat Defense	Elephant Flow or IAB Configuration
Management Center 7.0 or 7.1	Snort 2 device	Configuration from IAB is applicable.
	Snort 3 device	Configuration from IAB is applicable.
Management Center 7.2.0	Snort 2 device	Configuration from IAB is applicable.
	Snort 3 device (7.1.0 and earlier)	Configuration from IAB is applicable.
	Snort 3 device (7.2.0 and later)	Configuration from Elephant Flow is applicable.

Configure Elephant Flow

You can configure elephant flow to take actions on elephant flows, which helps resolve issues, such as system duress, high CPU utilization, packet drops, and so on.

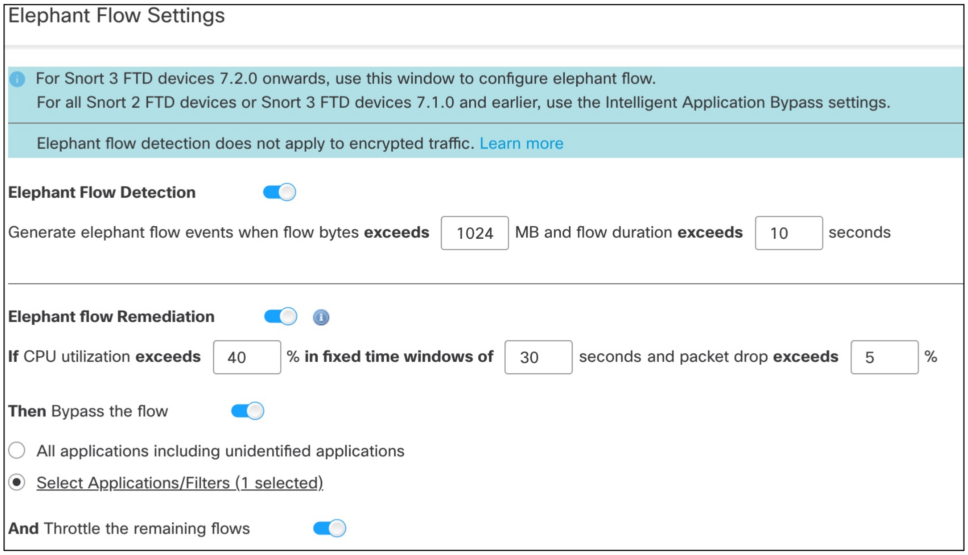
Attention Elephant flow detection is not applicable for prefiltered, trusted, or fast-forwarded flows, which do not process through Snort. As elephant flows are detected by Snort, elephant flow detection is not applicable for encrypted traffic.

Procedure

Step 1 In the access control policy editor, click **Advanced Settings** from the **More** drop-down arrow at the end of the packet flow line. Then, click **Edit** (✎) next to **Elephant Flow Settings**.

If **View** (👁) appears instead, settings are inherited from an ancestor policy, or you do not have permission to modify the settings. If the configuration is unlocked, uncheck **Inherit from base policy** to enable editing.

Figure 1: Configure Elephant Flow Detection



Elephant Flow Settings

For Snort 3 FTD devices 7.2.0 onwards, use this window to configure elephant flow.
For all Snort 2 FTD devices or Snort 3 FTD devices 7.1.0 and earlier, use the Intelligent Application Bypass settings.

Elephant flow detection does not apply to encrypted traffic. Learn more

Elephant Flow Detection

Generate elephant flow events when flow bytes **exceeds** 1024 MB and flow duration **exceeds** 10 seconds

Elephant flow Remediation

If CPU utilization **exceeds** 40 % **in fixed time windows of** 30 seconds and packet drop **exceeds** 5 %

Then Bypass the flow

○ All applications including unidentified applications

● Select Applications/Filters (1 selected)

And Throttle the remaining flows

Step 2 The **Elephant Flow Detection** toggle button is enabled by default. You can configure the values for flow bytes and flow duration. When they exceed your configured values, elephant flow events are generated.

Step 3 To remediate elephant flows, enable the **Elephant Flow Remediation** toggle button.

Step 4 To set the criteria for remediation of the elephant flow, configure the values for CPU utilization %, duration of fixed time windows, and packet drop %.

Step 5 You can perform the following actions for elephant flow remediation when it meets the configured criteria:

a. **Bypass the flow**—Enable this button to bypass Snort inspection for selected applications or filters. Choose from:

   - **All applications including unidentified applications**—Select this option to bypass all the application traffic. If you configure this option, you cannot configure the throttle action (rate-limit) for any flow.

   - **Select Applications/Filters**—Select this option to select the applications or filters whose traffic you want to bypass; see the topic **Configuring Application Conditions and Filters** in the **Access Control Rules** chapter in the Cisco Secure Firewall Management Center Device Configuration Guide.

b. **Throttle the flow**—Enable this button to apply rate-limit to the flow and continue to inspect flows. Note that you can select the applications or filters to bypass Snort inspection and throttle the remaining flows.

Note Automatic removal of throttle from a throttled elephant flow occurs when the system is out of duress, that is, the percentage of Snort packet drops is lesser than your configured threshold. Consequently, rate limiting is also removed.

You can also manually remove throttling from a throttled elephant flow, using the following threat defense commands:

- **clear efd-throttle <5-tuple/all> bypass**—This command removes throttling from the throttled elephant flow and bypasses Snort inspection.
- **clear efd-throttle <5-tuple/all>**—This command removes throttling from the throttled elephant flow and Snort inspection continues. Elephant flow remediation is skipped after using this command.

  For more information about these commands, see the Cisco Secure Firewall Threat Defense Command Reference.

Note Taking action on elephant flows (bypass and throttle the flow) is not supported on Cisco Firepower 2100 series devices.

Step 6 In the **Remediation Exemption Rule** section, click **Add Rule** to configure L4 access control list (ACL) rules for flows that must be exempted from remediation.

Step 7 In the **Add Rule** window, use the **Networks** tab to add the network details, that is the source network and the destination network. Use the **Ports** tab to add the source port and the destination port.

If an elephant flow is detected and it matches the rules that are defined, an event is generated with the reason as **Elephant Flow Exempted** in the **Reason** column header of **Connection Events**.

Step 8 In the **Remediation Exemption Rule** section, you can view the flows that are exempt from the remediation action.

Step 9 Click **OK** to save the elephant flow settings.

Step 10 Click **Save** to save the policy.

What to do next

Deploy configuration changes.

After configuring your elephant flow settings, monitor your connection events to see if any flows are detected, bypassed, or throttled. You can view this in the **Reason** field of your connection event. The three reasons for elephant flow connections are:

- Elephant Flow
- Elephant Flow Throttled
- Elephant Flow Trusted

Attention Enabling elephant flow detection alone does not cause generation of connection events for elephant flows. If a connection event is already logged for another reason and the flow is also an elephant flow, then the **Reason** field contains this information. However, to ensure that you are logging all elephant flows, you must enable connection logging in the applicable access control rules.

Refer to Cisco Secure Firewall Elephant Flow Detection for more information.