



Defining Signatures

This chapter explains how to create signature definition policies and how to configure signatures. It contains the following sections:

- [Understanding Security Policies, page 10-1](#)
- [Configuring Signature Definition Policies, page 10-8](#)
- [sig0 Pane, page 10-10](#)
- [Understanding Signatures, page 10-1](#)
- [MySDN, page 10-11](#)
- [Configuring Signatures, page 10-12](#)
- [Configuring Signature Variables, page 10-38](#)
- [Configuring Miscellaneous Settings, page 10-40](#)

Understanding Security Policies



Note

You must be administrator or operator to add, clone, or delete signature policies.

You can create multiple security policies and apply them to individual virtual sensors. A security policy is made up of a signature definition policy, an event action rules policy, and an anomaly detection policy. Cisco IPS contains a default signature definition policy called sig0, a default event action rules policy called rules0, and a default anomaly detection policy called ad0. You can assign the default policies to a virtual sensor or you can create new policies. The use of multiple security policies lets you create security policies based on different requirements and then apply these customized policies per VLAN or physical interface.

Understanding Signatures

Attacks or other misuses of network resources can be defined as network intrusions. Sensors that use a signature-based technology can detect network intrusions. A signature is a set of rules that your sensor uses to detect typical intrusive activity, such as DoS attacks. As sensors scan network packets, they use signatures to detect known attacks and respond with actions that you define.

The sensor compares the list of signatures with network activity. When a match is found, the sensor takes an action, such as logging the event or sending an alert. Sensors let you modify existing signatures and define new ones.

Signature-based intrusion detection can produce false positives because certain normal network activity can be misinterpreted as malicious activity. For example, some network applications or operating systems may send out numerous ICMP messages, which a signature-based detection system might interpret as an attempt by an attacker to map out a network segment. You can minimize false positives by tuning your signatures.

To configure a sensor to monitor network traffic for a particular signature, you must enable the signature. By default, the most critical signatures are enabled when you install the signature update. When an attack is detected that matches an enabled signature, the sensor generates an alert, which is stored in the Event Store of the sensor. The alerts, as well as other events, may be retrieved from the Event Store by web-based clients. By default the sensor logs all Informational alerts or higher.

Some signatures have subsignatures, that is, the signature is divided into subcategories. When you configure a subsignature, changes made to the parameters of one subsignature apply only to that subsignature. For example, if you edit signature 3050 subsignature 1 and change the severity, the severity change applies to only subsignature 1 and not to 3050 2, 3050 3, and 3050 4.

The Cisco IPS contains over 10,000 built-in default signatures. You cannot rename or delete signatures from the list of built-in signatures, but you can retire signatures to remove them from the sensing engine. You can later activate retired signatures; however, this process requires the sensing engines to rebuild their configuration, which takes time and could delay the processing of traffic. You can tune built-in signatures by adjusting several signature parameters. Built-in signatures that have been modified are called tuned signatures.


Note

We recommend that you retire any signatures that you are not using. This improves sensor performance.

You can create signatures, which are called custom signatures. Custom signature IDs begin at 60000. You can configure them for several things, such as matching of strings on UDP connections, tracking of network floods, and scans. Each signature is created using a signature engine specifically designed for the type of traffic being monitored.

Event Actions

The Cisco IPS supports the following event actions. Most of the event actions belong to each signature engine unless they are not appropriate for that particular engine.

Alert and Log Actions

- Product Alert—Writes the event to the Event Store as an alert.


Note

The Product Alert action is not automatic when you enable alerts for a signature. To have an alert created in the Event Store, you must select Product Alert. If you add a second action, you must include Product Alert if you want an alert sent to the Event Store. Also, every time you configure the event actions, a new list is created and it replaces the old list. Make sure you include all the event actions you need for each signature.



Note There are other event actions that force a Product Alert. These actions use Product Alert as the vehicle for performing the action. Even if Product Alert is not selected or is filtered, the alert is still produced. The actions are the following: Produce Verbose Alert, Request SNMP Trap, Log Attacker Packets, Log Victim Packets, and Log Pair Packets.



Note A Produce Alert event action is added for an event when global correlation has increased the risk rating of an event, and has added either the Deny Packet Inline or Deny Attacker Inline event action.

- Produce Verbose Alert—Includes an encoded dump of the offending packet in the alert. This action causes an alert to be written to the Event Store, even if Product Alert is not selected.
- Log Attacker Packets—Starts IP logging on packets that contain the attacker address and sends an alert. This action causes an alert to be written to the Event Store, even if Product Alert is not selected.
- Log Victim Packets—Starts IP logging on packets that contain the victim address and sends an alert. This action causes an alert to be written to the Event Store, even if Product Alert is not selected.
- Log Pair Packets—Starts IP logging on packets that contain the attacker/victim address pair. This action causes an alert to be written to the Event Store, even if Product Alert is not selected.
- Request SNMP Trap—Sends a request to the Notification Application component of the sensor to perform SNMP notification. This action causes an alert to be written to the Event Store, even if Product Alert is not selected. You must have SNMP configured on the sensor to implement this action.

Deny Actions

- Deny Packet Inline (inline only)—Terminates the packet.



Note You cannot delete the event action override for Deny Packet Inline because it is protected. If you do not want to use that override, disable it.

- Deny Connection Inline (inline only)—Terminates the current packet and future packets on this TCP flow.
- Deny Attacker Victim Pair Inline (inline only)—Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time.



Note For deny actions, to set the specified period of time and maximum number of denied attackers, choose **Configuration > sensor_name > Policies > Event Action Rules > rules0 > General Settings**.

- Deny Attacker Service Pair Inline (inline only)—Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
- Deny Attacker Inline (inline only)—Terminates the current packet and future packets from this attacker address for a specified period of time.

- The sensor maintains a list of attackers being denied by the system. To remove an entry from the denied attacker list, you can view the list of attackers and clear the entire list, or you can wait for the timer to expire. The timer is a sliding timer for each entry. Therefore, if attacker A is being denied, but issues another attack, the timer for attacker A is reset and attacker A remains in the denied attacker list until the timer expires. If the denied attacker list is at capacity and cannot add a new entry, the packet is still denied.



Note This is the most severe of the deny actions. It denies current and future packets from a single attacker address. To clear all denied attacker entries, choose **Configuration > sensor_name > Sensor Management > Time-Based Actions > Denied Attackers > Clear List**, which permits the addresses back on the network.

- Modify Packet Inline (inline only)—Modifies packet data to remove ambiguity about what the end point might do with the packet.



Note You cannot use Modify Packet Inline as an action when adding event action filters or overrides.

Other Actions

- Request Block Connection—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action.



Note Connection blocks and network blocks are not supported on adaptive security appliances. Adaptive security appliances only support host blocks with additional connection information.



Note IPv6 does not support Request Block Connection.

- Request Block Host—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action.



Note IPv6 does not support Request Block Host.



Note For block actions, to set the duration of the block, choose **Configuration > sensor_name > Policies > Event Action Rules > rules0 > General Settings**.

- Request Rate Limit—Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action.



Note Request Rate Limit applies to a select set of signatures.



Note IPv6 does not support Request Rate Limit.

- **Reset TCP Connection**—Sends TCP resets to hijack and terminate the TCP flow. Reset TCP Connection only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.

Understanding Deny Packet Inline

For signatures that have Deny Packet Inline configured as an action or for an event action override that adds Deny Packet Inline as an action, the following actions may be taken:

- Dropped Packet
- Denied Flow
- TCP One Way Reset Sent TCP

The Deny Packet Inline action is represented as a dropped packet action in the alert. When a Deny Packet Inline occurs for a TCP connection, it is automatically upgraded to a Deny Connection Inline action and seen as a denied flow in the alert. If the IPS denies just one packet, the TCP continues to try to send that same packet again and again, so the IPS denies the entire connection to ensure it never succeeds with the resends.

When a Deny Connection Inline occurs, the IPS also automatically sends a TCP one-way reset, which shows up as a TCP one-way reset sent in the alert. When the IPS denies the connection, it leaves an open connection on both the client (generally the attacker) and the server (generally the victim). Too many open connections can result in resource problems on the victim. So the IPS sends a TCP reset to the victim to close the connection on the victim side (usually the server), which conserves the resources of the victim. It also prevents a failover that would otherwise allow the connection to fail over to a different network path and reach the victim. The IPS leaves the attacker side open and denies all traffic from it.

TCP Reset Differences Between IPS Appliances and ASA IPS Modules

The IPS appliance sends TCP reset packets to both the attacker and victim when Reset TCP Connection is selected. The IPS appliance sends a TCP reset packet only to the victim under the following circumstances:

- When a Deny Packet Inline or Deny Connection Inline is selected
- When TCP-based signatures and Reset TCP Connection have NOT been selected

In the case of the ASA IPS modules, the TCP reset request is sent to the ASA, and then the ASA sends the TCP reset packets. The ASA sends TCP reset packets to both the attacker and victim when the Reset TCP Connection is selected. When Deny Packet Inline or Deny Connection Inline is selected, the ASA sends the TCP reset packet to either the attacker or victim depending on the configuration of the signature. Signatures configured to swap the attacker and victim when reporting the alert can cause the ASA to send the TCP reset packet to the attacker.

TCP Normalizer Signature Warning

You receive the following warning if you disable a default-enabled TCP Normalizer signature or remove a default-enabled modify packet inline, deny packet inline, or deny connection inline action:

Use caution when disabling, retiring, or changing the event action settings of a <Sig ID> TCP Normalizer signature for a sensor operating in IPS mode. The TCP Normalizer signature default values are essential for proper operation of the sensor.

If the sensor is seeing duplicate packets, consider assigning the traffic to multiple virtual sensors. If you are having problems with asymmetric or out-of-order TCP packets, consider changing the normalizer mode from strict evasion protection to asymmetric mode protection. Contact Cisco TAC if you require further assistance.

Signature Engines

A signature engine is a component of the Cisco IPS that is designed to support many signatures in a certain category. An engine is composed of a parser and an inspector. Each engine has a set of parameters that have allowable ranges or sets of values.

**Note**

The Cisco IPS engines support a standardized Regex.

Cisco IPS contains the following signature engines:

- **AIC**—Provides thorough analysis of web traffic. The AIC engine provides granular control over HTTP sessions to prevent abuse of the HTTP protocol. It allows administrative control over applications, such as instant messaging and gotomypc, that try to tunnel over specified ports. You can also use AIC to inspect FTP traffic and control the commands being issued. There are two AIC engines: AIC FTP and AIC HTTP.
- **Atomic**—The Atomic engines are combined into four engines with multi-level selections. You can combine Layer 3 and Layer 4 attributes within one signature, for example IP + TCP. The Atomic engine uses the standardized Regex support. The Atomic engines consist of the following types:
 - **Atomic ARP**—Inspects Layer 2 ARP protocol. The Atomic ARP engine is different because most engines are based on Layer 3 IP protocol.
 - **Atomic IP Advanced**—Inspects IPv6 Layer 3 and ICMPv6 Layer 4 traffic.
 - **Atomic IP**—Inspects IP protocol packets and associated Layer 4 transport protocols. This engine lets you specify values to match for fields in the IP and Layer 4 headers, and lets you use Regex to inspect Layer 4 payloads.

**Note**

All IP packets are inspected by the Atomic IP engine. This engine replaces the 4.x Atomic ICMP, Atomic IP Options, Atomic L3 IP, Atomic TCP, and Atomic UDP engines.

- **Atomic IPv6**—Detects two IOS vulnerabilities that are stimulated by malformed IPv6 traffic.
- **Fixed**—Performs parallel regular expression matches up to a fixed depth, then stops inspection using a single regular expression table. There are three Fixed engines: ICMP, TCP, and UDP.
- **Flood**—Detects ICMP and UDP floods directed at hosts and networks. There are two Flood engines: Flood Host and Flood Net.
- **Meta**—Defines events that occur in a related manner within a sliding time interval. This engine processes events rather than packets.
- **Multi String**—Inspects Layer 4 transport protocols and payloads by matching several strings for one signature. This engine inspects stream-based TCP and single UDP and ICMP packets.
- **Normalizer**—Configures how the IP and TCP Normalizer functions and provides configuration for signature events related to the IP and TCP Normalizer. Allows you to enforce RFC compliance.
- **Service**—Deals with specific protocols. The Service engines are divided in to the following protocol types:
 - **DNS**—Inspects DNS (TCP and UDP) traffic.
 - **FTP**—Inspects FTP traffic.

- FTP V2—Supports IOS IPS. This signature engine provides a protocol decode engine tuned for IOS IPS. If you try to use this engine, you receive an error message.
- Generic—Decodes custom service and payload, and generically analyzes network protocols.
- H225—Inspects VoIP traffic. Helps the network administrator make sure the SETUP message coming in to the VoIP network is valid and within the bounds that the policies describe. Is also helps make sure the addresses and Q.931 string fields such as url-ids, email-ids, and display information adhere to specific lengths and do not contain possible attack patterns.
- HTTP—Inspects HTTP traffic. The WEBPORTS variable defines inspection port for HTTP traffic.
- HTTP V2—Supports IOS IPS. This signature engine provides a protocol decode engine tuned for IOS IPS. If you try to use this engine, you receive an error message.
- IDENT—Inspects IDENT (client and server) traffic.
- MSRPC—Inspects MSRPC traffic.
- MSSQL—Inspects Microsoft SQL traffic.
- NTP—Inspects NTP traffic.
- P2P—Inspects P2P traffic.
- RPC—Inspects RPC traffic.
- SMB Advanced—Processes Microsoft SMB and Microsoft DCE/RPC (MSRPC) over SMB packets.

**Note**

The SMB engine has been replaced by the SMB Advanced engine. Even though the SMB engine is still visible in IDM, IME, and the CLI, its signatures have been obsoleted; that is, the new signatures have the obsoletes parameter set with the IDs of their corresponding old signatures. Use the new SMB Advanced engine to rewrite any custom signature that were in the SMB engine.

- SMTP V1—Supports IOS IPS.
This signature engine provides a protocol decode engine tuned for IOS IPS. If you try to use this engine, you receive an error message.
- SNMP—Inspects SNMP traffic.
- SSH—Inspects SSH traffic.
- TNS—Inspects TNS traffic.
- State—Conducts stateful searches of strings in protocols such as SMTP. The state engine has a hidden configuration file that is used to define the state transitions so new state definitions can be delivered in a signature update.
- String—Searches on Regex strings based on ICMP, TCP, or UDP protocol. There are three String engines: String ICMP, String TCP, and String UDP.
- String XL—Searches on Regex strings based on ICMP, TCP, or UDP protocol. The String XL engines provide optimized operation for the Regex accelerator card. There are three String engines: String ICMP XL, String TCP XL, and String UDP XL.

**Note**

The IPS 4345, IPS 4360, IPS 4510, IPS 4520, IPS 4520-XL, ASA 5525-X IPS SSP, ASA 5545-X IPS SSP, ASA 5555-X IPS SSP, and ASA 5585-X IPS SSP support the String XL engines and the Regex accelerator card.

**Note**

The Regex accelerator card is used for both the standard String engines and the String XL engines. Most standard String engine signatures can be compiled and analyzed by the Regex accelerator card without modification. However, there are special circumstances in which the standard String engine signatures cannot be compiled for the Regex accelerator card. In these situations a new signature is written in a String XL engine using the specific parameters in the String XL engine that do compile on the Regex accelerator card. The new signature in the String XL engine obsoletes the original signature in the standard String engine.

- Sweep—Analyzes sweeps from a single host (ICMP and TCP), from destination ports (TCP and UDP), and multiple ports with RPC requests between two nodes. There are two Sweep engines: Sweep and Sweep Other TCP.
- Traffic Anomaly—Inspects TCP, UDP, and other traffic for worms.
- Traffic ICMP—Analyzes nonstandard protocols, such as TFN2K, LOKI, and DDOS. There are only two signatures with configurable parameters.
- Trojan—Analyzes traffic from nonstandard protocols, such as BO2K andTFN2K. There are three Trojan engines: Bo2k, Tfn2k, and UDP. There are no user-configurable parameters in these engines.

Configuring Signature Definition Policies

This section describes how to configure signature definition policies, and contains the following topics:

- [Signature Definitions Pane, page 10-8](#)
- [Signature Definitions Pane Field Definitions, page 10-9](#)
- [Add and Clone Policy Dialog Boxes Field Definitions, page 10-9](#)
- [Adding, Cloning, and Deleting Signature Policies, page 10-9](#)

Signature Definitions Pane

**Note**

You must be administrator or operator to add, clone, or delete signature policies.

In the Signature Definitions pane, you can add, clone, or delete a signature definition policy. The default signature definition policy is called sig0. When you add a policy, a control transaction is sent to the sensor to create the policy instance. If the response is successful, the new policy instance is added under Signature Definitions. If the control transaction fails, for example because of resource limitations, an error message appears.

If your platform does not support virtual policies, this means you can only have one instance for each component and you cannot create new ones or delete the existing one. In this case, the Add, Clone, and Delete buttons are disabled.

Signature Definitions Pane Field Definitions

The following fields are found in the Signature Definitions pane:

- Policy Name—Identifies the name of this signature definition policy.
- Assigned Virtual Sensor—Identifies the virtual sensor to which this signature definition policy is assigned.

Add and Clone Policy Dialog Boxes Field Definitions

The following field is found in the Add and Clone Policy dialog boxes:

- Name—Specifies the name of the virtual sensor. The default virtual sensor is vs0.
- Assigned Interfaces (or Pairs)—Specifies the interfaces or interface pairs that belong to this virtual sensor.
- Signature Definition Policy—Specifies the name of the signature definition policy for this virtual sensor. The default signature definition policy is sig0.
- Event Action Override Policy—Specifies the name of the event action rules overrides policy for this virtual sensor. The default event action rules policy is rules0.
 - Risk Rating—Indicates the risk rating range (low, medium, or high risk) that should be used to trigger this event action override.
 - Actions to Add—Specifies the event action that will be added to an event if the conditions of this event action override are satisfied.
 - Enabled—Indicates whether or not this event action overrides policy is enabled.
- Anomaly Detection Policy—Specifies the name of the anomaly detection policy for this virtual sensor. The default anomaly detection policy is ad0.



Note Anomaly detection is disabled by default. You must enable it to configure or apply an anomaly detection policy. Enabling anomaly detection results in a decrease in performance.

- Description—The description of this virtual sensor.

Adding, Cloning, and Deleting Signature Policies

To add, clone, or delete a signature definition policy, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions**, and then click **Add**.
- Step 3** In the Policy Name field, enter a name for the signature definition policy.



Tip To discard your changes and close the Add Policy dialog box, click **Cancel**.

- Step 4** Click **OK**. The signature definition policy appears in the list in the Signature Definitions pane.

Step 5 To clone an existing signature definition policy, select it in the list, and then click **Clone**. The Clone Policy dialog box appears with “_copy” appended to the existing signature definition policy name.

Step 6 In the Policy Name field, enter a unique name.



Tip To discard your changes and close the Clone Policy dialog box, click **Cancel**.

Step 7 Click **OK**. The cloned signature definition policy appears in the list in the Signature Definitions pane.

Step 8 To remove a signature definition policy, select it, and then click **Delete**. The Delete Policy dialog box appears asking if you are sure you want to delete this policy permanently.



Caution You cannot delete the default signature definition policy, sig0.

Step 9 Click **Yes**. The signature definition policy no longer appears in the list in the Signature Definitions pane.

sig0 Pane

The sig0 menu in the left navigation pane contains the list of signatures listed by categories, for example, by active signatures, signature types, or all signatures. Once you choose a signature type in the menu, the sig0 pane is populated with the tools to configure signatures. You can filter the signatures by a variety of categories, for example, by signature ID, signature name, whether the signature is enabled, severity, fidelity rating, base risk rating, action, type, and engine.



Note You must select a signature category to see the signature configuration and add, clone, or edit signatures.

You can sort the data in each column by clicking the column head. The following columns are shown by default:

- ID
- Name
- Enabled
- Severity
- Fidelity Rating
- Base RR
- Signature Actions (Alert and Log, Deny, and Other)
- Type
- Engine
- Retired

To change the default column view, click the **Column** icon in the upper right of the pane and check or clear the check boxes in the Choose Columns to Display dialog box. You can also move the columns to a new location by selecting it and dragging it to a different place in the table.

There are configuration buttons grouped around the following configuration actions:

- Signature Configuration—Lets you edit event actions, enable and disable signatures, restore signature defaults, view signature information on MySDN, edit, add, delete, clone, and export signatures.
- Signature Wizard—Lets you use a wizard to create custom signatures.
- Advanced
 - Signature Variables—Lets you set up variables to use within multiple signatures.
 - Miscellaneous—Lets you configure application policy signatures, set up the mode for IP fragmentation and TCP stream reassembly, and configure IP logging.

MySDN



Note

Currently when you click **MySDN**, you are redirected to the IntelliShield site, which has replaced MySDN.

MySDN is a repository of information for individual signatures. It provides the following information about a signature:

- Signature ID
- Release version
- Original release date
- Latest release date
- Default enabled
- Default retired
- CVE
- Bugtraq ID
- Alarm severity
- Fidelity
- Description
- Recommended filters
- Benign filters
- IntelliShield alerts

The information from MySDN is available in the lower half of the sig0 pane. Select a signature in the list, and the information appears in the lower half. Or you can select a signature on **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**, and then click **MySDN**. After logging in to Cisco.com, you are taken to the specific information about that signature through the MySDN site ending at the IntelliShield site.

The IME launches MySDN from the last browser window you opened, which is the default setting for Windows. To change this default behavior, in Internet Explorer, choose **Tools > Internet Options**, and then click the **Advanced** tab. Scroll down and uncheck the **Reuse windows for launching shortcuts** check box.

**Note**

The MySDN website has been decommissioned and is no longer available to Cisco.com users. You can get to the information only through the IME.

Configuring Signatures

This section describes how to configure signatures. It contains the following topics:

- [Sig0 Pane Field Definitions, page 10-12](#)
- [Add, Clone, and Edit Signatures Dialog Boxes Field Definitions, page 10-13](#)
- [Edit Actions Dialog Box Field Definitions, page 10-15](#)
- [Enabling, Disabling, and Retiring Signatures, page 10-19](#)
- [Adding Signatures, page 10-19](#)
- [Cloning Signatures, page 10-21](#)
- [Tuning Signatures, page 10-22](#)
- [Assigning Actions to Signatures, page 10-23](#)
- [Configuring Alert Frequency, page 10-25](#)
- [Example Meta Engine Signature, page 10-27](#)
- [Example Atomic IP Advanced Engine Signature, page 10-30](#)
- [Example String XL TCP Match Offset Signature, page 10-32](#)
- [Example String XL TCP Engine Minimum Match Length Signature, page 10-35](#)

Sig0 Pane Field Definitions

The following fields are found in the Sig0 pane:

- **Filter**—Lets you sort the list of signatures by selecting an attribute to filter.
- **ID**—Identifies the unique numerical value assigned to this signature and subsignature. This value lets the sensor identify a particular signature.
- **Name**—Identifies the name assigned to the signature.
- **Enabled**—Identifies whether or not the signature is enabled. A signature must be enabled for the sensor to protect against the traffic specified by the signature.
- **Severity**—Identifies the severity level that the signature will report: High, Informational, Low, Medium.
- **Fidelity Rating**—Identifies the weight associated with how well this signature might perform in the absence of specific knowledge of the target.
- **Base RR**—Displays the base risk rating value of each signature. IDM automatically calculates the base risk rating by multiplying the fidelity rating and the severity factor and dividing them by 100 (Fidelity Rating x Severity Factor /100). Severity Factor has the following values:
 - Severity Factor = 100 if the severity level of the signature is high
 - Severity Factor = 75 if severity level of the signature is medium
 - Severity Factor = 50 if severity level of the signature is low

- Severity Factor = 25 if severity level of the signature is informational
- Signature Actions—Identifies the actions the sensor will take when this signature fires.
- Type—Identifies whether this signature is a default (built-in), tuned, or custom signature.
- Engine—Identifies the engine that parses and inspects the traffic specified by this signature.
- Retired—Identifies whether or not the signature is retired. A retired signature is removed from the signature engine. You can activate a retired signature to place it back in the signature engine.



Note We recommend that you retire any signatures that you are not using. This improves sensor performance.

Button and Right-Click Menu Functions:

- Edit Actions—Opens the Edit Actions dialog box.
- Enable—Enables the selected signature.
- Disable—Disables the selected signature.
- Set Severity To—Lets you set the severity level that the signature will report: High, Medium, Low or Informational.
- Restore Default—Returns all parameters to the default settings for the selected signature.
- Show Events—Displays the events related to this signature in real time, from the last 10 minutes, or from the last hour.
- MySDN—Takes you to the description of that signature on the MySDN site on Cisco.com.
- Edit—Opens the Edit Signature dialog box. In the Edit Signature dialog box, you can change the parameters associated with the selected signature and effectively *tune* the signature. You can edit only one signature at a time.
- Add—Opens the Add Signature dialog box. In the Add Signature dialog box, you can add the parameters associated with the selected signature and effectively *tune* the signature.
- Delete—Deletes the selected custom signature. You cannot delete built-in signatures.
- Clone—Opens the Clone Signature dialog box. In the Clone Signature dialog box, you can create a signature by changing the prepopulated values of the existing signature you chose to clone.
- Change Status To—Lets you change the status to Active, Retired, Low Memory Retired, Medium Memory Retired.
- Export—Lets you export currently displayed signatures in the table to a comma-separated Excel file (using CSV) or HTML file. You can also use **Ctrl-C** to copy the contents in to a clipboard and later paste in to Notepad or Word using **Ctrl-V**.

Add, Clone, and Edit Signatures Dialog Boxes Field Definitions

The following fields are found in the Add, Clone, and Edit Signature dialog boxes:

- Signature Definition—Defines the signature:
 - Signature ID—Identifies the unique numerical value assigned to this signature. This value lets the sensor identify a particular signature. The value is 1000 to 65000.
 - SubSignature ID—Identifies the unique numerical value assigned to this subsignature. The subsignature ID identifies a more granular version of a broad signature. The value is 0 to 255.

- Alert Severity—Lets you choose the severity level of the signature: High, Informational, Low, Medium.
- Sig Fidelity Rating—Lets you choose the weight associated with how well this signature might perform in the absence of specific knowledge of the target. The value is 0 to 100. The default is 75.
- Promiscuous Delta—Lets you determine the seriousness of the alert.

**Caution**

We recommend that you do NOT change the promiscuous delta setting for a signature.

- Sig Description—Lets you specify the following attributes that help you distinguish this signature from other signatures:
 - Signature Name—Specifies the name your signature. The default is MySig.
 - Alert Notes—Lets you add alert notes in this field.
 - User Comments—Lets you add your comments about this signature in this field.
 - Alarm Traits—Lets you add the alarm trait in this field. The value is 0 to 65535. The default is 0.
 - Release—Lets you add the software release in which the signature first appeared.
 - Signature Creation Date—Specifies the date this signature was created.
 - Signature Type—Specifies the type of signature (anomaly, component, exploit, other, vulnerability)
- Engine—Lets you choose the engine that parses and inspects the traffic specified by this signature.
- Event Action—Lets you assign the actions the sensor takes when it responds to events.
- Event Counter—Lets you configure how the sensor counts events. For example, you can specify that you want the sensor to send an alert only if the same signature fires 5 times for the same address set:
 - Event Count—Specifies the number of times an event must occur before an alert is generated. The value is 1 to 65535. The default is 1.
 - Event Count Key—Specifies the storage type used to count events for this signature. Choose attacker address, attacker address and victim port, attacker and victim addresses, attacker and victim addresses and ports, or victim address. The default is attacker address.
 - Specify Alert Interval—Specifies the time in seconds before the event count is reset. Choose Yes or No from the drop-down list and then specify the amount of time.
- Alert Frequency—Lets you configure how often the sensor alerts you when this signature is firing. Specify the following parameters for this signature:
 - Summary Mode—Specifies the mode of alert summarization. Choose Fire All, Fire Once, Global Summarize, or Summarize.

**Note**

When multiple contexts from the adaptive security appliance are contained in one virtual sensor, the summary alerts contain the context name of the last context that was summarized. Thus, the summary is the result of all alerts of this type from all contexts that are being summarized.

- Summary Interval—Specifies the time in seconds used in each summary alert. The value is 1 to 65535. The default is 15.

- Summary Key—Specifies the storage type used to summarize alerts. Choose Attacker address, Attacker address and victim port, Attacker and victim addresses, Attacker and victim addresses and ports, or Victim address. The default is Attacker address.
- Specify Global Summary Threshold—Lets you specify the threshold number of events to take the alert into global summary. Choose Yes or No and then specify the threshold number of events.
- Status—Lets you enable or disable a signature, or retire or unretire a signature:
 - Enabled—Lets you choose whether the signature is enabled or disabled. The default is yes (enabled).
 - Retired—Let you choose whether the signature is retired or not and whether it is low memory retired or medium memory retired. The default is no (not retired). Low memory retired platforms have less than 1 GB of maximum sensor memory. Medium memory retired platforms have at least 1 GB and less than 2 GB of maximum sensor memory. As the signatures are loaded, the value of retired is evaluated based on the platform loading the signatures.
 - Obsoletes—Specifies the list of the signatures that are obsoleted by this signature.
 - Vulnerable OS List—Lets you choose a vulnerable OS for this signature.
- Mars Category—Lets you map this signature to a MARS attack category. This is a static information category that you can view in the alerts.

Edit Actions Dialog Box Field Definitions

An event action is the response of the sensor to an event. Event actions are configurable on a per-signature basis. The following fields are found in the Edit Actions dialog box:

Alert and Log Actions

- Product Alert—Writes the event to the Event Store as an alert.



Note The Product Alert action is not automatic when you enable alerts for a signature. To have an alert created in the Event Store, you must select Product Alert. If you add a second action, you must include Product Alert if you want an alert sent to the Event Store. Also, every time you configure the event actions, a new list is created and it replaces the old list. Make sure you include all the event actions you need for each signature.



Note There are other event actions that force a Product Alert. These actions use Product Alert as the vehicle for performing the action. Even if Product Alert is not selected or is filtered, the alert is still produced. The actions are the following: Produce Verbose Alert, Request SNMP Trap, Log Attacker Packets, Log Victim Packets, and Log Pair Packets.



Note A Produce Alert event action is added for an event when global correlation has increased the risk rating of an event, and has added either the Deny Packet Inline or Deny Attacker Inline event action.

- Produce Verbose Alert—Includes an encoded dump of the offending packet in the alert. This action causes an alert to be written to the Event Store, even if Product Alert is not selected.

- **Log Attacker Packets**—Starts IP logging on packets that contain the attacker address and sends an alert. This action causes an alert to be written to the Event Store, even if Product Alert is not selected.
- **Log Victim Packets**—Starts IP logging on packets that contain the victim address and sends an alert. This action causes an alert to be written to the Event Store, even if Product Alert is not selected.
- **Log Pair Packets**—Starts IP logging on packets that contain the attacker/victim address pair. This action causes an alert to be written to the Event Store, even if Product Alert is not selected.
- **Request SNMP Trap**—Sends a request to the Notification Application component of the sensor to perform SNMP notification. This action causes an alert to be written to the Event Store, even if Product Alert is not selected. You must have SNMP configured on the sensor to implement this action.

Deny Actions

- **Deny Packet Inline (inline only)**—Terminates the packet.



Note You cannot delete the event action override for Deny Packet Inline because it is protected. If you do not want to use that override, disable it.

- **Deny Connection Inline (inline only)**—Terminates the current packet and future packets on this TCP flow.
- **Deny Attacker Victim Pair Inline (inline only)**—Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time.



Note For deny actions, to set the specified period of time and maximum number of denied attackers, choose **Configuration > sensor_name > Policies > Event Action Rules > rules0 > General Settings**.

- **Deny Attacker Service Pair Inline (inline only)**—Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
- **Deny Attacker Inline (inline only)**—Terminates the current packet and future packets from this attacker address for a specified period of time.
- The sensor maintains a list of attackers being denied by the system. To remove an entry from the denied attacker list, you can view the list of attackers and clear the entire list, or you can wait for the timer to expire. The timer is a sliding timer for each entry. Therefore, if attacker A is being denied, but issues another attack, the timer for attacker A is reset and attacker A remains in the denied attacker list until the timer expires. If the denied attacker list is at capacity and cannot add a new entry, the packet is still denied.



Note This is the most severe of the deny actions. It denies current and future packets from a single attacker address. To clear all denied attacker entries, choose **Configuration > sensor_name > Sensor Management > Time-Based Actions > Denied Attackers > Clear List**, which permits the addresses back on the network.

- **Modify Packet Inline (inline only)**—Modifies packet data to remove ambiguity about what the end point might do with the packet.



Note You cannot use Modify Packet Inline as an action when adding event action filters or overrides.

Other Actions

- Request Block Connection—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action.



Note Connection blocks and network blocks are not supported on adaptive security appliances. Adaptive security appliances only support host blocks with additional connection information.



Note IPv6 does not support Request Block Connection.

- Request Block Host—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action.



Note IPv6 does not support Request Block Host.



Note For block actions, to set the duration of the block, choose **Configuration > sensor_name > Policies > Event Action Rules > rules0 > General Settings**.

- Request Rate Limit—Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action.



Note Request Rate Limit applies to a select set of signatures.



Note IPv6 does not support Request Rate Limit.

- Reset TCP Connection—Sends TCP resets to hijack and terminate the TCP flow. Reset TCP Connection only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.

Understanding Deny Packet Inline

For signatures that have Deny Packet Inline configured as an action or for an event action override that adds Deny Packet Inline as an action, the following actions may be taken:

- Dropped Packet
- Denied Flow
- TCP One Way Reset Sent TCP

The Deny Packet Inline action is represented as a dropped packet action in the alert. When a Deny Packet Inline occurs for a TCP connection, it is automatically upgraded to a Deny Connection Inline action and seen as a denied flow in the alert. If the IPS denies just one packet, the TCP continues to try to send that same packet again and again, so the IPS denies the entire connection to ensure it never succeeds with the resends.

When a Deny Connection Inline occurs, the IPS also automatically sends a TCP one-way reset, which shows up as a TCP one-way reset sent in the alert. When the IPS denies the connection, it leaves an open connection on both the client (generally the attacker) and the server (generally the victim). Too many open connections can result in resource problems on the victim. So the IPS sends a TCP reset to the victim to close the connection on the victim side (usually the server), which conserves the resources of the victim. It also prevents a failover that would otherwise allow the connection to fail over to a different network path and reach the victim. The IPS leaves the attacker side open and denies all traffic from it.

TCP Reset Differences Between IPS Appliances and ASA IPS Modules

The IPS appliance sends TCP reset packets to both the attacker and victim when Reset TCP Connection is selected. The IPS appliance sends a TCP reset packet only to the victim under the following circumstances:

- When a Deny Packet Inline or Deny Connection Inline is selected
- When TCP-based signatures and Reset TCP Connection have NOT been selected

In the case of the ASA IPS modules, the TCP reset request is sent to the ASA, and then the ASA sends the TCP reset packets. The ASA sends TCP reset packets to both the attacker and victim when the Reset TCP Connection is selected. When Deny Packet Inline or Deny Connection Inline is selected, the ASA sends the TCP reset packet to either the attacker or victim depending on the configuration of the signature. Signatures configured to swap the attacker and victim when reporting the alert can cause the ASA to send the TCP reset packet to the attacker.

TCP Normalizer Signature Warning

You receive the following warning if you disable a default-enabled TCP Normalizer signature or remove a default-enabled modify packet inline, deny packet inline, or deny connection inline action:

Use caution when disabling, retiring, or changing the event action settings of a <Sig ID> TCP Normalizer signature for a sensor operating in IPS mode. The TCP Normalizer signature default values are essential for proper operation of the sensor.

If the sensor is seeing duplicate packets, consider assigning the traffic to multiple virtual sensors. If you are having problems with asymmetric or out-of-order TCP packets, consider changing the normalizer mode from strict evasion protection to asymmetric mode protection. Contact Cisco TAC if you require further assistance.

For More Information

- For the procedure for configuring the general settings, see [Configuring General Settings, page 12-33](#).
- For the procedure for configuring SNMP, see [Chapter 18, “Configuring SNMP.”](#)
- For the procedure for configuring denied attackers, see [Configuring and Monitoring Denied Attackers, page 17-1](#).

Enabling, Disabling, and Retiring Signatures

To enable, disable, and retire signatures, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**.
- Step 3** To locate a signature, choose a sorting option from the Filter drop-down list. For example, if you are searching for a Flood Host signature, chose **Engine** from the drop-down list, then **Flood Host**, and then select the individual signature. The sig0 pane refreshes and displays only those signatures that match your sorting criteria.
- Step 4** To enable or disable an existing signature, select the signature, and follow these steps:
- View the Enabled column to determine the status of the signature. A signature that is enabled has the check box checked.
 - To enable a signature that is disabled, check the **Enabled** check box.
 - To disable a signature that is enabled, remove the check from the **Enabled** check box.
 - To retire one or more signatures, select the signature(s), right-click, and then click **Change Status To > Retired**.



Note We recommend that you retire any signatures that you are not using. This improves sensor performance.



Tip To discard your changes, click **Reset**.

- Step 5** Click **Apply** to apply your changes and save the revised configuration.
-

Adding Signatures



Tip An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.

To create a custom signature that is not based on an existing signature, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**, and then click **Add**.
- Step 3** In the Signature ID field, enter a unique signature ID for the new signature. Custom signature IDs start at 60000.
- Step 4** In the Subsignature field, enter a unique subsignature ID for the new signature.
- Step 5** From the Alert Severity drop-down list, choose the severity you want to associate with this signature.

Step 6 In the Sig Fidelity Rating field, enter a value between 1 and 100 to represent the signature fidelity rating for this signature.

Step 7 In the Promiscuous Delta field, enter the promiscuous delta (between 0 and 30) that you want to associate with this signature.



Caution We recommend that you do NOT change the promiscuous delta setting for a signature.

Step 8 Complete the Sig Description fields and add any comments about this signature.

Step 9 From the Engine drop-down list, choose the engine the sensor will use to enforce this signature.



Note If you do not know which engine to select, use the Custom Signature Wizard to help you create a custom signature.

Step 10 Assign actions to this signature.

Step 11 Configure the engine-specific parameters for this signature.

Step 12 Configure the Event Counter:

- a. In the Event Count field, enter the number of events you want counted (1 to 65535).
- b. From the Event Count Key drop-down list, choose the key you want to use.
- c. From the Specify Alert Interface drop-down list, choose whether you want to specify the alert interval (Yes or No).
- d. If you chose Yes, enter the alert interval (2 to 1000) in the Alert Interval field.

Step 13 Configure the alert frequency.

Step 14 Configure the status of the signature:

- a. From the Enabled drop-down list, choose **Yes** to enable the signature.



Note A signature must be enabled for the sensor to actively detect the attack specified by the signature.

- b. From the Retired drop-down list, choose **Yes** to make sure the signature is active. This places the signature in the engine.



Note A signature must not be retired for the sensor to actively detect the attack specified by the signature.

- c. Choose the vulnerable OS(es).



Tip To select more than one OS, hold down the **Ctrl** key.

Step 15 Choose the MARS category and click **OK**.



Tip To discard your changes and close the Add Signature dialog box, click **Cancel**.

Step 16 Click **OK**. The new signature appears in the list with the Type set to Custom.



Tip To discard your changes, click **Reset**.

Step 17 Click **Apply** to apply your changes and save the revised configuration.

Cloning Signatures



Tip An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.



Caution Some signature values in built-in signature are protected, which means that you cannot copy that value. You can still clone the signature, but you cannot configure certain values. You will receive an error message similar to the following when a signature value cannot be configured:
[Obsoletes] is protected, cannot copy the value. [Mars Category] is protected, cannot copy the value.

On the sig0 pane, you can create a signature by cloning an existing signature. This task can save you time when you are creating signatures that are similar. To create a signature by using an existing signature as the starting point, follow these steps:

Step 1 Log in to the IME using an account with administrator or operator privileges.

Step 2 Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**.

Step 3 To locate a signature, choose a sorting option from the Filter drop-down list. For example, if you are searching for a Flood Host signature, choose **Engine** from the drop-down list, then **Flood Host**, and then select the individual signature. The sig0 pane refreshes and displays only those signatures that match your sorting criteria.

Step 4 Select the signature and click **Clone**.

Step 5 In the Signature field, enter a unique signature ID for the new signature. Custom signature IDs start at 60000.

Step 6 In the Subsignature field, enter a unique subsignature ID for the new signature.

Step 7 Review the parameter values and change the value of any parameter you want to be different for this new signature.



Tip To select more than one OS or event action, hold down the **Ctrl** key.

Step 8 Configure the status of the signature:

- a. From the Enabled drop-down list, choose **Yes** to enable the signature.



Note A signature must be enabled for the sensor to actively detect the attack specified by the signature.

- b. From the Retired drop-down list, choose **Yes** to make sure the signature is active. This places the signature in the engine.



Note A signature must not be retired for the sensor to actively detect the attack specified by the signature.



Tip To discard your changes and close the Clone Signature dialog box, click **Cancel**.

- c. Click **OK**. The cloned signature now appears in the list with the Type set to Custom.



Tip To discard your changes, click **Reset**.

Step 9 Click **Apply** to apply your changes and save the revised configuration.

Tuning Signatures



Tip An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.

You can tune built-in signatures by adjusting several signature parameters. Built-in signatures that have been modified are called *tuned* signatures. On the sig0 pane, you can edit, or *tune* a signature.

To tune an existing signature, follow these steps:

- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**.
- Step 3** To locate a signature, choose a sorting option from the Filter drop-down list. For example, if you are searching for a Flood Host signature, choose **Engine** from the drop-down list, then **Flood Host**, and then select the individual signature. The sig0 pane refreshes and displays only those signatures that match your sorting criteria.
- Step 4** Select the signature and click **Edit**.
- Step 5** Review the parameter values and change the value of any parameter you want to tune.



Tip To select more than one OS, event action, vulnerable OS, or MARS category, hold down the **Ctrl** key.

Step 6 Configure the status of the signature:

- a. From the Enabled drop-down list, choose **Yes** to enable the signature.



Note A signature must be enabled for the sensor to actively detect the attack specified by the signature.

- b. From the Retired drop-down list, choose **Yes** to make sure the signature is active. This places the signature in the engine.



Note A signature must not be retired for the sensor to actively detect the attack specified by the signature.



Tip To discard your changes and close the Edit Signature dialog box, click **Cancel**.

Step 7 Click **OK**. The edited signature now appears in the list with the Type set to Tuned.



Tip To discard your changes, click **Reset**.

Step 8 Click **Apply** to apply your changes and save the revised configuration.

Assigning Actions to Signatures

On the sig0 pane, you can assign actions to a signature.

To edit actions for a signature or a set of signatures, follow these steps:

Step 1 Log in to the IME using an account with administrator or operator privileges.

Step 2 Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**.

Step 3 To locate a signature, choose a sorting option from the Filter drop-down list. For example, if you are searching for a Flood Host signature, choose **Engine** from the drop-down list, then **Flood Host**, and then select the individual signature. The sig0 pane refreshes and displays only those signatures that match your sorting criteria.

Step 4 Select the signature(s), and click **Edit Actions**.

Step 5 Check the check boxes next to the actions you want to assign to the signature(s).



Note A check mark indicates that the action is assigned to the selected signature(s). No check mark indicates that the action is not assigned to any of the selected signatures. A gray check mark indicates that the action is assigned to some of the selected signatures.



Tip To select more than one action, hold down the **Ctrl** key.

Choose from the following actions:

- Produce Alert—Writes the event to Event Store as an alert.
- Produce Verbose Alert—Includes an encoded dump of the offending packet in the alert. This action causes an alert to be written to Event Store, even if Produce Alert is not selected.
- Log Attacker Packets—Starts IP logging on packets that contain the attacker address and sends an alert. This action causes an alert to be written to Event Store, even if Produce Alert is not selected.
- Log Victim Packets—Starts IP Logging on packets that contain the victim address and sends an alert. This action causes an alert to be written to Event Store, even if Produce Alert is not selected.
- Log Pair Packets—Starts IP Logging on packets that contain the attacker/victim address pair. This action causes an alert to be written to Event Store, even if Produce Alert is not selected.
- Request SNMP Trap—Sends a request to NotificationApp to perform SNMP notification. This action causes an alert to be written to Event Store, even if Produce Alert is not selected. You must have SNMP configured on the sensor to implement this action.
- Deny Packet Inline (inline only)—Does not transmit this packet.
- Deny Connection Inline (inline only)—Does not transmit this packet and future packets on the TCP flow.
- Deny Attacker Victim Pair Inline (inline only)—Does not transmit this packet and future packets on the attacker/victim address pair for a specified period of time.
- Deny Attacker Service Pair Inline (inline only)—Does not transmit this packet and future packets on the attacker address victim port pair for a specified period of time.
- Deny Attacker Inline (inline only)—Does not transmit this packet and future packets originating from the attacker address for a specified period of time.
- Modify Packet Inline (inline only)—Modifies packet data to remove ambiguity about what the end point might do with the packet.
- Request Block Connection—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action.
- Request Block Host—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action.
- Request Rate Limit—Sends a rate limit request to ARC to perform rate limiting. You must have rate limiting devices configured to implement this action.
- Reset TCP Connection—Sends TCP resets to hijack and terminate the TCP flow. Reset TCP Connection only works on TCP signatures that analyze a single connection. It does not work for sweeps or floods.



Tip

To discard your changes and close the Assign Actions dialog box, click **Cancel**.

Step 6

Click **OK** to save your changes and close the dialog box. The new action(s) now appears in the Action column.



Tip

To discard your changes, click **Reset**.

Step 7

Click **Apply** to apply your changes and save the revised configuration.

For More Information

- For the procedure for configuring the general settings, see [Configuring General Settings, page 12-33](#).
- For the procedure for configuring SNMP, see [Chapter 18, “Configuring SNMP.”](#)
- For the procedure for configuring denied attackers, see [Configuring and Monitoring Denied Attackers, page 17-1](#).

Configuring Alert Frequency

**Tip**

An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.

You can control how often a signature fires. For example, you may want to decrease the volume of alerts sent out from the sensor. Or you may want the sensor to provide basic aggregation of signature firings into a single alert. Or you may want to counter anti-IPS tools such as “stick,” which are designed to send bogus traffic so that the IPS produces thousands of alerts during a very short time.

To configure the alert frequency of a signature, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**.
- Step 3** Click **Add** to add a signature, choose a signature to clone, and click **Clone**, or choose a signature to edit, and click **Edit**.
- Step 4** Configure the event count, key, and alert interval:
- In the Event Count field, enter a value for the event count. This is the minimum number of hits the sensor must receive before sending one alert for this signature.
 - From the Event Count Key drop-down list, choose an attribute to use as the Event Count Key. For example, if you want the sensor to count events based on whether or not they are from the same attacker, choose Attacker address as the Event Count Key.
 - If you want to count events based on a rate, choose **Yes** from the Specify Event Interval drop-down list, and then in the Alert Interval field, enter the number of seconds that you want to use for your interval.
- Step 5** To control the volume of alerts and configure how the sensor summarizes alerts, choose one of the following options from the Summary Mode drop-down list:
- **Fire All**—Specifies that you want the sensor to send an alert every time the signature detects malicious traffic. You can then specify additional thresholds that let the sensor dynamically adjust the volume of alerts. Go to Step 6.
 - **Fire Once**—Specifies that you want the sensor to send an alert the first time the signature detects malicious traffic. You can then specify additional thresholds that let the sensor dynamically adjust the volume of alerts. Go to Step 7.
 - **Summarize**—Specifies that you want the sensor to only send summary alerts for this signature instead of sending alerts every time the signature fires. You can then specify additional thresholds that let the sensor dynamically adjust the volume of alerts. Go to Step 8.

- **Global Summarize**—Specifies that you want the sensor to send an alert the first time a signature fires on an address set, and then only send a global summary alert that includes a summary of all alerts for all address sets over a given time interval. Go to Step 9.

Step 6 Configure the Fire All option:

- a. From the Specify Summary Threshold drop-down list, choose **Yes**.
- b. In the Summary Threshold field, enter the minimum number of hits the sensor must receive before sending a summary alert for this signature.
- c. In the Summary Interval field, enter the number of seconds that you want to use for the time interval.
- d. To have the sensor enter global summarization mode, choose **Yes** from the Specify Global Summary Threshold drop-down list.
- e. In the Global Summary Threshold field, enter the minimum number of hits the sensor must receive before sending a global summary alert.
- f. From the Summary Key drop-down list, choose the type of summary key. The summary key identifies the attribute to use for counting events. For example, if you want the sensor to count events based on whether or not they are from the same attacker, choose Attacker address as the summary key.

Step 7 Configure the Fire Once option:

- a. From the Summary Key drop-down list, choose the type of summary key. The summary key identifies the attribute to use for counting events. For example, if you want the sensor to count events based on whether or not they are from the same attacker, choose Attacker address as the summary key.
- b. To have the sensor use global summarization, choose **Yes** from the Specify Global Summary Threshold drop-down list.
- c. In the Global Summary Threshold field, enter the minimum number of hits the sensor must receive before sending a global summary alert. When the alert rate exceeds a specified number of signatures in a specified number of seconds, the sensor changes from sending a single alert the first time a signature fires to sending a single global summary alert. When the rate during the interval drops below this threshold, the sensor reverts to its configured alert behavior.



Note

When multiple contexts from the adaptive security appliance are contained in one virtual sensor, the summary alerts contain the context name of the last context that was summarized. Thus, the summary is the result of all alerts of this type from all contexts that are being summarized.

- d. In the Summary Interval field, enter the number of seconds during which the sensor counts events for summarization.

Step 8 Configure the Summarize option:

- a. In the Summary Interval field, enter the number of seconds during which the sensor counts events for summarization.
- b. From the Summary Key drop-down list, choose the type of summary key. The summary key identifies the attribute to use for counting events. For example, if you want the sensor to count events based on whether or not they are from the same attacker, choose Attacker address as the summary key.
- c. To have the sensor use dynamic global summarization, choose **Yes** from the Specify Global Summary Threshold drop-down list.

- d. In the Global Summary Threshold field, enter the minimum number of hits the sensor must receive before sending a global summary alert.



Note When the alert rate exceeds a specified number of signatures in a specified number of seconds, the sensor changes from sending a single alert the first time a signature fires to sending a single global summary alert. When the rate during the interval drops below this threshold, the sensor reverts to its configured alert behavior.

Step 9 To configure the Global Summarize option, in the Summary Interval field, enter the number of seconds during which the sensor counts events for summarization.

Step 10 Click **OK** to save your alert behavior changes. You are returned to the sig0 pane.



Tip To discard your changes, click **Cancel**.

Step 11 To apply your alert behavior changes to the signature configuration, click **Apply**. The signature you added or edited is enabled and added to the list of signatures.

Example Meta Engine Signature



Caution A custom signature can affect the performance of your sensor. Test the custom signature against a baseline sensor performance for your network to determine the overall impact of the signature.



Caution A large number of Meta engine signatures could adversely affect overall sensor performance.

The Meta engine defines events that occur in a related manner within a sliding time interval. This engine processes events rather than packets. As signature events are generated, the Meta engine inspects them to determine if they match any or several Meta definitions. The Meta engine generates a signature event after all requirements for the event are met.

All signature events are handed off to the Meta engine by the Signature Event Action Processor. The Signature Event Action Processor hands off the event after processing the minimum hits option. Summarization and event action are processed after the Meta engine has processed the component events.



Note The Meta engine is different from other engines in that it takes alerts as input where most engines take packets as input.



Tip An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.

The following example demonstrates how to create a signature based on the Meta engine. For example, the following custom signature fires when it sees the alerts from signature 2000 subsignature 0 and signature 3000 subsignature 0 on the same source address. The source address selection is a result of the meta key default value of Axxx. You can change the behavior by changing the meta key setting to xxBx (destination address) for example.

To create a signature based on the Meta engine, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
 - Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**, and then click **Add**.
 - Step 3** In the Signature ID field, enter a unique signature ID for the new signature. Custom signature IDs start at 60000.
 - Step 4** In the Subsignature field, enter a unique subsignature ID for the new signature.
 - Step 5** From the Alert Severity drop-down list, choose the severity you want to associate with this signature.
 - Step 6** In the Signature Fidelity Rating field, enter a value between 1 and 100 to represent the signature fidelity rating for this signature.
 - Step 7** Leave the default value for the Promiscuous Delta field.
 - Step 8** Complete the signature description fields and add any comments about this signature.
 - Step 9** From the Engine drop-down list, choose **Meta**.
 - Step 10** Configure the Meta engine-specific parameters:
 - a. From the Event Action drop-down list, choose the actions you want the sensor to take when it responds to an event.



Tip To choose more than one action, hold down the **Ctrl** key.

- b. From the Swap Attacker Victim drop-down list, choose **Yes** to swap the destination and source ports in the alert message.
- c. In the Meta Reset Interval field, enter the time in seconds to reset the Meta signature. The valid range is 0 to 3600 seconds. The default is 60 seconds.
- d. Click the pencil icon next to Component List to insert the new Meta signature. The Component List dialog box appears.
- e. Click **Add** to insert the first Meta signature. The Add List Entry dialog box appears.
- f. In the Entry Key field, enter a name for the entry, for example, Entry1. The default is MyEntry.
- g. In the Component Sig ID field, enter the signature ID of the signature (2000 in this example) on which to match this component.
- h. In the Component SubSig ID field, specify the subsignature ID of the signature (0 in this example) on which to match this component.
- i. In the Component Count field, enter the number of times this component must fire before it is satisfied.
- j. In the Is a NOT component field, choose **Yes** if you want this to be NOT component.
- k. Click **OK**. You are returned to the Add List Entry dialog box.
- l. Select your entry and click **Select** to move it to the Selected Entries list.

- m. Click **OK**.
- n. Click **Add** to insert the next Meta signature. The Add List Entry dialog box appears.
- o. In the Entry Key field, enter a name for the entry, for example Entry2.
- p. In the Component Sig ID field, enter the signature ID of the signature (3000 in this example) on which to match this component.
- q. In the Component SubSig ID field, enter the subsignature ID of the signature (0 in this example) on which to match this component.
- r. In the Component Count field, enter the number of times this component must fire before it is satisfied.
- s. Click **OK**. You are returned to the Add List Entry dialog box.
- t. Select your entry and click **Select** to move it to the Selected Entries list.
- u. Select the new entry and click **Move Up** or **Move Down** to order the new entry.



Tip Click **Reset Ordering** to return the entries to the Entry Key list.

- v. Click **OK**.
- w. From the Meta Key drop-down list, choose the storage type for the Meta signature:
 - Attacker address
 - Attacker and victim addresses
 - Attacker and victim addresses and ports
 - Victim address
- x. In the Unique Victims field, enter the number of unique victims required for this signature. The valid value is 1 to 256. The default is 1.
- y. From the Component List in Order drop-down list, choose **Yes** to have the component list fire in order.

Step 11 In the Component List in Order field, you can choose **Yes** to have the component list fire in order. For example, if signature 3000 in the Entry2 component fires before signature 2000 in the Entry1 component, the custom Meta signature will not fire.

Step 12 In the All Components Required field and the All NOT Components Required field, choose **Yes** to use all components and NOT components. This option works with the All NOT Components Required option, if you have NOT components configured as required, the Meta signature will not fire.

Step 13 Configure Event Counter:

- a. In the Event Count field, enter the number of events you want counted (1 to 65535).
- b. From the Event Count Key drop-down list, choose the key you want to use.
- c. From the Specify Alert Interface drop-down list, choose whether you want to specify the alert interval (Yes or No).
- d. If you chose Yes, enter the alert interval (2 to 1000) in the Alert Interval field.

Step 14 Configure the alert frequency.

Step 15 Leave the default (**Yes**) for the Enabled field.



Note A signature must be enabled for the sensor to actively detect the attack specified by the signature.

Step 16 Leave the default (**Yes**) for the Retired field. This places the signature in the engine.



Note A signature must not be retired for the sensor to actively detect the attack specified by the signature.

Step 17 From the Vulnerable OS List drop-down list, choose the operating systems that are vulnerable to this signature.



Tip To choose more than one action, hold down the **Ctrl** key.

Step 18 From the Mars Category drop-down list, choose the MARS categories you want this signature to identify.



Tip To choose more than one action, hold down the **Ctrl** key.



Tip To discard your changes and close the Add Signature dialog box, click **Cancel**.

Step 19 Click **OK**. The new signature appears in the list with the Type set to Custom.



Tip To discard your changes, click **Reset**.

Step 20 Click **Apply** to apply your changes and save the revised configuration.

For More Information

For detailed information on the Meta engine, see [Meta Engine, page B-32](#).

Example Atomic IP Advanced Engine Signature



Tip An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.



Caution A custom signature can affect the performance of your sensor. Test the custom signature against a baseline sensor performance for your network to determine the overall impact of the signature.

The following example demonstrates how to create a signature based on the Atomic IP Advanced engine. For example, the following custom signature matches any packets that are IPv6 with a HOP Option Header where the header is type 1 and the length is 8.

To create a signature based on the Atomic IP Advanced engine, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**, and then click **Add**.
- Step 3** In the Signature ID field, enter a unique signature ID for the new signature. Custom signature IDs start at 60000.
- Step 4** In the Subsignature field, enter a unique subsignature ID for the new signature.
- Step 5** From the Alert Severity drop-down list, choose the severity you want to associate with this signature.
- Step 6** In the Signature Fidelity Rating field, enter a value between 1 and 100 to represent the signature fidelity rating for this signature.
- Step 7** Leave the default value for the Promiscuous Delta field.
- Step 8** Complete the signature description fields and add any comments about this signature.
- Step 9** From the Engine drop-down list, choose **Atomic IP Advanced**.
- Step 10** Configure the Atomic IP Advanced engine-specific parameters:
- From the Event Action drop-down list, choose the actions you want the sensor to take when it responds to an event.



Note IPv6 does not support the following event actions: Request Block Host, Request Block Connection, or Request Rate Limit.



Tip To choose more than one action, hold down the **Ctrl** key.

- From the IP Version drop-down list, choose **Yes** to enable the IP version, and then from the IP Version drop-down list, choose **IPv6** to enable IPv6.
 - From the HOP Options Header drop-down list, choose **Yes** to enable hop-by-hop options, and then from the HOH Present drop-down list, choose **Have HOH**.
 - From the HOH Options field, choose **Yes**, and then in the HOH Option Type field, enter **1**.
 - In the HOH Option Length drop-down list, choose **Yes** to enable hop-by-hop length, and then in the HOH Option Length field, enter **8**.
- Step 11** Configure Event Counter:
- In the Event Count field, enter the number of events you want counted (1 to 65535).
 - From the Event Count Key drop-down list, choose the key you want to use.
 - From the Specify Alert Interface drop-down list, choose whether you want to specify the alert interval (Yes or No).
 - If you chose Yes, enter the alert interval (2 to 1000) in the Alert Interval field.
- Step 12** Configure the alert frequency.
- Step 13** Leave the default (**Yes**) for the Enabled field.



Note A signature must be enabled for the sensor to actively detect the attack specified by the signature.

Step 14 Leave the default (**Yes**) for the Retired field. This places the signature in the engine.



Note A signature must not be retired for the sensor to actively detect the attack specified by the signature.

Step 15 From the Vulnerable OS List drop-down list, choose the operating systems that are vulnerable to this signature.



Tip To choose more than one action, hold down the **Ctrl** key.

Step 16 From the Mars Category drop-down list, choose the MARS categories you want this signature to identify.



Tip To choose more than one action, hold down the **Ctrl** key.



Tip To discard your changes and close the Add Signature dialog box, click **Cancel**.

Step 17 Click **OK**. The new signature appears in the list with the Type set to Custom.



Tip To discard your changes, click **Reset**.

Step 18 Click **Apply** to apply your changes and save the revised configuration.

For More Information

For more information on the Atomic IP engines, see [Atomic Engine, page B-13](#).

Example String XL TCP Match Offset Signature



Caution A custom signature can affect the performance of your sensor. Test the custom signature against a baseline sensor performance for your network to determine the overall impact of the signature.



Note This procedure also applies to String XLUDP and String XL ICMP signatures, with the exception of the parameter **service-ports**, which does not apply to String XL ICMP signatures.

You can create a custom String XL TCP signature by either cloning an existing String XL TCP signature and then tuning it, or by adding a new signature and assigning the String XL TCP signature engine to it.

The following example demonstrates how to create a custom String XL TCP signature that searches for exact, maximum, or minimum offsets. You can modify the following optional match offset parameters for this custom String XLTCP signature:

- Specify Exact Match Offset { Yes | No }—Enables exact match offset:
 - Exact Match Offset—Specifies the exact stream offset in bytes the regular expression string must report for a match to be valid. The valid value is 0 to 65535.
- Specify Maximum Match Offset { Yes | No }—Enables maximum match offset:
 - Maximum Match Offset—Specifies the maximum stream offset in bytes the regular expression string must report for a match to be valid. The valid value is 0 to 65535.
- Specify Min Match Offset { Yes | No }—Enables minimum match offset:
 - Min Match Offset—Specifies the minimum stream offset in bytes the regular expression string must report for a match to be valid. The valid value is 0 to 65535.

To create a custom String XL TCP signature that searches for matches, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**.
- Step 3** To create a custom signature by cloning an existing String XL TCP signature, from the Filter drop-down list, choose Engine, and then String TCP XL from the signature engine drop-down list, highlight the signature you want to clone, and then click **Clone**. Continue with Step 5.
- Step 4** To create a custom signature based on the String XL TCP engine, click **Add** and in the Engine field in the Add Signature dialog box, click **Click to edit** and choose String XL TCP from the drop-down list. Continue with Step 5.
- Step 5** In the Signature ID field, enter a number for the signature. Custom signatures range from 60000 to 65000.
- Step 6** In the Subsignature ID field, enter a number for the signature. The default is 0. You can assign a subsignature ID if you are grouping signatures together that are similar.
- Step 7** (Optional) In the Severity Alert field, choose the severity to be reported by Event Viewer when the sensor sends an alert. The default is Medium.
- Step 8** (Optional) In the Sig Fidelity Rating field, enter a value. The signature fidelity rating is a valid value between 0 and 100 that indicates your confidence in the signature, with 100 being the most confident. The default is 75.
- Step 9** In the Promiscuous Delta field, enter a value. The promiscuous delta is a value used to determine the seriousness of the alert. The valid range is 0 to 30. The default is 0.



Caution

We recommend that you do NOT change the promiscuous delta setting for a signature.

- Step 10** Under Sig Description specify the attributes that uniquely identify this signature:
- a. (Optional) In the Signature Name field, enter a name for the signature. A default name, My Sig, appears in the Signature Name field. Change it to a name that is more specific for your custom signature.



Note The signature name, along with the signature ID and subsignature ID, is reported to Event Viewer when an alert is generated.

- b. (Optional) In the Alert Notes field, enter text to be added to the alert. You can add text to be included in alerts associated with this signature. These notes are reported to Event Viewer when an alert is generated. The default is My Sig Info.
- c. (Optional) In the User Comments field, enter text that describes this signature. You can add any text that you find useful here. This field does not affect the signature or alert in any way. The default is Sig Comment.

Step 11 Under Engine, assign the engine-specific parameters:

- a. (Optional) In the Event Action field, assign the event actions you want the signature to report. The default is Produce Alert. You can assign more actions, such as deny or block, based on your security policy.



Tip To select more than one action, hold down the **Ctrl** key.

- b. (Optional) In the Strip Telnet Options field, choose **Yes** from the drop-down list to strip the Telnet option characters from the data before the pattern is searched.
- c. From the Direction drop-down list, choose the direction of the traffic:
 - From Service—Traffic from service port destined to client port.
 - To Service—Traffic from client port destined to service port.
- d. In the Service Ports field, enter the port number, for example, 80. The value is a comma-separated list of ports or port ranges where the target service resides.

Step 12 To specify the regular expression, in the Specify Raw Regex String, choose **No** from the drop-down list.



Note Raw Regex is regular expression syntax used for raw mode processing. It is expert mode only and targeted for use by the Cisco IPS signature development team or only those who are under supervision by the Cisco IPS signature development team. You can configure a String XL signature in either regular Regex or raw Regex.

- a. In the Regex String field, enter the string this signature will be looking for in the TCP packet, for example, tcpstring.
- b. (Optional) In the Specify Minimum Match Length field, choose **Yes** from the drop-down list to enable minimum match length, and then in the Minimum Match Length field, enter the minimum number of bytes the regular expression string must match (0 to 65535).
- c. (Optional) In the Swap Attacker Victim field, choose **Yes** from the drop-down list to swap the attacker and victim addresses and ports (source and destination) in the alert message and for any actions taken.

Step 13 (Optional) In the Specify Exact Match Offset field, choose **Yes** from the drop-down list to enable exact match offset, and then in the Exact Match Offset field, enter the exact offset the regular expression must trigger to be considered a match for this signature (0 to 65535).



Note If you have exact match offset set to Yes, you cannot configure maximum or minimum match offset. If you have exact match offset set to No, you can configure both maximum and minimum match offset at the same time.

- Step 14** (Optional) In the Specify Max Match Offset field, choose **Yes** from the drop-down list to enable maximum match offset, and then in the Specify Max Match Offset field, enter the maximum offset at which the regular expression must trigger to be considered a match for this signature (0 to 65535).
- Step 15** (Optional) In the Specify Min Match Offset field, choose **Yes** from the drop-down list to enable minimum match offset, and then in the Specify Min Match Offset field, enter the minimum offset at which the regular expression must trigger to be considered a match for this signature (0 to 65535).
- Step 16** (Optional) Under Alert Frequency, you can change the default alert frequency.
- Step 17** Click **OK** to create your custom signature. The signature you created is enabled and added to the list of signatures.



Tip To discard your changes, click **Cancel**.

For More Information

- For more information about the String XL engine, see [String XL Engines, page B-63](#).
- For information about regular expression syntax, see [Regular Expression Syntax, page B-9](#).

Example String XL TCP Engine Minimum Match Length Signature



Caution A custom signature can affect the performance of your sensor. Test the custom signature against a baseline sensor performance for your network to determine the overall impact of the signature.



Note This procedure also applies to String XLUDP and String XL ICMP signatures, with the exception of the parameter **service-ports**, which does not apply to String XL ICMP signatures.

You can create a custom String XL TCP signature by either cloning an existing String XL TCP signature and then tuning it, or by adding a new signature and assigning the String XL TCP signature engine to it.

You can configure the following options to work with a specific Regex string:

- Dot All { Yes | No }—If set to Yes, matches [\x00-\xFF] including \n; if set to No, matches anything in the range [\x00-\xFF] except \n. No is the default.
- End Optional { Yes | No }—Specifies that at the end of a packet, if all other conditions are satisfied but the end is not seen, a match is reported if the minimum is exceeded
- No Case { Yes | No }—Specifies to treat all alphabetic characters in the expression as case insensitive.
- Stingy { Yes | No }—Specifies to stop looking for larger matches after the first completed match.



Note Stingy can only be used with Min Match Length; otherwise, it is ignored

- UTF-8 {True | False}—Treats all legal UTF-8 byte sequences in the expression as a single character.

The following example demonstrates how to create a custom String XL TCP signature that searches for minimum match length with stingy, dot all, and UTF-8 turned on. To create a custom signature based on the String XL TCP engine that searches for minimum match length with stingy, dot all, and UTF-8 turned on, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**.
- Step 3** To create a custom signature by cloning an existing String XL TCP signature, from the Filter drop-down list, choose Engine, and then String TCP XL from the signature engine drop-down list, then highlight the signature you want to clone, and click **Clone**. Continue with Step 5.
- Step 4** To create a custom signature based on the String XL TCP engine, click **Add** and in the Engine field in the Add Signature dialog box, click **Click to edit** and choose String XL TCP from the drop-down list. Continue with Step 5.
- Step 5** In the Signature ID field, enter a number for the signature. Custom signatures range from 60000 to 65000.
- Step 6** In the Subsignature ID field, enter a number for the signature. The default is 0. You can assign a subsignature ID if you are grouping signatures together that are similar.
- Step 7** (Optional) In the Severity Alert field, choose the severity to be reported by Event Viewer when the sensor sends an alert. The default is Medium.
- Step 8** (Optional) In the Sig Fidelity Rating field, enter a value. The signature fidelity rating is a valid value between 0 and 100 that indicates your confidence in the signature, with 100 being the most confident. The default is 75.
- Step 9** In the Promiscuous Delta field, enter a value. The promiscuous delta is a value used to determine the seriousness of the alert. The valid range is 0 to 30. The default is 0.



Caution We recommend that you do NOT change the promiscuous delta setting for a signature.

- Step 10** Under Sig Description specify the attributes that uniquely identify this signature:
- (Optional) In the Signature Name field, enter a name for the signature. A default name, My Sig, appears in the Signature Name field. Change it to a name that is more specific for your custom signature.



Note The signature name, along with the signature ID and subsignature ID, is reported to Event Viewer when an alert is generated.

- (Optional) In the Alert Notes field, enter text to be added to the alert. You can add text to be included in alerts associated with this signature. These notes are reported to Event Viewer when an alert is generated. The default is My Sig Info.
- (Optional) In the User Comments field, enter text that describes this signature. You can add any text that you find useful here. This field does not affect the signature or alert in any way. The default is Sig Comment.

Step 11 Under Engine, assign the engine-specific parameters:

- a. (Optional) In the Event Action field, assign the event actions you want the signature to report. The default is Produce Alert. You can assign more actions, such as deny or block, based on your security policy.



Tip To select more than one action, hold down the **Ctrl** key.

- b. (Optional) In the Strip Telnet Options field, choose **Yes** from the drop-down list to strip the Telnet option characters from the data before the pattern is searched.
- c. From the Direction drop-down list, choose the direction of the traffic:
 - From Service—Traffic from service port destined to client port.
 - To Service—Traffic from client port destined to service port.
- d. In the Service Ports field, enter the port number, for example, 23. The value is a comma-separated list of ports or port ranges where the target service resides.

Step 12 To specify the regular expression, in the Specify Raw Regex String, choose **No** from the drop-down list.



Note Raw Regex is regular expression syntax used for raw mode processing. It is expert mode only and targeted for use by the Cisco IPS signature development team or only those who are under supervision by the Cisco IPS signature development team. You can configure a String XL signature in either regular Regex or raw Regex.

- a. In the Regex String field, enter the string this signature will be looking for in the TCP packet (for example, ht+p[*r*].).
- b. (Optional) In the Specify Minimum Match Length field, choose **Yes** from the drop-down list to enable minimum match length, and then in the Minimum Match Length field, enter the minimum number of bytes the regular expression string must match (0 to 65535).
- c. (Optional) In the Swap Attacker Victim field, choose **Yes** from the drop-down list to swap the attacker and victim addresses and ports (source and destination) in the alert message and for any actions taken.

Step 13 (Optional) Turn on the following options by choosing **Yes** in the drop-down list:

- Dot All
- Stingy
- UTF-8

Step 14 (Optional) Under Alert Frequency, you can change the default alert frequency.

Step 15 Click **OK** to create your custom signature. The signature you created is enabled and added to the list of signatures.



Tip To discard your changes, click **Cancel**.

For More Information

- For more information about the String XL engine, see [String XL Engines, page B-63](#).
- For information about regular expression syntax, see [Regular Expression Syntax, page B-9](#).

Configuring Signature Variables

This section describes how to configure signature variables, and contains the following topics:

- [Signature Variables Tab, page 10-38](#)
- [Signature Variables Field Definitions, page 10-38](#)
- [Adding, Editing, and Deleting Signature Variables, page 10-39](#)

Signature Variables Tab

**Note**

You must be administrator or operator to configure signature variables.

When you want to use the same value within multiple signatures, use a variable. When you change the value of a variable, that variable is updated in all signatures in which it appears. This saves you from having to change the variable repeatedly as you configure signatures.

**Note**

You must preface the variable with a dollar (\$) sign to indicate that you are using a variable rather than a string.

Some variables cannot be deleted because they are necessary to the signature system. If a variable is protected, you cannot select it to edit it. You receive an error message if you try to delete protected variables. You can edit only one variable at a time.

Signature Variables Field Definitions

The following fields are found on the Signature Variables tab and in the Add and Edit Signature Variable dialog boxes:






- **Name**—Identifies the name assigned to this variable.
- **Type**—Identifies the variable as a web port or IP address range.
- **Value**—Identifies the value(s) represented by this variable.

**Note**

To designate multiple port numbers for a single variable, place a comma between the entries. For example, 80, 3128, 8000, 8010, 8080, 8888, 24326.

Adding, Editing, and Deleting Signature Variables

To add, edit, and delete signature variables, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures > Advanced > Signature Variables**, and then click **Add** to create a variable.
- Step 3** In the Name field, enter the name of the signature variable. From the Type drop-down list, choose the type of signature variable.
-  **Note** A valid name can only contain numbers or letters. You can also use a hyphen (-) or underscore (_).
-
- Step 4** In the Value field, enter the value for the new signature variable. The web-ports type has a predefined set of ports where web servers are running, but you can edit the value. This variable affects all signatures that have web ports. The default is 80, 3128, 8000, 8010, 8080, 8888, 24326.
-  **Note** You can use commas as delimiters. Make sure there are no trailing spaces after the comma. Otherwise, you receive a `Validation failed` error.
-
-  **Tip** To discard your changes and close the Add Signature Variable dialog box, click **Cancel**.
-
- Step 5** Click **OK**. The new variable appears in the signature variables list on the Signature Variables tab.
- Step 6** To edit an existing variable, select it in the signature variables list, and then click **Edit**.
- Step 7** Make any changes needed in the Value field.
-  **Tip** To discard your changes and close the Edit Signature Variable dialog box, click **Cancel**.
-
- Step 8** Click **OK**. The edited variable appears in the signature variables list on the Signature Variables tab.
- Step 9** To delete a variable, select it in the signature variables list, and then click **Delete**. The variable no longer appears in the signature variables list on the Signature Variables tab.
-  **Tip** To discard your changes, click **Reset**.
-
- Step 10** Click **Apply** to apply your changes and save the revised configuration.
-

Configuring Miscellaneous Settings

This section describes the Miscellaneous tab and how to configure Application Inspection and Control (AIC) signatures, IP fragment reassembly signatures, TCP stream reassembly signatures, and IP logging. It contains the following topics:

- [Miscellaneous Tab, page 10-40](#)
- [Miscellaneous Tab Field Definitions, page 10-41](#)
- [Configuring Application Policy Signatures, page 10-42](#)
- [Configuring IP Fragment Reassembly Signatures, page 10-51](#)
- [Configuring TCP Stream Reassembly Signatures, page 10-54](#)
- [Configuring IP Logging, page 10-61](#)

Miscellaneous Tab

**Note**

You must be administrator or operator to configure the parameters on the Miscellaneous tab.

On the Miscellaneous tab, you can perform the following tasks:

- Configure the application policy parameters (also known as AIC signatures)
You can configure the sensor to provide Layer 4 to Layer 7 packet inspection to prevent malicious attacks related to web services. You first set up the AIC parameters, then you can either use the default AIC signatures or tune them.
- Configure IP fragment reassembly options
You can configure the sensor to reassemble a datagram that has been fragmented over multiple packets. You can specify boundaries that the sensor uses to determine how many datagrams and how long to wait for more fragments of a datagram. The goal is to ensure that the sensor does not allocate all its resources to datagrams that cannot be completely reassembled, either because the sensor missed some frame transmissions or because an attack has been launched that is based on generating random fragment datagrams. You first choose the method the sensor will use to perform IP fragment reassembly, then you can tune the IP fragment reassembly signatures, which are part of the Normalizer engine.
- Configure TCP stream reassembly
You can configure the sensor to monitor only TCP sessions that have been established by a complete three-way handshake. You can also configure how long to wait for the handshake to complete, and how long to keep monitoring a connection where no more packets have been seen. The goal is to prevent the sensor from creating alerts where a valid TCP session has not been established. There are known attacks against sensors that try to get the sensor to generate alerts by simply replaying pieces of an attack. The TCP session reassembly feature helps to mitigate these types of attacks against the sensor. You first choose the method the sensor will use to perform TCP stream reassembly, then you can tune TCP stream reassembly signatures, which are part of the Normalizer engine.



Note For signature 3050 Half Open SYN Attack, if you choose Modify Packet Inline as the action, you can see as much as 20 to 30% performance degradation while the protection is active. The protection is only active during an actual SYN flood.

- Configure IP logging options

You can configure a sensor to generate an IP session log when the sensor detects an attack. When IP logging is configured as a response action for a signature and the signature is triggered, all packets to and from the source address of the alert are logged for a specified period of time.

For More Information

- For the procedure for setting up the AIC parameters, see [Configuring Application Policy, page 10-49](#).
- For an example procedure for tuning AIC signatures, see [Tuning an AIC Signature, page 10-50](#).
- For the procedure for configuring the method the sensor uses for performing IP fragment reassembly, see [Configuring the IP Fragment Reassembly Mode, page 10-53](#).
- For an example procedure for tuning an IP fragment reassembly signature, see [Tuning an IP Fragment Reassembly Signature, page 10-53](#).
- For the procedure for configuring the method the sensor uses to perform TCP stream reassembly, see [Configuring the TCP Stream Reassembly Mode, page 10-59](#).
- For an example procedure for tuning a TCP stream reassembly signature, see [Tuning a TCP Stream Reassembly Signature, page 10-60](#).
- For the procedure for configuring IP logging, see [Configuring IP Logging, page 10-61](#).

Miscellaneous Tab Field Definitions

The following fields and buttons are found on the Miscellaneous tab:

- Application Policy—Lets you configure application policy enforcement:
 - Enable HTTP —Enables protection for web services. Check the **Yes** check box to require the sensor to inspect HTTP traffic for compliance with the RFC.
 - Max HTTP Requests—Specifies the maximum number of outstanding HTTP requests per connection.
 - AIC Web Ports—Specifies the variable for ports to look for AIC traffic.
 - Enable FTP—Enables protection for web services. Check the **Yes** check box to require the sensor to inspect FTP traffic.
- Fragment Reassembly—Lets you configure the mode for IP fragment reassembly:
 - IP Reassembly Mode—Identifies the method the sensor uses to reassemble the fragments, based on the operating system.
- Stream Reassembly—Lets you configure the mode for TCP stream reassembly:
 - TCP Handshake Required—Specifies that the sensor should only track sessions for which the three-way handshake is completed.
 - TCP Reassembly Mode—Specifies the mode the sensor should use to reassemble TCP sessions with the following options:

- Asymmetric—Can only see one direction of bidirectional traffic flow.



Note Asymmetric mode lets the sensor synchronize state with the flow and maintain inspection for those engines that do not require both directions. Asymmetric mode lowers security because full protection requires both sides of traffic to be seen.

- Strict—If a packet is missed for any reason, all packets after the missed packet are not processed.
- Loose—Use in environments where packets might be dropped.
- IP Log—Lets you configure the sensor to stop an in-progress IP Log when any one of the following thresholds is reached:
 - Max IP Log Packets—Identifies the maximum number of packets per log. Valid values are 0 to 65535 packets. The default is 0 packets.
 - IP Log Time—Identifies the duration in seconds to log. Valid values are 30 to 300 seconds. The default is 30 seconds.
 - Max IP Log Bytes—Identifies the maximum size in bytes per log. Valid values are 0 to 2147483647 bytes. The default is 0 bytes.

Configuring Application Policy Signatures

This section describes Application Policy (AIC) signatures and how to configure them. It contains the following topics:

- [Understanding AIC Signatures, page 10-42](#)
- [AIC Engine and Sensor Performance, page 10-43](#)
- [AIC Request Method Signatures, page 10-44](#)
- [AIC MIME Define Content Type Signatures, page 10-45](#)
- [AIC Transfer Encoding Signatures, page 10-48](#)
- [AIC FTP Commands Signatures, page 10-48](#)
- [Configuring Application Policy, page 10-49](#)
- [Tuning an AIC Signature, page 10-50](#)

Understanding AIC Signatures

AIC has the following categories of signatures:

- HTTP request method
 - Define request method
 - Recognized request methods
- MIME type
 - Define content type
 - Recognized content type

- Define web traffic policy

There is one predefined signature, 12674, that specifies the action to take when noncompliant HTTP traffic is seen. The parameter Alarm on Non HTTP Traffic enables the signature. By default this signature is enabled.
- Transfer encodings
 - Associate an action with each method
 - List methods recognized by the sensor
 - Specify which actions need to be taken when a chunked encoding error is seen
- FTP commands
 - Associates an action with an FTP command.

For More Information

- For more information on the AIC engine, see [AIC Engine, page B-10](#).
- For a list of AIC request method signature IDs and descriptions, see [AIC Request Method Signatures, page 10-44](#).
- For a list of AIC MIME define content type signature IDs and descriptions, see [AIC MIME Define Content Type Signatures, page 10-45](#).
- For a list of AIC transfer encoding signature IDs and descriptions, see [AIC Transfer Encoding Signatures, page 10-48](#).
- For a list of AIC FTP commands signature IDs and descriptions, see [AIC FTP Commands Signatures, page 10-48](#).

AIC Engine and Sensor Performance

Application policy enforcement is a unique sensor feature. Rather than being based on traditional IPS technologies that inspect for exploits, vulnerabilities, and anomalies, AIC policy enforcement is designed to enforce HTTP and FTP service policies. The inspection work required for this policy enforcement is extreme compared with traditional IPS inspection work. A large performance penalty is associated with using this feature. When AIC is enabled, the overall bandwidth capacity of the sensor is reduced.

AIC policy enforcement is disabled in the IPS default configuration. If you want to activate AIC policy enforcement, we highly recommend that you carefully choose the exact policies of interest and disable those you do not need. Also, if your sensor is near its maximum inspection load capacity, we recommend that you not use this feature since it can oversubscribe the sensor. We recommend that you use the adaptive security appliance firewall to handle this type of policy enforcement.

For More Information

For more information on the AIC signature engine, see [AIC Engine, page B-10](#).

AIC Request Method Signatures

The HTTP request method has two categories of signatures:

- Define request method—Allows actions to be associated with request methods. You can expand and modify the signatures (Define Request Method).
- Recognized request methods—Lists methods that are recognized by the sensor (Recognized Request Methods).

Table 10-1 lists the predefined define request method signatures. Enable the signatures that have the predefined method you need.

Table 10-1 Request Method Signatures

| Signature ID | Define Request Method |
|--------------|-------------------------------------|
| 12676 | Request Method Not Recognized |
| 12677 | Define Request Method PUT |
| 12678 | Define Request Method CONNECT |
| 12679 | Define Request Method DELETE |
| 12680 | Define Request Method GET |
| 12681 | Define Request Method HEAD |
| 12682 | Define Request Method OPTIONS |
| 12683 | Define Request Method POST |
| 12685 | Define Request Method TRACE |
| 12695 | Define Request Method INDEX |
| 12696 | Define Request Method MOVE |
| 12697 | Define Request Method MKDIR |
| 12698 | Define Request Method COPY |
| 12699 | Define Request Method EDIT |
| 12700 | Define Request Method UNEDIT |
| 12701 | Define Request Method SAVE |
| 12702 | Define Request Method LOCK |
| 12703 | Define Request Method UNLOCK |
| 12704 | Define Request Method REVLABEL |
| 12705 | Define Request Method REVLOG |
| 12706 | Define Request Method REVADD |
| 12707 | Define Request Method REVNUM |
| 12708 | Define Request Method SETATTRIBUTE |
| 12709 | Define Request Method GETATTRIBUTE |
| 12710 | Define Request Method GETPROPERTIES |
| 12711 | Define Request Method STARTENV |
| 12712 | Define Request Method STOPREV |

For More Information

For the procedure for enabling signatures, see [Enabling, Disabling, and Retiring Signatures](#), page 10-19.

AIC MIME Define Content Type Signatures

There are two policies associated with MIME types:

- Define content type—Associates specific actions for the following cases (Define Content Type):
 - Deny a specific MIME type, such as an image/jpeg
 - Message size violation
 - MIME-type mentioned in header and body do not match
- Recognized content type (Recognized Content Type)

[Table 10-2](#) lists the predefined define content type signatures. Enable the signatures that have the predefined content type you need. You can also create custom define content type signatures.

Table 10-2 Define Content Type Signatures

| Signature ID | Signature Description |
|--------------|--|
| 12621 | Content Type image/gif Invalid Message Length |
| 12622 2 | Content Type image/png Verification Failed |
| 12623 0 | Content Type image/tiff Header Check |
| 12623 1 | Content Type image/tiff Invalid Message Length |
| 12623 2 | Content Type image/tiff Verification Failed |
| 12624 0 | Content Type image/x-3ds Header Check |
| 12624 1 | Content Type image/x-3ds Invalid Message Length |
| 12624 2 | Content Type image/x-3ds Verification Failed |
| 12626 0 | Content Type image/x-portable-bitmap Header Check |
| 12626 1 | Content Type image/x-portable-bitmap Invalid Message Length |
| 12626 2 | Content Type image/x-portable-bitmap Verification Failed |
| 12627 0 | Content Type image/x-portable-graymap Header Check |
| 12627 1 | Content Type image/x-portable-graymap Invalid Message Length |
| 12627 2 | Content Type image/x-portable-graymap Verification Failed |
| 12628 0 | Content Type image/jpeg Header Check |
| 12628 1 | Content Type image/jpeg Invalid Message Length |
| 12628 2 | Content Type image/jpeg Verification Failed |
| 12629 0 | Content Type image/cgf Header Check |
| 12629 1 | Content Type image/cgf Invalid Message Length |
| 12631 0 | Content Type image/x-xpm Header Check |
| 12631 1 | Content Type image/x-xpm Invalid Message Length |
| 12633 0 | Content Type audio/midi Header Check |
| 12633 1 | Content Type audio/midi Invalid Message Length |
| 12633 2 | Content Type audio/midi Verification Failed |
| 12634 0 | Content Type audio/basic Header Check |
| 12634 1 | Content Type audio/basic Invalid Message Length |
| 12634 2 | Content Type audio/basic Verification Failed |

Table 10-2 Define Content Type Signatures (continued)

| Signature ID | Signature Description |
|---------------------|---|
| 12635 0 | Content Type audio/mpeg Header Check |
| 12635 1 | Content Type audio/mpeg Invalid Message Length |
| 12635 2 | Content Type audio/mpeg Verification Failed |
| 12636 0 | Content Type audio/x-adpcm Header Check |
| 12636 1 | Content Type audio/x-adpcm Invalid Message Length |
| 12636 2 | Content Type audio/x-adpcm Verification Failed |
| 12637 0 | Content Type audio/x-aiff Header Check |
| 12637 1 | Content Type audio/x-aiff Invalid Message Length |
| 12637 2 | Content Type audio/x-aiff Verification Failed |
| 12638 0 | Content Type audio/x-ogg Header Check |
| 12638 1 | Content Type audio/x-ogg Invalid Message Length |
| 12638 2 | Content Type audio/x-ogg Verification Failed |
| 12639 0 | Content Type audio/x-wav Header Check |
| 12639 1 | Content Type audio/x-wav Invalid Message Length |
| 12639 2 | Content Type audio/x-wav Verification Failed |
| 12641 0 | Content Type text/html Header Check |
| 12641 1 | Content Type text/html Invalid Message Length |
| 12641 2 | Content Type text/html Verification Failed |
| 12642 0 | Content Type text/css Header Check |
| 12642 1 | Content Type text/css Invalid Message Length |
| 12643 0 | Content Type text/plain Header Check |
| 12643 1 | Content Type text/plain Invalid Message Length |
| 12644 0 | Content Type text/richtext Header Check |
| 12644 1 | Content Type text/richtext Invalid Message Length |
| 12645 0 | Content Type text/sgml Header Check |
| 12645 1 | Content Type text/sgml Invalid Message Length |
| 12645 2 | Content Type text/sgml Verification Failed |
| 12646 0 | Content Type text/xml Header Check |
| 12646 1 | Content Type text/xml Invalid Message Length |
| 12646 2 | Content Type text/xml Verification Failed |
| 12648 0 | Content Type video/flc Header Check |
| 12648 1 | Content Type video/flc Invalid Message Length |
| 12648 2 | Content Type video/flc Verification Failed |
| 12649 0 | Content Type video/mpeg Header Check |
| 12649 1 | Content Type video/mpeg Invalid Message Length |
| 12649 2 | Content Type video/mpeg Verification Failed |
| 12650 0 | Content Type text/xmcd Header Check |
| 12650 1 | Content Type text/xmcd Invalid Message Length |
| 12651 0 | Content Type video/quicktime Header Check |
| 12651 1 | Content Type video/quicktime Invalid Message Length |
| 12651 2 | Content Type video/quicktime Verification Failed |
| 12652 0 | Content Type video/sgi Header Check |
| 12652 1 | Content Type video/sgi Verification Failed |

Table 10-2 Define Content Type Signatures (continued)

| Signature ID | Signature Description |
|--------------|---|
| 12653 0 | Content Type video/x-avi Header Check |
| 12653 1 | Content Type video/x-avi Invalid Message Length |
| 12654 0 | Content Type video/x-fli Header Check |
| 12654 1 | Content Type video/x-fli Invalid Message Length |
| 12654 2 | Content Type video/x-fli Verification Failed |
| 12655 0 | Content Type video/x-mng Header Check |
| 12655 1 | Content Type video/x-mng Invalid Message Length |
| 12655 2 | Content Type video/x-mng Verification Failed |
| 12656 0 | Content Type application/x-msvideo Header Check |
| 12656 1 | Content Type application/x-msvideo Invalid Message Length |
| 12656 2 | Content Type application/x-msvideo Verification Failed |
| 12658 0 | Content Type application/ms-word Header Check |
| 12658 1 | Content Type application/ms-word Invalid Message Length |
| 12659 0 | Content Type application/octet-stream Header Check |
| 12659 1 | Content Type application/octet-stream Invalid Message Length |
| 12660 0 | Content Type application/postscript Header Check |
| 12660 1 | Content Type application/postscript Invalid Message Length |
| 12660 2 | Content Type application/postscript Verification Failed |
| 12661 0 | Content Type application/vnd.ms-excel Header Check |
| 12661 1 | Content Type application/vnd.ms-excel Invalid Message Length |
| 12662 0 | Content Type application/vnd.ms-powerpoint Header Check |
| 12662 1 | Content Type application/vnd.ms-powerpoint Invalid Message Length |
| 12663 0 | Content Type application/zip Header Check |
| 12663 1 | Content Type application/zip Invalid Message Length |
| 12663 2 | Content Type application/zip Verification Failed |
| 12664 0 | Content Type application/x-gzip Header Check |
| 12664 1 | Content Type application/x-gzip Invalid Message Length |
| 12664 2 | Content Type application/x-gzip Verification Failed |
| 12665 0 | Content Type application/x-java-archive Header Check |
| 12665 1 | Content Type application/x-java-archive Invalid Message Length |
| 12666 0 | Content Type application/x-java-vm Header Check |
| 12666 1 | Content Type application/x-java-vm Invalid Message Length |
| 12667 0 | Content Type application/pdf Header Check |
| 12667 1 | Content Type application/pdf Invalid Message Length |
| 12667 2 | Content Type application/pdf Verification Failed |
| 12668 0 | Content Type unknown Header Check |
| 12668 1 | Content Type unknown Invalid Message Length |
| 12669 0 | Content Type image/x-bitmap Header Check |
| 12669 1 | Content Type image/x-bitmap Invalid Message Length |
| 12673 0 | Recognized content type |

For More Information

For the procedure for enabling signatures, see [Enabling, Disabling, and Retiring Signatures, page 10-19](#)

AIC Transfer Encoding Signatures

There are three policies associated with transfer encoding:

- Associate an action with each method (Define Transfer Encoding)
- List methods recognized by the sensor (Recognized Transfer Encodings)
- Specify which actions need to be taken when a chunked encoding error is seen (Chunked Transfer Encoding Error)

[Table 10-3](#) lists the predefined transfer encoding signatures. Enable the signatures that have the predefined transfer encoding method you need.

Table 10-3 *Transfer Encoding Signatures*

| Signature ID | Transfer Encoding Method |
|--------------|-----------------------------------|
| 12686 | Recognized Transfer Encoding |
| 12687 | Define Transfer Encoding Deflate |
| 12688 | Define Transfer Encoding Identity |
| 12689 | Define Transfer Encoding Compress |
| 12690 | Define Transfer Encoding GZIP |
| 12693 | Define Transfer Encoding Chunked |
| 12694 | Chunked Transfer Encoding Error |

For More Information

For the procedure for enabling signatures, see [Enabling, Disabling, and Retiring Signatures, page 10-19](#)

AIC FTP Commands Signatures

[Table 10-4](#) lists the predefined FTP commands signatures. Enable the signatures that have the predefined FTP command you need.

Table 10-4 *FTP Commands Signatures*

| Signature ID | FTP Command |
|--------------|--------------------------|
| 12900 | Unrecognized FTP command |
| 12901 | Define FTP command abor |
| 12902 | Define FTP command acct |
| 12903 | Define FTP command allo |
| 12904 | Define FTP command appe |
| 12905 | Define FTP command cdup |
| 12906 | Define FTP command cwd |
| 12907 | Define FTP command dele |

Table 10-4 FTP Commands Signatures (continued)

| Signature ID | FTP Command |
|--------------|-------------------------|
| 12908 | Define FTP command help |
| 12909 | Define FTP command list |
| 12910 | Define FTP command mkd |
| 12911 | Define FTP command mode |
| 12912 | Define FTP command nlst |
| 12913 | Define FTP command noop |
| 12914 | Define FTP command pass |
| 12915 | Define FTP command pasv |
| 12916 | Define FTP command port |
| 12917 | Define FTP command pwd |
| 12918 | Define FTP command quit |
| 12919 | Define FTP command rein |
| 12920 | Define FTP command rest |
| 12921 | Define FTP command retr |
| 12922 | Define FTP command rmd |
| 12923 | Define FTP command rnfr |
| 12924 | Define FTP command rnto |
| 12925 | Define FTP command site |
| 12926 | Define FTP command smnt |
| 12927 | Define FTP command stat |
| 12928 | Define FTP command stor |
| 12929 | Define FTP command stou |
| 12930 | Define FTP command stru |
| 12931 | Define FTP command syst |
| 12932 | Define FTP command type |
| 12933 | Define FTP command user |

For More Information

For the procedure for enabling signatures, see [Enabling, Disabling, and Retiring Signatures](#), page 10-19

Configuring Application Policy**Tip**

An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.

To configure the application policy parameters, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
 - Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures > Advanced > Miscellaneous**.
 - Step 3** In the Enable HTTP field, choose **Yes** from the drop-down list to enable inspection of HTTP traffic.
 - Step 4** In the Max HTTP Requests field, enter the number of outstanding HTTP requests per connection that can be outstanding without having received a response from the server.
 - Step 5** In the AIC Web Ports field, enter the ports that you want to be active.
 - Step 6** In the Enable FTP field choose **Yes** from the drop-down list to enable inspection of FTP traffic.



Note If you enable the application policy for HTTP or FTP, the sensor checks to be sure the traffic is compliant with the RFC.



Tip To discard your changes, click **Cancel**.

- Step 7** Click **OK**.



Tip To discard your changes, click **Reset**.

- Step 8** Click **Apply** to apply your changes and save the revised configuration.
-

Tuning an AIC Signature



Tip An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.

The following example demonstrates how to tune an AIC signature, a Recognized Content Type (MIME) signature, specifically, signature 12,623 1 Content Type image/tiff Invalid Message Length.

To tune a MIME-type policy signature, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
 - Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**.
 - Step 3** From the Filter drop-down list, choose **Engine** and then choose **AIC HTTP** as the engine.
 - Step 4** Scroll down the list and select Sig ID 12,623 Subsig ID 1 Content Type image/tiff Invalid Message Length, and click **Edit**.



Tip You can click the Sig ID column head to have the signature IDs appear in order.

Step 5 Under Status, choose **Yes** from the drop-down list in the Enabled field.

Step 6 Under Engine, choose one of the options, for example, **Length**, in the Content Type Details field.

Step 7 In the Length field, make the length smaller by changing the default to 30,000.



Tip To discard your changes and close the Edit Signature dialog box, click **Cancel**.

Step 8 Click **OK**, and then click **Apply** to save your changes.



Tip To discard your changes, click **Reset**.

Configuring IP Fragment Reassembly Signatures

This section describes IP fragment reassembly, lists the IP fragment reassembly signatures with their configurable parameters, and describes how to configure them. It contains the following topics:

- [Understanding IP Fragment Reassembly Signatures, page 10-51](#)
- [IP Fragment Reassembly Signatures and Configurable Parameters, page 10-52](#)
- [Configuring the IP Fragment Reassembly Mode, page 10-53](#)
- [Tuning an IP Fragment Reassembly Signature, page 10-53](#)

Understanding IP Fragment Reassembly Signatures

You can configure the sensor to reassemble a datagram that has been fragmented over multiple packets. You can specify boundaries that the sensor uses to determine how many datagram fragments it reassembles and how long to wait for more fragments of a datagram. The goal is to ensure that the sensor does not allocate all its resources to datagrams that cannot be completely reassembled, either because the sensor missed some frame transmissions or because an attack has been launched that is based on generating random fragmented datagrams.



Note You configure the IP fragment reassembly per signature.

For More Information

For more information on the Normalizer engine, see [Normalizer Engine, page B-35](#).

IP Fragment Reassembly Signatures and Configurable Parameters

Table 10-5 lists IP fragment reassembly signatures with the parameters that you can configure for IP fragment reassembly. The IP fragment reassembly signatures are part of the Normalizer engine.

Table 10-5 IP Fragment Reassembly Signatures

| Signature ID and Name | Description | Parameter With Default Value and Range | Default Action |
|---|---|---|---|
| 1200 IP Fragmentation Buffer Full | Fires when the total number of fragments in the system exceeds the threshold set by Max Fragments. | Specify Max Fragments 10000 (0-42000) | Deny Packet Inline Produce Alert ¹ |
| 1201 IP Fragment Overlap | Fires when the fragments queued for a datagram overlap each other. | — ² | Deny Packet Inline Produce Alert ¹ |
| 1202 IP Fragment Overrun - Datagram Too Long | Fires when the fragment data (offset and size) exceeds the threshold set with Max Datagram Size. | Specify Max Datagram Size 65536 (2000-65536) | Deny Packet Inline Produce Alert ³ |
| 1203 IP Fragment Overwrite - Data is Overwritten | Fires when the fragments queued for a datagram overlap each other and the overlapping data is different. ⁴ | — | Deny Packet Inline Produce Alert ⁵ |
| 1204 IP Fragment Missing Initial Fragment | Fires when the datagram is incomplete and missing the initial fragment. | — | Deny Packet Inline Produce Alert ⁶ |
| 1205 IP Fragment Too Many Datagrams | Fires when the total number of partial datagrams in the system exceeds the threshold set by Max Partial Datagrams. | Specify Max Partial Datagrams 1000 (0-10000) | Deny Packet Inline Produce Alert ⁷ |
| 1206 IP Fragment Too Small | Fires when there are more than Max Small Frags of a size less than Min Fragment Size in one datagram. ⁸ | Specify Max Small Frags 2 (8-1500) Specify Min Fragment Size 400 (1-8) | Deny Packet Inline Produce Alert ⁹ |
| 1207 IP Fragment Too Many Fragments in a Datagram | Fires when there are more than Max Fragments per Datagram in one datagram. | Specify Max Fragments per Datagram 170 (0-8192) | Deny Packet Inline Produce Alert ⁶ |
| 1208 IP Fragment Incomplete Datagram | Fires when all of the fragments for a datagram have not arrived during the Fragment Reassembly Timeout. ¹⁰ | Specify Fragment Reassembly Timeout 60 (0-360) | Deny Packet Inline Produce Alert ⁶ |
| 1225 Fragment Flags Invalid | Fires when a bad combination of fragment flags is detected. | — ¹¹ | — |

1. Modify Packet Inline and Deny Connection Inline have no effect on this signature. Deny Packet Inline drops the packets and all associated fragments for this datagram. If you disable this signature, the default values are still used and packets are dropped (inline mode) or not analyzed (promiscuous mode) and no alert is sent.
2. This signature does not fire when the datagram is an exact duplicate. Exact duplicates are dropped in inline mode regardless of the settings. Modify Packet Inline removes the overlapped data from all but one fragment so there is no ambiguity about how the endpoint treats the datagram. Deny Connection Inline has no effect on this signature. Deny Packet Inline drops the packet and all associated fragments for this datagram.
3. Modify Packet Inline and Deny Connection Inline have no effect on this signature. Deny Packet Inline drops the packet and all associated fragments for this datagram. Regardless of the actions set the datagram is not processed by the IPS if the datagram is larger than the Max Datagram size.
4. This is a very unusual event.
5. Modify Packet Inline removes the overlapped data from all but one fragment so there is no ambiguity about how the endpoint treats the datagram. Deny Connection Inline has no effect on this signature. Deny Packet Inline drops the packets and all associated fragments for this datagram.

6. IPS does not inspect a datagram missing the first fragments regardless of the settings. Modify Packet Inline and Deny Connection Inline have no effect on this signature. Deny Packet Inline drops the packet and all associated fragments for this datagram.
7. Modify Packet Inline and Deny Connection Inline have no effect on this signature. Deny Packet Inline drops the packet and all associated fragments for this datagram.
8. IPS does not inspect the datagram if this signature is on and the number of small fragments is exceeded.
9. Modify Packet Inline and Deny Connection Inline have no effect on this signature. Deny Packet Inline drops the packet and all associated fragments for this datagram.
10. The timer starts when the packet for the datagram arrives.
11. Modify Packet Inline modifies the flags to a valid combination. Deny Connection Inline has no effect on this signature. Deny Packet Inline drops the packet and all associated fragments for this datagram.

Configuring the IP Fragment Reassembly Mode



Tip

An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.



Note

You can configure the IP fragment reassembly mode if your sensor is operating in promiscuous mode. If your sensor is operating inline mode, the method is NT only.

To configure the mode the sensor uses for IP fragment reassembly, follow these steps:

- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures > Advanced > Miscellaneous**.
- Step 3** Under Fragment Reassembly, from the IP Reassembly Mode field choose the operating system you want to use to reassemble the fragments.



Tip

To discard your changes and close the Advanced dialog box, click **Cancel**.

- Step 4** Click **OK**, and then **Apply** to apply your changes and save the revised configuration



Tip

To discard your changes, click **Reset**.

Tuning an IP Fragment Reassembly Signature



Tip

An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.

The following procedure demonstrates how to tune an IP fragment reassembly signature, specifically, signature 1200 0 IP Fragmentation Buffer Full.

To tune an IP fragment reassembly signature, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**.
- Step 3** In the Filter field, choose **Engine** from the drop-down list, and then choose **Normalizer** as the engine.
- Step 4** Select the IP fragment reassembly signature you want to configure in the list, for example, Sig ID 1200 Subsig ID 0 IP Fragmentation Buffer Full, and then click **Edit**.
- Step 5** Change the default setting of any IP fragment reassembly parameters that can be configured for signature 1200. For example, in the Max Fragments field change the setting from the default of 10000 to 20000. For signature 1200, you can also change the parameters of these options:
- Specify TCP Idle Timeout
 - Specify Service Ports
 - Specify SYN Flood Max Embryonic



Tip To discard your changes and close the Edit Signature dialog box, click **Cancel**.

- Step 6** Click **OK**, and then **Apply** to apply your changes and save the revised configuration



Tip To discard your changes, click **Reset**.

Configuring TCP Stream Reassembly Signatures

This section describes TCP stream reassembly, lists the TCP stream reassembly signatures with the configurable parameters, describes how to configure TCP stream signatures, and how to configure the mode for TCP stream reassembly. This section contains the following topics:

- [Understanding TCP Stream Reassembly Signatures, page 10-54](#)
- [TCP Stream Reassembly Signatures and Configurable Parameters, page 10-55](#)
- [Configuring the TCP Stream Reassembly Mode, page 10-59](#)
- [Tuning a TCP Stream Reassembly Signature, page 10-60](#)

Understanding TCP Stream Reassembly Signatures

You can configure the sensor to monitor only TCP sessions that have been established by a complete three-way handshake. You can also configure how long to wait for the handshake to complete, and how long to keep monitoring a connection where no more packets have been seen. The goal is to prevent the sensor from creating alerts where a valid TCP session has not been established. There are known attacks against sensors that try to get the sensor to generate alerts by simply replaying pieces of an attack. The TCP session reassembly feature helps to mitigate these types of attacks against the sensor.

You configure TCP stream reassembly parameters per signature. You can configure the mode for TCP stream reassembly.

For More Information

For more information on Normalizer engine, see [Normalizer Engine, page B-35](#).

TCP Stream Reassembly Signatures and Configurable Parameters

Table 10-6 lists TCP stream reassembly signatures with the parameters that you can configure for TCP stream reassembly. TCP stream reassembly signatures are part of the Normalizer engine.

Table 10-6 TCP Stream Reassembly Signatures

| Signature ID and Name | Description | Parameter With Default Value and Range | Default Actions |
|--|---|---|-----------------------------------|
| 1301 TCP Session Inactivity Timeout ¹ | Fires when a TCP session has been idle for a TCP Idle Timeout. | TCP Idle Timeout 3600 (15-3600) | — ² |
| 1302 TCP Session Embryonic Timeout ³ | Fires when a TCP session has not completes the three-way handshake in TCP embryonic timeout seconds. | TCP Embryonic Timeout 15 (3-300) | — ⁴ |
| 1303 TCP Session Closing Timeout ⁵ | Fires when a TCP session has not closed completely in TCP Closed Timeout seconds after the first FIN. | TCP Closed Timeout 5 (1-60) | — ⁶ |
| 1304 TCP Session Packet Queue Overflow | This signature allows for setting the internal TCP Max Queue size value for the Normalizer engine. As a result it does not function in promiscuous mode. By default this signature does not fire an alert. If a custom alert event is associated with this signature and if the queue size is exceeded, an alert fires. Note The IPS signature team discourages modifying this value. | TCP Max Queue 32 (0-128) TCP Idle Timeout 3600 | — ⁷ |
| 1305 TCP Urg Flag Set ⁸ | Fires when the TCP urgent flag is seen | TCP Idle Timeout 3600 | Modify Packet Inline ⁹ |

Table 10-6 TCP Stream Reassembly Signatures (continued)

| Signature ID and Name | Description | Parameter With Default Value and Range | Default Actions |
|---|---|--|---|
| 1306 0 TCP Option Other | Fires when a TCP option in the range of TCP Option Number is seen. All 1306 signatures fire an alert and do not function in promiscuous mode. | TCP Option Number 6-7,9-255 (Integer Range Allow Multiple 0-255 constraints) TCP Idle Timeout 3600 | Modify Packet Inline Produce Alert ¹⁰ |
| 1306 1 TCP SACK Allowed Option | Fires when a TCP selective ACK allowed option is seen. All 1306 signatures fire an alert and do not function in promiscuous mode. | TCP Idle Timeout 3600 | Modify Packet Inline ¹¹ |
| 1306 2 TCP SACK Data Option | Fires when a TCP selective ACK data option is seen. All 1306 signatures fire an alert and do not function in promiscuous mode. | TCP Idle Timeout 3600 | Modify Packet Inline ¹² |
| 1306 3 TCP Timestamp Option | Fires when a TCP timestamp option is seen. All 1306 signatures fire an alert and do not function in promiscuous mode. | TCP Idle Timeout 3600 | Modify Packet Inline ¹³ |
| 1306 4 TCP Window Scale Option | Fires when a TCP window scale option is seen. All 1306 signatures fire an alert and do not function in promiscuous mode. | TCP Idle Timeout 3600 | Modify Packet Inline ¹⁴ |
| 1306 5 TCP MSS Option | Fires when a TCP MSS option is detected. All 1306 signatures fire an alert and do not function in promiscuous mode. | TCP Idle Timeout 3600 | Modify Packet Inline |
| 1306 6 TCP option data after EOL option | Fires when the TCP option list has data after the EOL option. All 1306 signatures fire an alert and do not function in promiscuous mode. | TCP Idle Timeout 3600 | Modify Packet Inline |
| 1307 TCP Window Variation | Fires when the right edge of the rcv window for TCP moves to the right (decreases). | TCP Idle Timeout 3600 | Deny Connection Inline Produce Alert ¹⁵ |
| 1308 TTL Evasion ¹⁶ | Fires when the TTL seen on one direction of a session is higher than the minimum that has been observed. | TCP Idle Timeout 3600 | Modify Packet Inline ¹⁷ |

Table 10-6 TCP Stream Reassembly Signatures (continued)

| Signature ID and Name | Description | Parameter With Default Value and Range | Default Actions |
|---|--|---|---|
| 1309 TCP Reserved Flags Set | Fires when the reserved bits (including bits used for ECN) are set on the TCP header. | TCP Idle Timeout 3600 | Modify Packet Inline Produce Alert ¹⁸ |
| 1311 TCP Packet Exceeds MSS | Fires when a packet exceeds the MSS that was exchanged during the three-way handshake. | TCP Idle Timeout 3600 | Produce Alert ¹⁹ |
| 1312 TCP MSS Below Minimum | Fires when the MSS value in a packet containing a SYN flag is less than TCP Min MSS. | TCP Min MSS 400 (0-16000) TCP Idle Timeout 3600 | Modify Packet Inline ²⁰ |
| 1313 TCP Max MSS | Fires when the MSS value in a packet containing a SYN flag exceeds TCP Max MSS | TCP Max MSS 1460 (0-16000) | Modify Packet Inline disabled ²¹ |
| 1314 TCP Data SYN | Fires when TCP payload is sent in the SYN packet. | — | Deny Packet Inline disabled ²² |
| 1315 ACK Without TCP Stream | Fires when an ACK packet is sent that does not belong to a stream. | — | Produce Alert disabled ²³ |
| 1317 Zero Window Probe | Fires when a zero window probe packet is detected. | Modify Packet Inline removes data from the Zero Window Probe packet. | Modify Packet Inline |
| 1330 ²⁴ 0 TCP Drop - Bad Checksum | Fires when TCP packet has bad checksum. | Modify Packet Inline corrects the checksum. | Deny Packet Inline |
| 1330 1 TCP Drop - Bad TCP Flags | Fires when TCP packet has bad flag combination. | — | Deny Packet Inline |
| 1330 2 TCP Drop - Urgent Pointer With No Flag | Fires when TCP packet has a URG pointer and no URG flag. | Modify Packet Inline clears the pointer. | Modify Packet Inline disabled |
| 1330 3 TCP Drop - Bad Option List | Fires when TCP packet has a bad option list. | — | Deny Packet Inline |
| 1330 4 TCP Drop - Bad Option Length | Fires when TCP packet has a bad option length. | — | Deny Packet Inline |
| 1330 5 TCP Drop - MSS Option Without SYN | Fires when TCP MSS option is seen in packet without the SYN flag set. | Modify Packet Inline clears the MSS option. | Modify Packet Inline |
| 1330 6 TCP Drop - WinScale Option Without SYN | Fires when TCP window scale option is seen in packet without the SYN flag set. | Modify Packet Inline clears the window scale option. | Modify Packet Inline |

Table 10-6 TCP Stream Reassembly Signatures (continued)

| Signature ID and Name | Description | Parameter With Default Value and Range | Default Actions |
|---|---|--|----------------------|
| 1330 7 TCP Drop - Bad WinScale Option Value | Fires when a TCP packet has a bad window scale value. | Modify Packet Inline sets the value to the closest constraint value. | Modify Packet Inline |
| 1330 8 TCP Drop - SACK Allow Without SYN | Fires when the TCP SACK allowed option is seen in a packet without the SYN flags set. | Modify Packet Inline clears the SACK allowed option. | Modify Packet Inline |
| 1330 9 TCP Drop - Data in SYN/ACK | Fires when TCP packet with SYN and ACK flags set also contains data. | — | Deny Packet Inline |
| 1330 10 TCP Drop - Data Past FIN | Fires when TCP data is sequenced after FIN. | — | Deny Packet Inline |
| 1330 11 TCP Drop - Timestamp not Allowed | Fires when TCP packet has timestamp option when timestamp option is not allowed. | — | Deny Packet Inline |
| 1330 12 TCP Drop - Segment Out of Order | Fires when TCP segment is out of order and cannot be queued. | — | Deny Packet Inline |
| 1330 13 TCP Drop - Invalid TCP Packet | Fires when TCP packet has invalid header. | — | Deny Packet Inline |
| 1330 14 TCP Drop - RST or SYN in window | Fires when TCP packet with RST or SYN flag was sent in the sequence window but was not the next sequence. | — | Deny Packet Inline |
| 1330 15 TCP Drop - Segment Already ACKed | Fires when TCP packet sequence is already ACKed by peer (excluding keepalives). | — | Deny Packet Inline |
| 1330 16 TCP Drop - PAWS Failed | Fires when TCP packet fails PAWS check. | — | Deny Packet Inline |
| 1330 17 TCP Drop - Segment out of State Order | Fires when TCP packet is not proper for the TCP session state. | — | Deny Packet Inline |
| 1330 18 TCP Drop - Segment out of Window | Fires when TCP packet sequence number is outside of allowed window. | — | Deny Packet Inline |
| 3050 Half Open SYN Attack | | syn-flood-max-embryonic 5000 | |
| 3250 TCP Hijack | | max-old-ack 200 | |
| 3251 TCP Hijack Simplex Mode | | max-old-ack 100 | |

- The timer is reset to 0 after each packet on the TCP session. By default, this signature does not produce an alert. You can choose to produce alerts for expiring TCP connections if desired. A statistic of total number of expired flows is updated any time a flow expires.

2. Modify Packet Inline, Deny Connection Inline, and Deny Packet Inline have no effect on this signature.
3. The timer starts with the first SYN packet and is not reset. State for the session is reset and any subsequent packets for this flow appear to be out of order (unless it is a SYN).
4. Modify Packet Inline, Deny Connection Inline, and Deny Packet Inline have no effect on this signature.
5. The timer starts with the first FIN packet and is not reset. State for the session is reset and any subsequent packets for this flow appear to be out of order (unless it is a SYN).
6. Modify Packet Inline, Deny Connection Inline, and Deny Packet Inline have no effect on this signature.
7. Modify Packet Inline and Deny Packet Inline have no effect on this signature. Deny Connection Inline drops the current packet and the TCP session.
8. Phrak 57 describes a way to evade security policy using URG pointers. You can normalize the packet when it is in inline mode with this signature.
9. Modify Packet Inline strips the URG flag and zeros the URG pointer from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
10. Modify Packet Inline strips the selected option(s) from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
11. Modify Packet Inline strips the selected ACK allowed option from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
12. Modify Packet Inline strips the selected ACK allowed option from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
13. Modify Packet Inline strips the timestamp option from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
14. Modify Packet Inline strips the window scale option from the packet. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
15. Modify Packet Inline has no effect on this signature. Deny Connection Inline drops the current packet and the TCP connection. Deny Packet Inline drops the packet.
16. This signature is used to cause TTLs to monotonically decrease for each direction on a session. For example, if TTL 45 is the lowest TTL seen from A to B, then all future packets from A to B will have a maximum of 45 if Modify Packet Inline is set. Each new low TTL becomes the new maximum for packets on that session.
17. Modify Packet Inline ensures that the IP TTL monotonically decreases. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
18. Modify Packet Inline clears all reserved TCP flags. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
19. Modify Packet Inline has no effect on this signature. Deny Connection Inline drops the current packet and the TCP connection. Deny Packet Inline drops the packet.
20. 2.4.21-15.EL.cisco.1 Modify Packet Inline raises the MSS value to TCP Min MSS. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet 2.4.21-15.EL.cisco.1.
21. Modify Packet Inline lowers the MSS value to TCP Max MSS. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet 2.4.21-15.EL.cisco.1.
22. Modify Packet Inline has no effect on this signature. Deny Connection Inline drops the current packet and the TCP session. Deny Packet Inline drops the packet.
23. Modify Packet Inline, Deny Connection Inline, and Deny Packet Inline have no effect on this signature. By default, the 1330 signatures drop packets for which this signature sends alerts.
24. These subsignatures represent the reasons why the Normalizer might drop a TCP packet. By default these subsignatures drop packets. These subsignatures let you permit packets that fail the checks in the Normalizer through the IPS. The drop reasons have an entry in the TCP statistics. By default these subsignatures do not produce an alert.

Configuring the TCP Stream Reassembly Mode



Tip

An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.

**Note**

The parameters TCP Handshake Required and TCP Reassembly Mode only impact sensors inspecting traffic in promiscuous mode, not inline mode. To configure asymmetric options for sensors inspecting inline traffic, use the Normalizer Mode parameter.

To configure the TCP stream reassembly mode, follow these steps:

- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures > Advanced > Miscellaneous**.
- Step 3** Under Stream Reassembly, in TCP Handshake Required field, choose **Yes**. Choosing TCP Handshake Required specifies that the sensor should only track sessions for which the three-way handshake is completed.
- Step 4** In the TCP Reassembly Mode field, from the drop-down list, choose the mode the sensor should use to reassemble TCP sessions:
- **Asymmetric**—Lets the sensor synchronize state with the flow and maintain inspection for those engines that do not require both directions.
 - **Strict**—If a packet is missed for any reason, all packets after the missed packet are processed.
 - **Loose**—Use in environments where packets might be dropped.

**Tip**

To discard your changes and close the Advanced dialog box, click **Cancel**.

- Step 5** Click **OK**, and then **Apply** to apply your changes and save the revised configuration

**Tip**

To discard your changes, click **Reset**.

For More Information

For information on asymmetric inspection options for sensors configured in inline mode, see [Inline TCP Session Tracking Mode, page 8-3](#) and [Adding, Editing, and Deleting Virtual Sensors, page 8-12](#).

Tuning a TCP Stream Reassembly Signature

**Tip**

An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.

**Caution**

For signature 3050 Half Open SYN Attack, if you choose Modify Packet Inline as the action, you can see as much as 20 to 30% performance degradation while the protection is active. The protection is only active during an actual SYN flood.

The following procedure demonstrates how to tune a TCP stream reassembly signatures, for example, signature 1313 0 TCP MSS Exceeds Maximum.

To tune a TCP stream reassembly signature, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures**.
- Step 3** From the Filter drop-down list, choose **Engine** and then choose **Normalizer**.
- Step 4** Select the TCP fragment reassembly signature you want to configure in the list, for example, Sig ID 1313 Subsig ID 0 TCP MSS Exceeds Maximum, and click **Edit**.
- Step 5** Change the default setting of any configurable IP fragment reassembly parameters for signature 1313. For example, in the TCP Max MSS field, change the setting from the default of 1460 to 1380. Changing this parameter from the default of 1460 to 1380 helps prevent fragmentation of traffic going through a VPN tunnel.
- Step 6** For signature 1313 0, you can also change the parameters of these options:
- Specify Hijack Max Old Ack
 - Specify TCP Idle Timeout
 - Specify Service Ports
 - Specify SYN Flood Max Embryonic



Tip To discard your changes and close the Edit Signature dialog box, click **Cancel**.

- Step 7** Click **OK**, and then **Apply** to apply your changes and save the revised configuration



Tip To discard your changes, click **Reset**.

Configuring IP Logging



Tip An empty check box indicates the default value is being used. Check the check box to configure that parameter. Click the value field to change the parameter. A green check indicates that a user-defined value is being used. Click the green check to change the value back to the default.

You can configure a sensor to generate an IP session log when the sensor detects an attack. When IP logging is configured as a response action for a signature and the signature is triggered, all packets to and from the source address of the alert are logged for a specified period of time.



Note IP logging allows a maximum limit of 20 concurrent IP log files. Once the limit of 20 is reached, you receive the following message in main.log: Cid/W errWarnIpLogProcessor::addIpLog: Ran out of file descriptors.

When the sensor meets any one of the IP logging conditions, it stops IP logging.

To configure IP logging parameters, follow these steps:

-
- Step 1** Log in to the IME using an account with administrator or operator privileges.
- Step 2** Choose **Configuration > sensor_name > Policies > Signature Definitions > sig0 > Active Signatures > Advanced > Miscellaneous**.
- Step 3** Under IP Log in the Max IP Log Packets field, enter the maximum number of packets per log. The range is 0 to 65535. The default is 0 packets.
- Step 4** In the IP Log Time field, enter the duration in seconds you want the sensor to log. A valid value is 30 to 300 seconds. The default is 30 seconds.
- Step 5** In the Max IP Log Bytes field, enter the maximum size in bytes per log. The range is 0 to 2147483647 bytes. The default is 0 bytes.



Tip To discard your changes and close the Advanced dialog box, click **Cancel**.

- Step 6** Click **OK**, and then **Apply** to apply your changes and save the revised configuration



Tip To discard your changes, click **Reset**.
