CHAPTER 5

Defining Signatures

This chapter explains how to configure signatures. It contains the following sections:

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 sensor's event store. 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.

IPS 5.1 contains over 1000 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.

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.

Configuring Signature Variables

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

Overview

When you want to use the same value within multiple signatures, use a variable. When you change the value of a variable, the variables in all signatures are updated. 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 choose 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.

Supported User Role

The following user roles are supported:

- Administrator
- Operator
- Viewer

You must be Administrator or Operator to configure signature variables.

Field Definitions

This section lists the field definitions for signature variables, and contains the following topics:

Signature Variables Pane

The following fields and buttons are found in the Signature Variables pane.

Field Descriptions:

- 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.

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

Button Functions:

- Add—Opens the Add Signature Variable dialog box. From this dialog box, you can add a new variable and specify the values associated with that variable.
- Edit—Opens the Edit Signature Variable dialog box. From this dialog box, you can change the values associated with this variable.
- Delete—Removes the selected variable from the list of available variables.
- Apply—Applies your changes and saves the revised configuration.
- Reset—Refreshes the pane by replacing any edits you made with the previously configured value.

Add and Edit Signature Variable Dialog Boxes

The following fields and buttons are found in the Add and Edit Signature Variable dialog boxes.

Field Descriptions:

- 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.

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

Button Functions:

- OK—Accepts your changes and closes the dialog box.
- Cancel—Discards your changes and closes the dialog box.
- Help—Displays the help topic for this feature.

Configuring Signature Variables

To configure signature variables, follow these steps:

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

Step 2 Choose **Configuration > Signature Definition > Signature Variables**.

The Signature Variables pane appears.

Step 3 Click **Add** to create a variable.

The Add Signature Variable dialog box appears.

Step 4 Type the name of the signature variable in the Name field.

Note A valid name can only contain numbers or letters. You can also use a hyphen (-) or underscore (_).

Step 5 Type the value into the Value field for the new signature variable.

Note You can use commas as delimiters. Make sure there are no trailing spaces after the comma. Otherwise, you receive a `Validation failed` error.

WEBPORTS 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.

Step 6 Click **OK**.

The new variable appears in the signature variables list in the Signature Variables pane.

Step 7 To edit an existing variable, select it in the signature variables list, and then click **Edit**.

The Edit Signature Variable dialog box appears for the variable that you chose.

Step 8 Make any necessary changes to the Value field.

Step 9 Click **OK**.

The edited variable appears in the signature variables list in the Signature Variables pane.

Step 10 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 in the Signature Variables pane.

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

Step 11 Click **Apply** to apply your changes and save the revised configuration.

Configuring Signatures

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

Overview

You can perform the following tasks in the Signature Configuration pane:

- Sort and view all signatures stored on the sensor.

  You can sort by attack type, protocol, service, operating system, action to be performed, engine, signature ID, or signature name.
- View the MySDN information about the selected signature.

  The MySDN pages list the key attributes, a description, any benign triggers, and any recommended filters for the selected signature.
- Edit (tune) an existing signature to change the value(s) associated with the parameter(s) for that signature.
- Create a signature, either by cloning an existing signature and using the parameters of that signature as a starting point for the new signature, or by adding a new signature from scratch.

  You can also use the Custom Signature Wizard to create a signature. The wizard guides you through the parameters that you must choose to configure a custom signature, including selection of the appropriate signature engine.
- Enable or disable an existing signature.
- Restore the factory defaults to the signature.
- Delete a custom signature.

  You cannot delete built-in signatures.
- Activate or retire an existing signature.
- Assign actions to a signature.

Supported User Role

The following user roles are supported:

- Administrator
- Operator
- Viewer

You must be Administrator or Operator to configure signatures.

Field Definitions

This section lists the field definitions for configuring signatures, and contains the following topics:

Signature Configuration Pane

The following fields and buttons are found in the Signature Configuration pane.

Field Descriptions:

- Select By—Lets you sort the list of signatures by selecting an attribute to sort on, such as protocol, service, or action.
- Select Criteria—Lets you further sort within a category by selecting a specific class within that category.

  For example, if you choose to sort by protocol, you can choose L2/L3/L4 protocol and view only signatures that are related to L2/L3/L4 protocol.
- Sig ID—Identifies the unique numerical value assigned to this signature.

  This value lets the sensor identify a particular signature.
- SubSig ID—Identifies the unique numerical value assigned to this subsignature.

  A SubSig ID is used to identify a more granular version of a broad 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.
- Action—Identifies the actions the sensor will take when this signature fires.
- 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 RR 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 signature's severity level is high
  - Severity Factor = 75 if signature's severity level is medium
  - Severity Factor = 50 if signature's severity level is low
  - Severity Factor = 25 if signature's severity level is informational

- 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.

Button Functions:

- Select All—Selects all signatures.
- MySDN Link—Opens the MySDN page for the selected signature.

  The MySDN pages lists the key attributes, a description, any benign triggers, and any recommended filters for the selected signature.
- Add—Opens the Add Signature dialog box. You can create a signature by selecting the appropriate parameters.
- Clone—Opens the Clone Signature dialog box. You can create a signature by changing the prepopulated values of the existing signature you chose to clone.
- Edit—Opens 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.
- Enable—Enables the selected signature.
- Disable—Disables the selected signature.
- Actions—Displays the Assign Actions dialog box.
- Restore Defaults—Returns all parameters to the default settings for the selected signature.
- Delete—Deletes the selected custom signature.

  You cannot delete built-in signatures.
- Activate—Activates the selected signature if the signature is retired.

  This process can take some time because the sensor has to add the signature back to the appropriate signature engine and reconstruct the signature engine.
- Retire—Retires the selected signature and removes it from the signature engine.
- Apply—Applies your changes and saves the revised configuration.
- Reset—Refreshes the pane by replacing any edits you made with the previously saved value.

Add Signatures Dialog Box

The following fields and buttons are found in the Add Signature dialog box:

Field Descriptions:

- 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. A subsignature ID is used to identify 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.

- Sig Description—Lets you specify the following attributes that help you distinguish this signature from other signatures:
  - Signature Name—Name your signature. The default is MySig.
  - Alert Notes—Add alert notes in this field.
  - User Comments—Add your comments about this signature in this field.
  - Alarm Traits—Add the alarm trait in this field. The value is 0 to 65535. The default is 0.
  - Release—Add the software release in which the signature first appeared.

- Engine—Lets you choose the engine that parses and inspects the traffic specified by this signature.
  - AIC FTP—Inspects FTP traffic and lets you control the commands being issued.
  - AIC HTTP—Provides granular control over HTTP sessions to prevent abuse of the HTTP protocol.
  - Atomic ARP—Inspects Layer-2 ARP protocol. The Atomic ARP engine is different because most engines are based on Layer-3-IP.
  - Atomic IP—Inspects IP protocol packets and associated Layer-4 transport protocols.
  - Flood Host—Detects ICMP and UDP floods directed at hosts.
  - Flood Net—Detects ICMP and UDP floods directed at networks.
  - Meta—Defines events that occur in a related manner within a sliding time interval. This engine processes events rather than packets.
  - Multi String—Defines signatures that inspect Layer 4 transport protocol (ICMP, TCP, and UDP) payloads using multiple string matches for one signature. You can specify a series of regular expression patterns that must be matched to fire the signature.
  - 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 DNS—Inspects DNS (TCP and UDP) traffic.
  - Service FTP—Inspects FTP traffic.

  - Service Generic—Decodes custom service and payload.
  - Service H225— Inspects VoIP traffic.
  - Service HTTP—Inspects HTTP traffic. The WEBPORTS variable defines inspection port for HTTP traffic.
  - Service IDENT—Inspects IDENT (client and server) traffic.
  - Service MSRPC—Inspects MSRPC traffic.
  - Service MSSQL—Inspects Microsoft SQL traffic.
  - Service NTP—Inspects NTP traffic.
  - Service RPC—Inspects RPC traffic.
  - Service SMB—Inspects SMB traffic.
  - Service SNMP—Inspects SNMP traffic.
  - Service SSH—Inspects SSH traffic.
  - State—Stateful searches of strings in protocols such as SMTP.
  - String ICMP—Searches on Regex strings based on ICMP protocol.
  - String TCP—Searches on Regex strings based on TCP protocol.
  - String UDP—Searches on Regex strings based on UDP protocol.
  - Sweep—Analyzes sweeps of ports, hosts, and services, from a single host (ICMP and TCP), from destination ports (TCP and UDP), and multiple ports with RPC requests between two nodes.
  - Sweep Other TCP—Analyzes TCP flag combinations from reconnaissance scans that are trying to get information about a single host. The signatures look for flags A, B, and C. When all three are seen, an alert is fired.
  - Traffic ICMP—Analyzes nonstandard protocols, such as TFN2K, LOKI, and DDOS. There are only two signatures with configurable parameters.
  - Trojan Bo2k—Analyzes traffic from the nonstandard protocol BO2K. There are no user-configurable parameters in this engine.
  - Trojan Tfn2k—Analyzes traffic from the nonstandard protocol TFN2K. There are no user-configurable parameters in this engine.
  - Trojan UDP—Analyzes traffic from the UDP protocol. There are no user-configurable parameters in this engine.

- Event Action—Lets you assign the actions the sensor takes when it responds to events.
  - 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 will still be 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 **Monitoring > Denied Attackers > Clear List**, which permits the addresses back on the network. For the procedure, see Monitoring the Denied Attackers List, page 11-2.

- 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 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 > Event Action Rules > General Settings**. For the procedure, see Configuring the General Settings, page 7-27.

- Deny Connection Inline—(Inline only) Terminates the current packet and future packets on this TCP flow.
- 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.

- 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 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.
- 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.
- Modify Packet Inline— Modifies packet data to remove ambiguity about what the end point might do with the packet.
- 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.
- Request Block Connection—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."
- Request Block Host—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."

Note For block actions, to set the duration of the block, choose **Configuration > Event Action Rules > General Settings**. For the procedure, see Configuring the General Settings, page 7-27.

  - 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. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."

Note Request Rate Limit applies to a select set of signatures. For the list of signatures for which you can request a rate limit, see Understanding Rate Limiting, page 8-3.

  - 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 Event Store, even if Produce Alert is not selected. You must have SNMP configured on the sensor to implement this action. For more information, see Chapter 9, "Configuring SNMP."
  - 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.

- 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—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—The storage type used to count events for this signature. You can 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. You can choose Yes or No 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—The mode of alert summarization. You can choose Fire All, Fire Once, Global Summarize, or Summarize.
  - Summary Interval—The time in seconds used in each summary alert. The value is 1 to 65535. The default is 15.
  - Summary Key—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. You can 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.
  - Retired—Let you choose whether the signature is retired or not. The default is no.

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:

- droppedPacket
- deniedFlow
- tcpOneWayResetSent

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.

Clone and Edit Signature Dialog Boxes

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

Field Descriptions:

- 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. A subsignature ID is used to identify 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.

- Sig Description—Lets you specify the following attributes that help you distinguish this signature from other signatures:
  - Signature Name—Name your signature. The default is MySig.
  - Alert Notes—Add alert notes in this field.
  - User Comments—Add your comments about this signature in this field.
  - Alarm Traits—Add the alarm trait in this field. The value is 0 to 65535. The default is 0.
  - Release—Add the software release in which the signature first appeared.

- Engine—Lets you choose the engine that parses and inspects the traffic specified by this signature.
  - AIC FTP—Inspects FTP traffic and lets you control the commands being issued.
  - AIC HTTP—Provides granular control over HTTP sessions to prevent abuse of the HTTP protocol.
  - Atomic ARP—Inspects Layer-2 ARP protocol. The Atomic ARP engine is different because most engines are based on Layer-3-IP.
  - Atomic IP—Inspects IP protocol packets and associated Layer-4 transport protocols.

- Flood Host—Detects ICMP and UDP floods directed at hosts.
- Flood Net—Detects ICMP and UDP floods directed at networks.
- Meta—Defines events that occur in a related manner within a sliding time interval. This engine processes events rather than packets.
- Multi String—Defines signatures that inspect Layer 4 transport protocol (ICMP, TCP, and UDP) payloads using multiple string matches for one signature. You can specify a series of regular expression patterns that must be matched to fire the signature.
- 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 DNS—Inspects DNS (TCP and UDP) traffic.
- Service FTP—Inspects FTP traffic.
- Service Generic—Decodes custom service and payload.
- Service H225— Inspects VoIP traffic.
- Service HTTP—Inspects HTTP traffic. The WEBPORTS variable defines inspection port for HTTP traffic.
- Service IDENT—Inspects IDENT (client and server) traffic.
- Service MSRPC—Inspects MSRPC traffic.
- Service MSSQL—Inspects Microsoft SQL traffic.
- Service NTP—Inspects NTP traffic.
- Service RPC—Inspects RPC traffic.
- Service SMB—Inspects SMB traffic.
- Service SNMP—Inspects SNMP traffic.
- Service SSH—Inspects SSH traffic.
- State—Stateful searches of strings in protocols such as SMTP.
- String ICMP—Searches on Regex strings based on ICMP protocol.
- String TCP—Searches on Regex strings based on TCP protocol.
- String UDP—Searches on Regex strings based on UDP protocol.
- Sweep—Analyzes sweeps of ports, hosts, and services, from a single host (ICMP and TCP), from destination ports (TCP and UDP), and multiple ports with RPC requests between two nodes.
- Sweep Other TCP—Analyzes TCP flag combinations from reconnaissance scans that are trying to get information about a single host. The signatures look for flags A, B, and C. When all three are seen, an alert is fired.
- Traffic ICMP—Analyzes nonstandard protocols, such as TFN2K, LOKI, and DDOS. There are only two signatures with configurable parameters.
- Trojan Bo2k—Analyzes traffic from the nonstandard protocol BO2K. There are no user-configurable parameters in this engine.
- Trojan Tfn2k—Analyzes traffic from the nonstandard protocol TFN2K. There are no user-configurable parameters in this engine.
- Trojan UDP—Analyzes traffic from the UDP protocol. There are no user-configurable parameters in this engine.

- Event Action—Lets you assign the actions the sensor takes when it responds to events.
  - 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 will still be 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 **Monitoring > Denied Attackers > Clear List**, which permits the addresses back on the network. For the procedure, see Monitoring the Denied Attackers List, page 11-2.

  - 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 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 > Event Action Rules > General Settings**. For the procedure, see Configuring the General Settings, page 7-27.

  - Deny Connection Inline—(Inline only) Terminates the current packet and future packets on this TCP flow.
  - 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.

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

  - Request Block Connection—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."
  - Request Block Host—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."

Note For block actions, to set the duration of the block, choose **Configuration > Event Action Rules > General Settings**. For the procedure, see Configuring the General Settings, page 7-27.

  - 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. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."

Note Request Rate Limit applies to a select set of signatures. For the list of signatures for which you can request a rate limit, see Understanding Rate Limiting, page 8-3.

  - 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 Event Store, even if Produce Alert is not selected. You must have SNMP configured on the sensor to implement this action. For more information, see Chapter 9, "Configuring SNMP."
  - 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.

- 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—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—The storage type used to count events for this signature. You can 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. You can choose Yes or No 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—The mode of alert summarization. You can choose Fire All, Fire Once, Global Summarize, or Summarize.
  - Summary Interval—The time in seconds used in each summary alert. The value is 1 to 65535. The default is 15.
  - Summary Key—The storage type used to summarize alerts. You can 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. You can 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.
  - Retired—Lets you choose whether the signature is retired or not. The default is no.

Button Functions:

- OK—Accepts your changes and closes the dialog box.
- Cancel—Discards your changes and closes the dialog box.
- Help—Displays the help topic for this feature.

Assign Actions Dialog Box

The following fields and buttons are found in the Assign Actions dialog box.

An event action is the sensor's response to an event. Event actions are configurable on a per signature basis.

Field Descriptions:

- 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 will still be 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 **Monitoring > Denied Attackers > Clear List**, which permits the addresses back on the network. For the procedure, see Monitoring the Denied Attackers List, page 11-2.

- 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 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 > Event Action Rules > General Settings**. For the procedure, see Configuring the General Settings, page 7-27.

- Deny Connection Inline—(Inline only) Terminates the current packet and future packets on this TCP flow.
- 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.

- 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 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.
- 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.
- Modify Packet Inline— Modifies packet data to remove ambiguity about what the end point might do with the packet.
- 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.
- Request Block Connection—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."
- Request Block Host—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."

Note For block actions, to set the duration of the block, choose **Configuration > Event Action Rules > General Settings**. For the procedure, see Configuring the General Settings, page 7-27.

- 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. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."

Note Request Rate Limit applies to a select set of signatures. For the list of signatures for which you can request a rate limit, see Understanding Rate Limiting, page 8-3.

- 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 Event Store, even if Produce Alert is not selected. You must have SNMP configured on the sensor to implement this action. For more information, see Chapter 9, "Configuring SNMP."
- 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.

Button Functions:

- Select All—Lets you choose all event actions.
- Select None—Clears all event action selections.

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:

- droppedPacket
- deniedFlow

- tcpOneWayResetSent

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.

Adding Signatures

To add signatures, follow these steps:

Tip A + icon indicates that more options are available for this parameter. Click the + icon to expand the section and view the remaining parameters.

Tip A green icon indicates that the parameter is currently using the default value. Click the green icon to change it to red, which activates the parameter field so you can edit the value.

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

Step 2 Choose **Configuration > Signature Definition > Signature Configuration**.

The Signature Configuration pane appears.

Step 3 To create a custom signature that is not based on an existing signature, follow these steps:

a. Click **Add** to open the Add Signature dialog box.

b. Specify a unique signature ID for the new signature in the Signature field.

c. Specify a unique subsignature ID for the new signature in the Subsignature field.

d. Click the green icon next to the Alert Severity field and choose the severity you want to associate with this signature.

e. Click the green icon next to the Signature Fidelity Rating field and specify a value between 1and 100 to represent the signature fidelity rating for this signature.

f. Complete the signature description fields and add any comments about this signature.

g. Select the engine the sensor will use to enforce this signature.

Note If you do not know which engine to choose, use the Custom Signature Wizard to help you create a custom signature. For more information, see Creating Custom Signatures, page 6-22.

h. Click the green icon next to the Event Actions field, and 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.

i. Under Event Counter, complete the Event Counter fields if you want events counted.

j. Under Alert Frequency, complete the Alert Frequency fields to specify how you want to receive alerts.

k. Under Status, choose **Yes** to enable the signature.

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

l. Under Status, specify if this signature is retired. Click **No** to activate the signature. This places the signature in the engine.

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

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

m. Click **OK**.

The new signature appears in the list with the Type set to Custom.

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

Step 4 Click **Apply** to apply your changes and save the revised configuration.

Cloning Signatures

In the Signature Configuration 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.

Tip A + icon indicates that more options are available for this parameter. Click the + icon to expand the section and view the remaining parameters.

Tip A green icon indicates that the parameter is currently using the default value. Click the green icon to change it to red, which activates the parameter field so you can edit the value.

To clone signatures, follow these steps:

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

Step 2 Choose **Configuration > Signature Definition > Signature Configuration**.

The Signature Configuration pane appears.

Step 3 To locate a signature, choose a sorting option from the Select By list.

For example, if you are searching for a UDP Flood signature, choose L2/L3/L4 Protocol and then UDP Floods.

The Signature Configuration pane refreshes and displays only those signatures that match your sorting criteria.

Step 4 To create a signature by using an existing signature as the starting point, select the signature and follow these steps:

a. Click **Clone** to open the Clone Signature dialog box.

b. Specify a unique signature ID for the new signature in the Signature field.

c. Specify a unique subsignature ID for the new signature in the Subsignature field.

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

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

e. Under Status, choose **Yes** to enable the signature.

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

f. Under Status, specify if this signature is retired. Click **No** to activate the signature. This places the signature in the engine.

Note A signature must be activated 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**.

g. Click **OK**.

The cloned signature now appears in the list with the Type set to Custom.

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

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

Tuning Signatures

To tune signatures, follow these steps:

Tip A + icon indicates that more options are available for this parameter. Click the + icon to expand the section and view the remaining parameters.

Tip A green icon indicates that the parameter is currently using the default value. Click the green icon to change it to red, which activates the parameter field so you can edit the value.

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

Step 2 Choose **Configuration > Signature Definition > Signature Configuration**.

The Signature Configuration pane appears.

Step 3 To locate a signature, choose a sorting option from the Select By list.

For example, if you are searching for a UDP Flood signature, choose L2/L3/L4 Protocol and then UDP Floods.

The Signature Configuration pane refreshes and displays only those signatures that match your sorting criteria.

Step 4 To tune an existing signature, select the signature, and follow these steps:

a. Click **Edit** to open the Edit Signature dialog box.

b. Review the parameter values and change the value of any parameter you want to tune.

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

c. Under Status, choose **Yes** to enable the signature.

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

d. Under Status, specify if this signature is retired. Click **No** to activate the signature. This places the signature in the engine.

Note A signature must be activated 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**.

e. Click **OK**.

The edited signature now appears in the list with the Type set to Tuned.

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

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

Enabling and Disabling Signatures

To enable signatures, follow these steps:

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

Step 2 Choose **Configuration > Signature Definition > Signature Configuration**.

The Signature Configuration pane appears.

Step 3 To locate a signature, choose a sorting option from the Select By list.

For example, if you are searching for a UDP Flood signature, choose L2/L3/L4 Protocol and then UDP Floods.

The Signature Configuration 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:

a. View the Enabled column to determine the status of the signature. A signature that is enabled has the value Yes in this column.

b. To enable a signature that is disabled, select the signature and click **Enable**.

c. To disable a signature that is enabled, select the signature and click **Disable**.

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

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

Activating and Retiring Signatures

Caution Activating and retiring signatures can take a very long time, up to 30 minutes or longer.

To activate and retire signatures, follow these steps:

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

Step 2 Choose **Configuration > Signature Definition > Signature Configuration**.

The Signature Configuration pane appears.

Step 3 To locate a signature, choose a sorting option from the Select By list.

For example, if you are searching for a UDP Flood signature, choose L2/L3/L4 Protocol and then UDP Floods.

The Signature Configuration pane refreshes and displays only those signatures that match your sorting criteria.

Step 4 To activate a signature that is retired, select the signature, and then click **Activate**.

Step 5 To retire a signature that is activated, select the signature and then click **Retire**.

Note If you retire a signature, that signature is removed from the engine but remains in the signature configuration list. You can later activate the retired signature, but doing so requires the sensor to rebuild the signature list for that engine and could delay signature processing.

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

Step 6 Click **Apply** to apply your changes and save the revised configuration.

Assigning Actions to Signatures

To assign actions to signatures, follow these steps:

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

Step 2 Choose **Configuration > Signature Definition > Signature Configuration**.

The Signature Configuration pane appears.

Step 3 To locate a signature, choose a sorting option from the Select By list.

For example, if you are searching for a UDP Flood signature, choose L2/L3/L4 Protocol and then UDP Floods.

The Signature Configuration pane refreshes and displays only those signatures that match your sorting criteria.

Step 4 To assign actions to a signature or set of signatures, select the signature(s), and then click **Actions**.

The Assign Actions dialog box appears.

a. Select the actions you want to assign to the signature(s).

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.

- 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 will still be 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 **Monitoring > Denied Attackers > Clear List**, which permits the addresses back on the network. For the procedure, see Monitoring the Denied Attackers List, page 11-2.

- 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 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 > Event Action Rules > General Settings**. For the procedure, see Configuring the General Settings, page 7-27.

- Deny Connection Inline—(Inline only) Terminates the current packet and future packets on this TCP flow.
- 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.

- 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 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.
- 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.
- Modify Packet Inline— Modifies packet data to remove ambiguity about what the end point might do with the packet.
- 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.
- Request Block Connection—Sends a request to ARC to block this connection. You must have blocking devices configured to implement this action. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."
- Request Block Host—Sends a request to ARC to block this attacker host. You must have blocking devices configured to implement this action. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."

Note For block actions, to set the duration of the block, choose **Configuration > Event Action Rules > General Settings**. For the procedure, see Configuring the General Settings, page 7-27.

- 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. For more information, see Chapter 8, "Configuring Attack Response Controller for Blocking and Rate Limiting."

Note Request Rate Limit applies to a select set of signatures. For the list of signatures for which you can request a rate limit, see Understanding Rate Limiting, page 8-3.

- 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 Event Store, even if Produce Alert is not selected. You must have SNMP configured on the sensor to implement this action. For more information, see Chapter 9, "Configuring SNMP."
- 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 choose more than one action, hold down the **Ctrl** key.

b. If you want to assign all actions to the selected signatures, click **All**. Or, if you want to remove all actions from the selected signatures, choose **None**.

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

c. Click **OK** to save your changes and close the dialog box.

The new action now appears in the Action column.

Configuring the Miscellaneous Pane

This section describes how to configure the Miscellaneous pane, and contains the following topics:

Overview

In the Miscellaneous pane, you can perform the following tasks:

- Configure the application policy parameters

  You can configure the sensor to provide Layer 4 to Layer 7 packet inspection to prevent malicious attacks related to web services.

- 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.

- 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.

- 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.

Supported User Role

The following user roles are supported:

- Administrator
- Operator
- Viewer

You must be Administrator or Operator to configure the parameters in the Miscellaneous pane.

Field Definitions

The following fields and buttons are found in the Miscellaneous pane.

- Application Policy—Lets you configure application policy enforcement.
  - Enable HTTP —Enables protection for web services. Choose **Yes** 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.

Note We recommend that you not configure AIC web ports, but rather use the default web ports.

  - Enable FTP—Enables protection for web services. Choose **Yes** to require the sensor to inspect FTP traffic.

- Fragment Reassembly—Lets you configure 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 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—May only be seeing 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 IP logging when any of the following conditions are met:
  - Max IP Log Packets—Identifies the number of packets you want logged.
  - IP Log Time—Identifies the duration you want the sensor to log. A valid value is 1 to 60 seconds. The default is 30 seconds.
  - Max IP Log Bytes—Identifies the maximum number of bytes you want logged.

Button Functions:

- Apply—Applies your changes and saves the revised configuration.
- Reset—Refreshes the pane by replacing any edits you made with the previously configured value.

Configuring Application Policy

This section describes Application Policy (AIC) signatures and how to configure them. For more information on this signature engine, see AIC Engine, page B-8. This section contains the following topics:

Overview

AIC provides detailed analysis of web traffic. It provides granular control over HTTP sessions to prevent abuse of the HTTP protocol. It also allows administrative control over applications that attempt to tunnel over specified ports, such as instant messaging, and tunneling applications such as, gotomypc. Inspection and policy checks for P2P and instant messaging is possible if these applications are running over HTTP.

AIC also provides a way to inspect FTP traffic and control the commands being issued.

You can enable or disable the predefined signatures or you can create policies through custom signatures.

The AIC engine runs when HTTP traffic is received on AIC web ports. If traffic is web traffic, but not received on the AIC web ports, the Service HTTP engine is executed. AIC inspection can be on any port if it is configured as an AIC web port and the traffic to be inspected is HTTP traffic.

Caution The AIC web ports are regular HTTP web ports. You can turn on AIC web ports to distinguish which ports should watch for regular HTTP traffic and which ports should watch for AIC enforcement. You might use AIC web ports, for example, if you have a proxy on port 82 and you need to monitor it. We recommend that you do not configure separate ports for AIC enforcement.

AIC has the following categories of signatures:

- HTTP request method
  - Define request method
  - Recognized request methods

  For a list of signature IDs and descriptions, see AIC Request Method Signatures, page 5-29.
- MIME type
  - Define content type
  - Recognized content type

  For a list of signature IDs and descriptions, see AIC MIME Define Content Type Signatures, page 5-30. For the procedure for creating a custom MIME signature, see Example Recognized Define Content Type (MIME) Signature, page 5-35.
- 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

  For a list of signature IDs and descriptions, see AIC Transfer Encoding Signatures, page 5-33.

- FTP commands

  Associates an action with an FTP command. For a list of signature IDs and descriptions, see AIC FTP Commands Signatures, page 5-33.

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 5-1 lists the predefined define request method signatures. Enable the signatures that have the predefined method you need. For the procedure for enabling signatures, see Enabling and Disabling Signatures, page 5-22.

Table 5-1 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 GETATTRIBUTENAME

Table 5-1 Request Method Signatures (continued)

Signature ID	Define Request Method
12710	Define Request Method GETPROPERTIES
12711	Define Request Method STARTENV
12712	Define Request Method STOPREV

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 5-2 on page 5-30 lists the predefined define content type signatures. Enable the signatures that have the predefined content type you need. For the procedure for enabling signatures, see Enabling and Disabling Signatures, page 5-22. You can also create custom define content type signatures. For the procedure, see Example Recognized Define Content Type (MIME) Signature, page 5-35.

Table 5-2 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 12623 1 12623 2	Content Type image/tiff Header Check Content Type image/tiff Invalid Message Length Content Type image/tiff Verification Failed
12624 0 12624 1 12624 2	Content Type image/x-3ds Header Check Content Type image/x-3ds Invalid Message Length Content Type image/x-3ds Verification Failed
12626 0 12626 1 12626 2	Content Type image/x-portable-bitmap Header Check Content Type image/x-portable-bitmap Invalid Message Length Content Type image/x-portable-bitmap Verification Failed
12627 0 12627 1 12627 2	Content Type image/x-portable-graymap Header Check Content Type image/x-portable-graymap Invalid Message Length Content Type image/x-portable-graymap Verification Failed
12628 0 12628 1 12628 2	Content Type image/jpeg Header Check Content Type image/jpeg Invalid Message Length Content Type image/jpeg Verification Failed
12629 0 12629 1	Content Type image/cgf Header Check Content Type image/cgf Invalid Message Length
12631 0 12631 1	Content Type image/x-xpm Header Check Content Type image/x-xpm Invalid Message Length

Table 5-2 Define Content Type Signatures (continued)

Signature ID	Signature Description
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
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

Table 5-2 Define Content Type Signatures (continued)

Signature ID	Signature Description
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
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

Table 5-2 Define Content Type Signatures (continued)

Signature ID	Signature Description
12669 0	Content Type image/x-bitmap Header Check
12669 1	Content Type image/x-bitmap Invalid Message Length
12673 0	Recognized content type

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 5-3 lists the predefined transfer encoding signatures. Enable the signatures that have the predefined transfer encoding method you need. For the procedure for enabling signatures, see Enabling and Disabling Signatures, page 5-22.

Table 5-3 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

AIC FTP Commands Signatures

Table 5-4 on page 5-33 lists the predefined FTP commands signatures. Enable the signatures that have the predefined FTP command you need. For the procedure for enabling signatures, see Enabling and Disabling Signatures, page 5-22.

Table 5-4 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

Table 5-4 FTP Commands Signatures (continued)

Signature ID	FTP Command
12906	Define FTP command cwd
12907	Define FTP command dele
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

Configuring Application Policy

Tip A + icon indicates that more options are available for this parameter. Click the + icon to expand the section and view the remaining parameters.

Tip A green icon indicates that the parameter is using the default value. Click the green icon to change it to red, which activates the parameter field so you can edit the value.

To configure the application policy parameters, follow these steps:

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

Step 2 Choose **Configuration > Signature Definition > Miscellaneous**.

The Miscellaneous pane appears.

Step 3 Under Application Policy, click the green icon next to Enable HTTP and choose Yes to enable inspection of HTTP traffic.

Step 4 Click the green icon next to Max HTTP Requests and specify the number of outstanding HTTP requests per connection that can be outstanding without having received a response from the server.

Step 5 (Optional) Click the green icon next to AIC Web Ports and specify the ports you want active.

Note We recommend that you not configure AIC web ports, but rather use the default web ports.

Step 6 Click the green icon next to Enable FTP and choose Yes 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 **Reset**.

Step 7 Click **Apply** to apply your changes and save the revised configuration.

Example Recognized Define Content Type (MIME) Signature

Tip A + icon indicates that more options are available for this parameter. Click the + icon to expand the section and view the remaining parameters.

Tip A green icon indicates that the parameter is using the default value. Click the green icon to change it to red, which activates the parameter field so you can edit the value.

The following example demonstrates how to tune a Recognized Content Type (MIME) signature.

To tune a MIME-type policy signature, for example Signature 12623 1 (Content Type image/tiff Invalid Message Length), follow these steps:

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

Step 2 Choose **Configuration > Signature Definition > Signature Configuration**.

The Signature Configuration pane appears.

Step 3 In the Select By box, choose **Engine**.

Step 4 In the Select Engine box, choose AIC HTTP.

Step 5 Scroll down the list and select Sig ID 12623 1, and click **Edit**.

The Edit Signature dialog box appears.

Step 6 Under Status, click the green icon next to Enabled and choose **Yes**.

Step 7 Click the green icon next to Content Type Details and choose one of the options, for example, Length.

Step 8 In the Length field, make the length smaller by changing the default to 30,000.

Step 9 Click **OK**.

Step 10 Click **Apply** to save the changes or click **Reset** to discard them.

Configuring IP Fragment Reassembly

This section describes IP fragment reassembly, lists the IP fragment reassembly signatures with their configurable parameters, and describes how to configure them. For more information on this signature engine, see Normalizer Engine, page B-15.

This section contains the following topics:

Overview

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 reassemble 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.

You configure the IP fragment reassembly per signature.

IP Fragment Reassembly Signatures and Configurable Parameters

Table 5-5 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 5-5 IP Fragment Reassembly Signatures

IP Fragment Reassembly Signature	Parameter With Default Value
1200 IP Fragmentation Buffer Full	Specify Max Fragments 10000
1201 IP Fragment Overlap	None
1202 IP Fragment Overrun - Datagram Too Long	Specify Max Datagram Size 65536
1203 IP Fragment Overwrite - Data is Overwritten	None
1204 IP Fragment Missing Initial Fragment	None
1205 IP Fragment Too Many Datagrams	Specify Max Partial Datagrams 1000
1206 IP Fragment Too Small	Specify Max Small Frags 2 Specify Min Fragment Size 400
1207 IP Fragment Too Many Datagrams	Specify Max Fragments per Datagram 170
1208 IP Fragment Incomplete Datagram	Specify Fragment Reassembly Timeout 60
1220 Jolt2 Fragment Reassembly DoS attack	Specify Max Last Fragments 4
1225 Fragment Flags Invalid	None

Configuring IP Fragment Reassembly Signatures

Tip A + icon indicates that more options are available for this parameter. Click the + icon to expand the section and view the remaining parameters.

Tip A green icon indicates that the parameter is using the default value. Click the green icon to change it to red, which activates the parameter field so you can edit the value.

To configure IP fragment reassembly parameters for a particular signature, follow these steps:

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

Step 2 Choose **Configuration > Signature Definition > Signature Configuration.**

Step 3 In the Select By box, choose **Engine**.

Step 4 In the Select Engine box, choose **Normalizer**.

Step 5 Select the IP fragment reassembly signature you want to configure in the list, for example, Sig ID 1200 SubSig 0, and click **Edit**.

The Edit Signature dialog box appears.

Step 6 Change the default setting of any IP fragment reassembly parameters that can be configured for signature 1200. For example, click the green icon next to Max Fragments and 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, click **Reset**.

Step 7 Click **Apply** to apply your changes and save the revised configuration.

Configuring the Method for IP Fragment Reassembly

Tip A + icon indicates that more options are available for this parameter. Click the + icon to expand the section and view the remaining parameters.

Tip A green icon indicates that the parameter is using the default value. Click the green icon to change it to red, which activates the parameter field so you can edit the value.

Note You can configure this option if your sensor is operating in promiscuous mode. If your sensor is operating in line mode, the method is NT only.

To configure the method the sensor will use for IP fragment reassembly, follow these steps:

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

Step 2 Choose **Configuration > Signature Definition > Miscellaneous**.

The Miscellaneous pane appears.

Step 3 Under Fragment Reassembly, click the green icon next to IP Reassembly Mode and choose the operating system you want to use to reassemble the fragments.

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

Step 4 Click **Apply** to apply your changes and save the revised configuration.

Configuring TCP Stream Reassembly

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. For more information on this signature engine, see Normalizer Engine, page B-15.

This section contains the following topics:

Overview

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.

TCP Stream Reassembly Signatures and Configurable Parameters

Table 5-6 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 5-6 TCP Stream Reassembly Signatures

Signature ID and Name	Description	Parameter With Default Value and Range	Default Actions
1300 TCP Segment Overwrite[1]	Fires when the data in an overlapping TCP segment (such as a retransmit) sends data that is different from the data already seen on this session	—	Deny Connection Inline Product Alert[2]
1301 TCP Inactive Timeout[3]	Fires when a TCP session has been idle for a TCP Idle Timeout.	TCP Idle Timeout 3600 (15-3600)	None[4]
1302 TCP Embryonic Timeout[5]	Fires when a TCP session has not completes the three-way handshake in TCP embryonic timeout seconds.	TCP Embryonic Timeout 15 (3-300)	None[6]

Table 5-6 TCP Stream Reassembly Signatures (continued)

Signature ID and Name	Description	Parameter With Default Value and Range	Default Actions
1303 TCP Closing Timeout[7]	Fires when a TCP session has not closed completely in TCP Closed Timeout seconds after the first FIN.	TCP Closed Timeout 5 (1-60)	None[8]
1304 TCP Max Segments Queued Per Session	Fires when the number of queued out of order segments for a session exceed TCP Max Queue. The segment containing the sequence furthermost from the expected sequence is dropped.	TCP Max Queue 32 (0-128)	Deny Packet Inline Product Alert[9]
1305 TCP Urgent Flag[10]	Fires when the TCP urgent flag is seen	None	Modify Packet Inline is disabled[11]
1306 0 TCP Option Other	Fires when a TCP option in the range of TCP Option Number is seen.	TCP Option Number 6-7,9-255 (Integer Range Allow Multiple 0-255 constraints)	Modify Packet Inline Produce Alert[12]
1306 1 TCP SACK Allowed Option	Fires when a TCP selective ACK allowed option is seen.	—	Modify Packet Inline disabled[13]
1306 2 TCP SACK Data Option	Fires when a TCP selective ACK data option is seen.	—	Modify Packet Inline disabled[14]
1306 3 TCP Timestamp Option	Fires when a TCP timestamp option is seen.	—	Modify Packet Inline disabled[15]
1306 4 TCP Window Scale Option	Fires when a TCP window scale option is seen.	—	Modify Packet Inline disabled[16]
1307 TCP Window Size Variation	Fires when the right edge of the recv window for TCP moves to the right (decreases).	—	Deny Connection Inline Produce Alert disabled[17]
1308 TTL Varies[18]	Fire when the TTL seen on one direction of a session is higher than the minimum that has been observed	—	Modify Packet Inline[19]
1309 TCP Reserved Bits Set	Fires when the reserved bits (including bits used for ECN) are set on the TCP header.	—	Modify Packet Inline Produce Alert disabled[20]

Table 5-6 TCP Stream Reassembly Signatures (continued)

Signature ID and Name	Description	Parameter With Default Value and Range	Default Actions
1310 TCP Retransmit Protection[21]	Fires when the sensor detects that a retransmitted segment has different data than the original segment.	—	Deny Connection Inline Produce Alert[22]
1311 TCP Packet Exceeds MSS	Fires when a packet exceeds the MSS that was exchanged during the three-way handshake.	—	Deny Connection Inline Produce Alert[23]
1312 TCP Min MSS	Fires when the MSS value in a packet containing a SYN flag is less that TCP Min MSS.	TCP Min MSS 400 (0-16000)	Modify Packet Inline disabled[24]
1313 TCP Max MSS	Fires when the MSS value in a packet containing a SYN flag exceed TCP Max MSS	TCP Max MSS1460 (0-16000)	Modify Packet Inline disabled[25]
1314 TCP Data SYN	Fires when TCP payload is sent in the SYN packet.	—	Deny Packet Inline disabled[26]
1315 ACK Without TCP Stream	Fires when an ACK packet is sent that does not belong to a stream.	—	Produce Alert disabled[27]
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[28] 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

Table 5-6 TCP Stream Reassembly Signatures (continued)

Signature ID and Name	Description	Parameter With Default Value and Range	Default Actions
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
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

Table 5-6 TCP Stream Reassembly Signatures (continued)

Signature ID and Name	Description	Parameter With Default Value and Range	Default Actions
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	

1. IPS keeps the last 256 bytes in each direction of the TCP session.
2. 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.
3. 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.
4. Modify Packet Inline, Deny Connection Inline, and Deny Packet Inline have no effect on this signature.
5. 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).
6. Modify Packet Inline, Deny Connection Inline, and Deny Packet Inline have no effect on this signature.
7. 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).
8. Modify Packet Inline, Deny Connection Inline, and Deny Packet Inline have no effect on this signature.
9. Modify Packet Inline and Deny Packet Inline have no effect on this signature. Deny Connection Inline drops the current packet and the TCP session.
10. 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.
11. 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.
12. 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.
13. 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.
14. 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.
15. 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.
16. 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.
17. 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.
18. 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.
19. 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.
20. 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.
21. This signature is not limited to the last 256 bytes like signature 1300.
22. 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.
23. 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.

24. 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.
25. 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.
26. 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.
27. 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.
28. 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 TCP Stream Reassembly Signatures

Tip A + icon indicates that more options are available for this parameter. Click the + icon to expand the section and view the remaining parameters.

Tip A green icon indicates that the parameter is using the default value. Click the green icon to change it to red, which activates the parameter field so you can edit the value.

To configure TCP stream reassembly parameters for a particular signature, follow these steps:

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

Step 2 Choose **Configuration > Signature Definition > Signature Configuration.**

Step 3 In the Select By box, choose **Engine**.

Step 4 In the Select Engine box, choose **Normalizer**.

Step 5 Select the TCP fragment reassembly signature you want to configure in the list, for example, Sig ID 1313 SubSig 0, and click **Edit**.

The Edit Signature dialog box appears.

Step 6 Change the default setting of any configurable IP fragment reassembly parameters for signature 1313. For example, click the green icon next to TCP Max MSS and change the setting from the default of 1460 to 1380.

Note Changing this parameter from the default of 1460 to 1380 helps prevent fragmentation of traffic going through a VPN tunnel.

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, click **Reset**.

Step 7 Click **Apply** to apply your changes and save the revised configuration.

Configuring the Mode for TCP Stream Reassembly

Tip A + icon indicates that more options are available for this parameter. Click the + icon to expand the section and view the remaining parameters.

Tip A green icon indicates that the parameter is using the default value. Click the green icon to change it to red, which activates the parameter field so you can edit the value.

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

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

Step 2 Choose **Configuration > Signature Definition > Miscellaneous**.

The Miscellaneous pane appears.

Step 3 Under Stream Reassembly, click the green icon next to TCP Handshake Required and choose yes.

Selecting TCP Handshake Required specifies that the sensor should only track sessions for which the three-way handshake is completed.

Step 4 Click the green icon next to TCP Reassembly Mode and 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, click **Reset**.

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

Configuring IP Logging

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.

Tip A + icon indicates that more options are available for this parameter. Click the + icon to expand the section and view the remaining parameters.

Tip A green icon indicates that the parameter is currently using the default value. Click the green icon to change it to red, which activates the parameter field so you can edit the value.

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

To configure the IP logging parameters, follow these steps:

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

Step 2 Choose **Configuration > Signature Definition > Miscellaneous**.

The Miscellaneous pane appears.

Step 3 Under IP Log, click the green icon next to Max IP Log Packets and then specify the number of packets you want logged.

Step 4 Click the green icon next to IP Log Time and then specify the duration you want the sensor to log.

A valid value is 1 to 60 minutes. The default is 30 minutes.

Step 5 Click the green icon next to Max IP Log Bytes and then specify the maximum number of bytes you want logged.

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

Step 6 Click **Apply** to apply your changes and save the revised configuration.

Example MEG Signature

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 SEAP. SEAP 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. For more information about SEAP, see Signature Event Action Processor, page 7-4.

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

The following example demonstrates how to create a MEG signature based on the Meta engine.

For example, signature 64000 subsignature 0 will fire 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.

Tip The Meta engine is different from other engines in that it takes alerts as input where most engines take packets as input. For more information on the Meta engine, see Meta Engine, page B-13.

Tip A + icon indicates that more options are available for this parameter. Click the + icon to expand the section and view the remaining parameters.

Tip A green icon indicates that the parameter is currently using the default value. Click the green icon to change it to red, which activates the parameter field so you can edit the value.

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

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

Step 2 Choose **Configuration > Signature Definition > Signature Configuration**.

The Signature Configuration pane appears.

Step 3 Click **Add** to open the Add Signature dialog box.

Step 4 Specify a unique signature ID for the new signature in the Signature field.

Step 5 Specify a unique subsignature ID for the new signature in the Subsignature field.

Step 6 Click the green icon next to the Alert Severity field and choose the severity you want to associate with this signature.

Step 7 Click the green icon next to the Signature Fidelity Rating field and specify a value between 1and 100 to represent the signature fidelity rating for this signature.

Step 8 Leave the default value for the Promiscuous Delta field.

Step 9 Complete the signature description fields and add any comments about this signature.

Step 10 Select Meta in the Engine field.

Step 11 Configure the Meta engine-specific parameters:

a. Click the green icon next to the Meta Reset Interval field and specify the time in seconds to reset the Meta signature.

The valid range is 0 to 3600 seconds. The default is 60 seconds.

b. Choose the storage type for the Meta signature from the Meta Key list:

- Attacker address
- Attacker and victim addresses
- Attacker and victim addresses and ports
- Victim address

c. Click the pencil icon next to Component List to insert the new MEG signature.

The Component List dialog box appears.

d. Click **Add** to insert the first MEG signature.

The Add List Entry dialog box appears.

e. Specify a name for the entry in the Entry Key field, for example, Entry1.

The default is MyEntry.

f. Under Component Group, specify the number of times this component must fire before it is satisfied in the Component Count field.

g. Under Component Group, specify the signature ID of the signature (2000 in this example) on which to match this component in the Component Sig ID field.

h. Under Component Group, specify the subsignature ID of the signature (0 in this example) on which to match this component in the Component SubSig ID field.

i. Click **OK**.

You are returned to the Add List Entry dialog box.

j. Highlight your entry and click **Select** to move it to the Selected Entries list.

k. Click **OK**.

l. Click **Add** to insert the next MEG signature.

The Add List Entry dialog box appears.

m. Specify a name for the entry in the Entry Key field, for example Entry2.

n. Under Component Group, specify the number of times this component must fire before it is satisfied in the Component Count field.

o. Under Component Group, specify the signature ID of the signature (3000 in this example) on which to match this component in the Component Sig ID field.

p. Under Component Group, specify the subsignature ID of the signature (0 in this example) on which to match this component in the Component SubSig ID field.

q. Click **OK**.

You are returned to the Add List Entry dialog box.

r. Highlight your entry and click **Select** to move it to the Selected Entries list.

s. Highlight the new entry and click **Move Up** or **Move Down** to order the new entry.

Tip To return the entries to the Entry Key list, click **Reset Ordering**.

t. Click **OK**.

u. Click the green icon next to the Component List in Order field and choose Yes to have the component list fire in order.

Step 12 Click the green icon next to the Event Action field, and 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 to ensure that all of the actions stay selected.

Step 13 Under Event Counter, complete the Event Counter fields if you want events counted.

Step 14 Under Alert Frequency, complete the Alert Frequency fields to specify how you want to receive alerts.

Step 15 Under Status, choose **Yes** to enable the signature.

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

Step 16 Under Status, specify if this signature is retired. Click **No** to activate the signature. This places the signature in the engine.

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

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.