

# Inspection of Basic Internet Protocols

The following topics explain application inspection for basic Internet protocols. For information on why you need to use inspection for certain protocols, and the overall methods for applying inspection, see Getting Started with Application Layer Protocol Inspection.


- DCERPC Inspection, on page 1
- DNS Inspection, on page 4
- FTP Inspection, on page 8
- HTTP Inspection, on page 12
- ICMP Inspection, on page 16
- ICMP Error Inspection, on page 16
- ILS Inspection, on page 17
- Instant Messaging Inspection, on page 17
- IP Options Inspection, on page 19
- IPsec Pass Through Inspection, on page 21
- IPv6 Inspection, on page 22
- NetBIOS Inspection, on page 24
- PPTP Inspection, on page 25
- RSH Inspection, on page 25
- SMTP and Extended SMTP Inspection, on page 25
- SNMP Inspection, on page 29
- SQL*Net Inspection, on page 30
- Sun RPC Inspection, on page 30
- TFTP Inspection, on page 32
- XDMCP Inspection, on page 32
- VXLAN Inspection, on page 32
- History for Basic Internet Protocol Inspection, on page 33


# DCERPC Inspection

DCERPC inspection is not enabled in the default inspection policy, so you must enable it if you need this inspection. You can simply edit the default global inspection policy to add DCERPC inspection. You can alternatively create a new service policy as desired, for example, an interface-specific policy.

The following sections describe the DCERPC inspection engine.

# DCERPC Overview

Microsoft Remote Procedure Call (MSRPC), based on DCERPC, is a protocol widely used by Microsoft distributed client and server applications that allows software clients to execute programs on a server remotely.

This typically involves a client querying a server called the Endpoint Mapper listening on a well known port number for the dynamically allocated network information of a required service. The client then sets up a secondary connection to the server instance providing the service. The security appliance allows the appropriate port number and network address and also applies NAT, if needed, for the secondary connection.

The DCERPC inspection engine inspects for native TCP communication between the EPM and client on well known TCP port 135. Map and lookup operations of the EPM are supported for clients. Client and server can be located in any security zone. The embedded server IP address and Port number are received from the applicable EPM response messages. Since a client may attempt multiple connections to the server port returned by EPM, multiple use of pinholes are allowed, which have configurable timeouts.

DCE inspection supports the following universally unique identifiers (UUIDs) and messages:

- End point mapper (EPM) UUID. All EPM messages are supported.
- ISystemMapper UUID (non-EPM). Supported messages are:
  - RemoteCreateInstance opnum4
  - RemoteGetClassObject opnum3
- OxidResolver UUID (non-EPM). Supported message is:
  - ServerAlive2 opnum5
- Any message that does not contain an IP address or port information because these messages do not require inspection.

# Configure a DCERPC Inspection Policy Map

To specify additional DCERPC inspection parameters, create a DCERPC inspection policy map. You can then apply the inspection policy map when you enable DCERPC inspection.

When defining traffic matching criteria, you can either create a class map or include the match statements directly in the policy map. The difference between creating a class map and defining the traffic match directly in the inspection policy map is that you can reuse class maps. The following procedure covers inspection policy maps, but also explains the traffic matching criteria available in the class map. To create a class map, select **Configuration** > **Firewall** > **Objects** > **Class Maps** > **DCERPC**.

**Tip** You can configure inspection maps while creating service policies, in addition to the procedure explained below. The contents of the map are the same regardless of how you create it.

### Procedure

**Step 1** Choose **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **DCERPC**.

**Step 2** Do one of the following:

- Click **Add** to add a new map.
- Select a map to view its contents. You can change the security level directly, or click **Customize** to edit the map. The remainder of the procedure assumes you are customizing or adding a map.

**Step 3** For new maps, enter a name (up to 40 characters) and description. When editing a map, you can change the description only.

**Step 4** In the **Security Level** view of the DCERPC Inspect Map dialog box, select the level that best matches your desired configuration.

If one of the preset levels matches your requirements, you are now done. Just click **OK**, skip the rest of this procedure, and use the map in a service policy rule for DCERPC inspection.

If you need to customize the settings further, click **Details** and continue with the procedure.

**Tip**
The **UUID Filtering** button is a shortcut to configure message filtering, which is explained later in this procedure.

**Step 5** Configure the desired options.

- **Pinhole Timeout**—Sets the pinhole timeout. Because a client may use the server information returned by the endpoint mapper for multiple connections, the timeout value is configurable based on the client application environment. Range is from 0:0:1 to 1193:0:0.
- **Enforce endpoint-mapper service**—Whether to enforce the endpoint mapper service during binding so that only its service traffic is processed.
- **Enable endpoint-mapper service lookup**—Whether to enable the lookup operation of the endpoint mapper service. You can also enforce a timeout for the service lookup. If you do not configure a timeout, the pinhole timeout is used.

**Step 6** (Optional.) Click the **Inspections** tab and define the actions to take for specific types of messages.

You can define traffic matching criteria based on DCERPC class maps, by configuring matches directly in the inspection map, or both.

a) Do any of the following:
   - Click **Add** to add a new criterion.
   - Select an existing criterion and click **Edit**.

b) Choose **Single Match** to define the criterion directly, or **Multiple Match**, in which case you select the DCERPC class map that defines the criteria.

c) If you are defining the criterion here, choose the match type for the criteria: **Match** (traffic must match the criterion) or **No Match** (traffic must not match the criterion). Then, select the desired UUID:
   - **ms-rpc-epm**—Matches Microsoft RPC EPM messages.
   - **ms-rpc-isystemactivator**—Matches ISystemMapper messages.
   - **ms-rpc-oxidresolver**—Matches OxidResolver messages.

d) Choose whether to **Reset** or **Log** the connection. You can also enable logging if you elect to reset the connection. Resetting the connection drops the packet, closes the connection, and sends a TCP reset to the server or client.
e) Click **OK** to add the criterion. Repeat the process as needed.

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

You can now use the inspection map in a DCERPC inspection service policy.

### What to do next

You can now configure an inspection policy to use the map. See Configure Application Layer Protocol Inspection.

# DNS Inspection

DNS inspection is enabled by default. You need to configure it only if you want non-default processing. The following sections describe DNS application inspection.

## Defaults for DNS Inspection

DNS inspection is enabled by default, using the preset_dns_map inspection class map:

- The maximum DNS message length is 512 bytes.
- DNS over TCP inspection is disabled.
- The maximum client DNS message length is automatically set to match the Resource Record.
- DNS Guard is enabled, so the ASA tears down the DNS session associated with a DNS query as soon as the DNS reply is forwarded by the ASA. The ASA also monitors the message exchange to ensure that the ID of the DNS reply matches the ID of the DNS query.
- Translation of the DNS record based on the NAT configuration is enabled.
- Protocol enforcement is enabled, which enables DNS message format check, including domain name length of no more than 255 characters, label length of 63 characters, compression, and looped pointer check.

**Note** Elevated CPU utilization is likely to occur in Cisco ASA when a large number of packets is inspected per second. As the number of packets being processed increases—especially when the device nears its maximum throughput or connection capacity—the CPU usage too rises. This occurs because ASA must perform more operations per second, which increases CPU resource consumption.

# Configure DNS Inspection Policy Map

You can create a DNS inspection policy map to customize DNS inspection actions if the default inspection behavior is not sufficient for your network.

You can optionally create a DNS inspection class map to define the traffic class for DNS inspection. The other option is to define the traffic classes directly in the DNS inspection policy map. The difference between creating a class map and defining the traffic match directly in the inspection map is that you can create more complex match criteria and you can reuse class maps. Although this procedure explains inspection maps, the matching criteria used in class maps are the same as those explained in the step relating to the **Inspection** tab. You can configure DNS class maps by selecting **Configuration** > **Firewall** > **Objects** > **Class Maps** > **DNS**, or by creating them while configuring the inspection map.

**Tip** You can configure inspection maps while creating service policies, in addition to the procedure explained below. The contents of the map are the same regardless of how you create it.

### Before you begin

Some traffic matching options use regular expressions for matching purposes. If you intend to use one of those techniques, first create the regular expression or regular expression class map.

### Procedure

**Step 1** Choose **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **DNS**.

**Step 2** Do one of the following:

- Click **Add** to add a new map.
- Select a map to view its contents. You can change the security level directly, or click **Customize** to edit the map. The remainder of the procedure assumes you are customizing or adding a map.

**Step 3** For new maps, enter a name (up to 40 characters) and description. When editing a map, you can change the description only.

**Step 4** In the **Security Level** view of the DNS Inspect Map dialog box, select the level that best matches your desired configuration. The default level is Low.

If one of the preset levels matches your requirements, you are now done. Just click **OK**, skip the rest of this procedure, and use the map in a service policy rule for DNS inspection.

If you need to customize the settings further, click **Details**, and continue with the procedure.

**Step 5** Click the **Protocol Conformance** tab and choose the desired options:

- **Enable DNS guard function**—Using DNS Guard, the ASA tears down the DNS session associated with a DNS query as soon as the DNS reply is forwarded by the ASA. The ASA also monitors the message exchange to ensure that the ID of the DNS reply matches the ID of the DNS query.
- **Enable NAT re-write function**—Translates the DNS record based on the NAT configuration.

- **Enable protocol enforcement**—Enables DNS message format check, including domain name length of no more than 255 characters, label length of 63 characters, compression, and looped pointer check.
- **Randomize the DNS identifier for DNS query**.
- **Enable TCP inspection**—Enables inspection of DNS over TCP traffic. Ensure that DNS/TCP port 53 traffic is part of the class to which you apply DNS inspection. The inspection default class includes TCP/53.
- **Enforce TSIG resource record to be present in DNS message**—You can drop or log non-conforming packets, and optionally log dropped packets.

**Step 6** Click the **Filtering** tab and choose the desired options.

- Global Settings—Choose whether to drop packets that exceed the specified maximum length regardless of whether they are from the client or server, from 512 to 65535 bytes.
- Server Settings—**Drop packets that exceed specified maximum length** and **Drop packets sent to server that exceed length indicated by the RR**—Sets the maximum server DNS message length, from 512 to 65535 bytes, or sets the maximum length to the value in the Resource Record. If you enable both settings, the lower value is used.
- Client Settings—**Drop packets that exceed specified maximum length** and **Drop packets sent to server that exceed length indicated by the RR**—Sets the maximum client DNS message length, from 512 to 65535 bytes, or sets the maximum length to the value in the Resource Record. If you enable both settings, the lower value is used.

**Step 7** Click the **Mismatch Rate** tab and choose whether to enable logging when the DNS ID mismatch rate exceeds the specified threshold. For example, you could set a threshold of 30 mismatches per 3 seconds.

**Step 8** Click the **Inspections** tab and define the specific inspections you want to implement based on traffic characteristics.

You can define traffic matching criteria based on DNS class maps, by configuring matches directly in the inspection map, or both.

a) Do any of the following:

- Click **Add** to add a new criterion.
- Select an existing criterion and click **Edit**.

b) Choose **Single Match** to define the criterion directly, or **Multiple Match**, in which case you select the DNS class map that defines the criteria.

c) If you are defining the criterion here, choose the match type for the criteria: **Match** (traffic must match the criterion) or **No Match** (traffic must not match the criterion). For example, if No Match is selected on the string “example.com,” then any traffic that contains “example.com” is excluded from the class map. Then, configure the criterion as follows:

- Header Flag—Select whether the flag should equal or contain the specified value, then either select the header flag name, or enter the hex value of the header (0x0 to 0xfff). If you select multiple header values, “equals” requires that all flags are present, “contains” that any one of the flags is present, in the packet. Header flag names are **AA** (Authoritative Answer), **QR** (Query), **RA** (Recursion Available), **RD** (Recursion Desired), **TC** (Truncation).

- Type—The DNS Type field name or value in the packet. Field names are **A** (IPv4 address), **AXFR** (full zone transfer), **CNAME** (canonical name), **IXFR** (incremental zone transfer), **NS** (authoritative name server), **SOA** (start of a zone of authority) or **TSIG** (transaction signature). Values are arbitrary numbers in the DNS Type field from 0 to 65535: either enter a specific value or a range of values.
- Class—The DNS Class field name or value in the packet. Internet is the only possible field name. Values are arbitrary numbers in the DNS Class field from 0 to 65535: either enter a specific value or a range of values.
- Question—The question portion of a DNS message.
- Resource Record—The DNS resource record. Choose whether to match the additional, answer, or authority resource record section.

d) Choose the primary action to take for matching traffic: drop packet, drop connection, mask (for Header Flag matches only) or none.
e) Choose whether to enable or disable logging. You must disable logging if you want to enforce TSIG.
f) Chose whether to enforce the presence of a TSIG resource record. You can drop the packet, log it, or drop and log it. Usually, you must select **Primary Action: None** and **Log: Disable** to enforce TSIG. However, for Header Flag matches, you can enforce TSIG along with the mask primary action.
g) Click **OK** to add the inspection. Repeat the process as needed.

**Step 9** Click the **Umbrella Connections** tab and enable the connection to Cisco Umbrella in the cloud.

The options on this tab work only if you configure the Cisco Umbrella connection on the **Configuration** > **Firewall** > **Objects** > **Umbrella** page. You must then configure the options on this tab to get the device to register with Cisco Umbrella, so that the device can redirect DNS lookups to Cisco Umbrella. Cisco Umbrella can then apply your FQDN-based security policies. For more information, see Cisco Umbrella.

- **Umbrella**—Enables Cisco Umbrella. You can optionally specify the name of the Cisco Umbrella policy to apply to the device in the **Umbrella Tag** field. If you do not specify a policy, the default policy is applied. After registration, the Umbrella device ID is displayed next to the tag.
- **Enable Dnscrypt**—Enables DNScrypt to encrypt connections between the device and Cisco Umbrella. Enabling DNScrypt starts the key-exchange thread with the Umbrella resolver. The key-exchange thread performs the handshake with the resolver every hour and updates the device with a new secret key. Because DNScrypt uses UDP/443, you must ensure that the class map used for DNS inspection includes that port. Note that the default inspection class already includes UDP/443 for DNS inspection.
- **Fail Open**—Enable fail open if you want DNS resolution to work if the Umbrella DNS server is unavailable. When failing open, if the Cisco Umbrella DNS server is unavailable, Umbrella disables itself on this policy map and allows DNS requests to go to the other DNS servers configured on the system, if any. When the Umbrella DNS servers are available again, the policy map resumes using them. If you do not select this option, DNS requests continue to go to the unreachable Umbrella resolver, so they will not get a response

**Step 10** Click **OK** in the DNS Inspect Map dialog box.

You can now use the inspection map in a DNS inspection service policy.

### What to do next

You can now configure an inspection policy to use the map. See Configure Application Layer Protocol Inspection.

# FTP Inspection

FTP inspection is enabled by default. You need to configure it only if you want non-default processing. The following sections describe the FTP inspection engine.

## FTP Inspection Overview

The FTP application inspection inspects the FTP sessions and performs four tasks:

- Prepares dynamic secondary data connection channels for FTP data transfer. Ports for these channels are negotiated through PORT or PASV commands. The channels are allocated in response to a file upload, a file download, or a directory listing event.
- Tracks the FTP command-response sequence.
- Generates an audit trail.
  - Audit record 303002 is generated for each file that is retrieved or uploaded.
  - Audit record 201005 is generated if the secondary dynamic channel preparation failed due to memory shortage.
- Translates the embedded IP address.



**Note** If you disable FTP inspection, outbound users can start connections only in passive mode, and all inbound FTP is disabled.

## Strict FTP

Strict FTP increases the security of protected networks by preventing web browsers from sending embedded commands in FTP requests. To enable strict FTP, click the **Configure** button next to FTP on the Configuration > Firewall > Service Policy Rules > Edit Service Policy Rule > Rule Actions > Protocol Inspection tab.

When you use strict FTP, you can optionally specify an FTP inspection policy map to specify FTP commands that are not permitted to pass through the ASA.

Strict FTP inspection enforces the following behavior:

- An FTP command must be acknowledged before the ASA allows a new command.
- The ASA drops connections that send embedded commands.
- The 227 and PORT commands are checked to ensure they do not appear in an error string.

**Caution** Using strict FTP may cause the failure of FTP clients that are not strictly compliant with FTP RFCs. Additionally, you must ensure you apply the inspection to your FTP ports only (TCP/21 is the normal FTP port). Strict FTP inspection applied to non-FTP traffic can result in unexpected traffic loss, especially HTTP traffic.

With strict FTP inspection, each FTP command and response sequence is tracked for the following anomalous activity:

- Truncated command—Number of commas in the PORT and PASV reply command is checked to see if it is five. If it is not five, then the PORT command is assumed to be truncated and the TCP connection is closed.
- Incorrect command—Checks the FTP command to see if it ends with <CR><LF> characters, as required by the RFC. If it does not, the connection is closed.
- Size of RETR and STOR commands—These are checked against a fixed constant. If the size is greater, then an error message is logged and the connection is closed.
- Command spoofing—The PORT command should always be sent from the client. The TCP connection is denied if a PORT command is sent from the server.
- Reply spoofing—PASV reply command (227) should always be sent from the server. The TCP connection is denied if a PASV reply command is sent from the client. This prevents the security hole when the user executes “227 xxxxx a1, a2, a3, a4, p1, p2.”
- TCP stream editing—The ASA closes the connection if it detects TCP stream editing.
- Invalid port negotiation—The negotiated dynamic port value is checked to see if it is less than 1024. As port numbers in the range from 1 to 1024 are reserved for well-known connections, if the negotiated port falls in this range, then the TCP connection is freed.
- Command pipelining—The number of characters present after the port numbers in the PORT and PASV reply command is cross checked with a constant value of 8. If it is more than 8, then the TCP connection is closed.
- The ASA replaces the FTP server response to the SYST command with a series of Xs to prevent the server from revealing its system type to FTP clients. To override this default behavior, use the **no mask-syst-reply** command in the FTP map.

# Configure an FTP Inspection Policy Map

FTP command filtering and security checks are provided using strict FTP inspection for improved security and control. Protocol conformance includes packet length checks, delimiters and packet format checks, command terminator checks, and command validation.

Blocking FTP based on user values is also supported so that it is possible for FTP sites to post files for download, but restrict access to certain users. You can block FTP connections based on file type, server name, and other attributes. System message logs are generated if an FTP connection is denied after inspection.

If you want FTP inspection to allow FTP servers to reveal their system type to FTP clients, and limit the allowed FTP commands, then create and configure an FTP inspection policy map. You can then apply the map when you enable FTP inspection.

You can optionally create an FTP inspection class map to define the traffic class for FTP inspection. The other option is to define the traffic classes directly in the FTP inspection policy map. The difference between creating a class map and defining the traffic match directly in the inspection map is that you can create more complex match criteria and you can reuse class maps. Although this procedure explains inspection maps, the matching criteria used in class maps are the same as those explained in the step relating to the **Inspection** tab. You can configure DNS class maps by selecting **Configuration** > **Firewall** > **Objects** > **Class Maps** > **FTP**, or by creating them while configuring the inspection map.

**Tip** You can configure inspection maps while creating service policies, in addition to the procedure explained below. The contents of the map are the same regardless of how you create it.

### Before you begin

Some traffic matching options use regular expressions for matching purposes. If you intend to use one of those techniques, first create the regular expression or regular expression class map.

### Procedure

**Step 1** Choose **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **FTP**.

**Step 2** Do one of the following:

- Click **Add** to add a new map.
- Select a map to view its contents. You can change the security level directly, or click **Customize** to edit the map. The remainder of the procedure assumes you are customizing or adding a map.

**Step 3** For new maps, enter a name (up to 40 characters) and description. When editing a map, you can change the description only.

**Step 4** In the **Security Level** view of the FTP Inspect Map dialog box, select the level that best matches your desired configuration. The default level is High.

If one of the preset levels matches your requirements, you are now done. Just click **OK**, skip the rest of this procedure, and use the map in a service policy rule for FTP inspection.

If you need to customize the settings further, click **Details**, and continue with the procedure.

**Tip**
The **File Type Filtering** button is a shortcut to configure file media or MIME type inspection, which is explained later in this procedure.

**Step 5** Click the **Parameters** tab and choose whether to mask the greeting banner from the server or mask the reply to the SYST command.

Masking these items prevents the client from discovering server information that might be helpful in an attack.

**Step 6** Click the **Inspections** tab and define the specific inspections you want to implement based on traffic characteristics.

You can define traffic matching criteria based on FTP class maps, by configuring matches directly in the inspection map, or both.

a) Do any of the following:
   - Click **Add** to add a new criterion.
   - Select an existing criterion and click **Edit**.

b) Choose **Single Match** to define the criterion directly, or **Multiple Match**, in which case you select the FTP class map that defines the criteria.
c) If you are defining the criterion here, choose the match type for the criteria: **Match** (traffic must match the criterion) or **No Match** (traffic must not match the criterion). For example, if No Match is selected on the string "example.com," then any traffic that contains "example.com" is excluded from the class map. Then, configure the criterion as follows:
   - File Name—Match the name of the file being transferred against the selected regular expression or regular expression class.
   - File Type—Match the MIME or media type of the file being transferred against the selected regular expression or regular expression class.
   - Server—Match the FTP server name against the selected regular expression or regular expression class.
   - User—Match the name of the logged-in user against the selected regular expression or regular expression class.
   - Request Command—The FTP command used in the packet, any combination of the following:
     - **APPE**—Append to a file.
     - **CDUP**—Changes to the parent directory of the current working directory.
     - **DELE**—Delete a file on the server.
     - **GET**—Gets a file from the server.
     - **HELP**—Provides help information.
     - **MKD**—Makes a directory on the server.
     - **PUT**—Sends a file to the server.
     - **RMD**—Deletes a directory on the server.
     - **RNFR**—Specifies the "rename-from" filename.
     - **RNTO**—Specifies the "rename-to" filename.
     - **SITE**—Used to specify a server-specific command. This is usually used for remote administration.
     - **STOU**—Stores a file using a unique file name.

d) Choose whether to enable or disable logging. The action is always to reset the connection, which drops the packet, closes the connection, and sends a TCP reset to the server or client.
e) Click **OK** to add the inspection. Repeat the process as needed.

**Step 7** Click **OK** in the FTP Inspect Map dialog box.

You can now use the inspection map in a FTP inspection service policy.

### What to do next

You can now configure an inspection policy to use the map. See Configure Application Layer Protocol Inspection.

# HTTP Inspection

HTTP inspection is not enabled in the default inspection policy, so you must enable it if you need this inspection. However, the default inspect class does include the default HTTP ports, so you can simply edit the default global inspection policy to add HTTP inspection. You can alternatively create a new service policy as desired, for example, an interface-specific policy.

The following sections describe the HTTP inspection engine.

## HTTP Inspection Overview

Use the HTTP inspection engine to protect against specific attacks and other threats that are associated with HTTP traffic.

HTTP application inspection scans HTTP headers and body, and performs various checks on the data. These checks prevent various HTTP constructs, content types, and tunneling and messaging protocols from traversing the security appliance.

The enhanced HTTP inspection feature, which is also known as an application firewall and is available when you configure an HTTP inspection policy map, can help prevent attackers from using HTTP messages for circumventing network security policy.

HTTP application inspection can block tunneled applications and non-ASCII characters in HTTP requests and responses, preventing malicious content from reaching the web server. Size limiting of various elements in HTTP request and response headers, URL blocking, and HTTP server header type spoofing are also supported.

Enhanced HTTP inspection verifies the following for all HTTP messages:

- Conformance to RFC 2616
- Use of RFC-defined methods only.
- Compliance with the additional criteria.

## Configure an HTTP Inspection Policy Map

To specify actions when a message violates a parameter, create an HTTP inspection policy map. You can then apply the inspection policy map when you enable HTTP inspection.

You can optionally create an HTTP inspection class map to define the traffic class for HTTP inspection. The other option is to define the traffic classes directly in the HTTP inspection policy map. The difference between creating a class map and defining the traffic match directly in the inspection map is that you can create more complex match criteria and you can reuse class maps. Although this procedure explains inspection maps, the

matching criteria used in class maps are the same as those explained in the step relating to the **Inspection** tab. You can configure HTTP class maps by selecting **Configuration** > **Firewall** > **Objects** > **Class Maps** > **HTTP**, or by creating them while configuring the inspection map.

**Tip** You can configure inspection maps while creating service policies, in addition to the procedure explained below. The contents of the map are the same regardless of how you create it.

### Before you begin

Some traffic matching options use regular expressions for matching purposes. If you intend to use one of those techniques, first create the regular expression or regular expression class map.

## Procedure

**Step 1** Choose **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **HTTP**.

**Step 2** Do one of the following:

- Click **Add** to add a new map.
- Select a map to view its contents. You can change the security level directly, or click **Customize** to edit the map. The remainder of the procedure assumes you are customizing or adding a map.

**Step 3** For new maps, enter a name (up to 40 characters) and description. When editing a map, you can change the description only.

**Step 4** In the **Security Level** view of the HTTP Inspect Map dialog box, select the level that best matches your desired configuration. The default level is Low.

If one of the preset levels matches your requirements, you are now done. Just click **OK**, skip the rest of this procedure, and use the map in a service policy rule for HTTP inspection.

If you need to customize the settings further, click **Details**, and continue with the procedure.

**Tip**
The **URI Filtering** button is a shortcut to configure Request URI inspection, which is explained later in this procedure.

**Step 5** Click the **Parameters** tab and configure the desired options.

- **Body Match Maximum**—The maximum number of characters in the body of an HTTP message that should be searched in a body match. Default is 200 bytes. A large number will have a significant impact on performance.
- **Check for protocol violations**—Whether to verify that packets conform to the HTTP protocol. For violations, you can drop the connection, reset it, or log it. When dropping or resetting, you can also enable logging.
- **Spoof server string**—Replaces the server HTTP header value with the specified string, up to 82 characters.

**Step 6** Click the **Inspections** tab and define the specific inspections you want to implement based on traffic characteristics.

You can define traffic matching criteria based on HTTP class maps, by configuring matches directly in the inspection map, or both.

a) Do any of the following:

- Click **Add** to add a new criterion.
- Select an existing criterion and click **Edit**.

b) Choose **Single Match** to define the criterion directly, or **Multiple Match**, in which case you select the HTTP class map that defines the criteria.

c) If you are defining the criterion here, choose the match type for the criteria: **Match** (traffic must match the criterion) or **No Match** (traffic must not match the criterion). For example, if No Match is selected on the string "example.com," then any traffic that contains "example.com" is excluded from the class map. Then, configure the criterion as follows:

- Request/Response Content Type Mismatch—Match packets where the content type in the response does not match one of the MIME types in the accept field of the request.
- Request Arguments—Match the arguments of the request against the selected regular expression or regular expression class.
- Request Body Length—Match packets where the body of the request is greater than the specified number of bytes.
- Request Body—Match the body of the request against the selected regular expression or regular expression class.
- Request Header Field Count—Match packets where the number of header fields in the request is greater than the specified count. You can match the field header type to a regular expression or to a predefined type. The predefined types are: accept, accept-charset, accept-encoding, accept-language, allow, authorization, cache-control, connection, content-encoding, content-language, content-length, content-location, content-md5, content-range, content-type, cookie, date, expect, expires, from, host, if-match, if-modified-since, if-none-match, if-range, if-unmodified-since, last-modified, max-forwards, pragma, proxy-authorization, range, referer, te, trailer, transfer-encoding, upgrade, user-agent, via, warning.
- Request Header Field Length—Match packets where the length of the header field in the request is greater than the specified bytes. You can match the field header type to a regular expression or to a predefined type. The predefined types are listed above for Request Header Field Count.
- Request Header Field—Match the content of the selected header field in the request against the selected regular expression or regular expression class. You can specify a predefined header type or use a regular expression to select the headers.
- Request Header Count—Match packets where the number of headers in the request is greater than the specified number.
- Request Header Length—Match packets where the length of the header in the request is greater than the specified bytes.
- Request Header Non-ASCII—Match packets where the header in the request contains non-ASCII characters.

- Request Method—Match packets where the request method matches the predefined type or the selected regular expression or regular expression class. The predefined types are: bcopy, bdelete, bmove, bpropfind, bproppatch, connect, copy, delete, edit, get, getattribute, getattributenames, getproperties, head, index, lock, mkcol, mkdir, move, notify, options, poll, post, propfind, proppatch, put, revadd, revlabel, revlog, revnum, save, search, setattribute, startrev, stoprev, subscribe, trace, unedit, unlock, unsubscribe.
- Request URI Length—Match packets where the length of the URI of the request is greater than the specified bytes.
- Request URI—Match the content of the URI of the request against the selected regular expression or regular expression class.
- Request Body—Match the body of the request against the selected regular expression or regular expression class, or to ActiveX or Java Applet content.
- Response Body Length—Match packets where the length of the body of the response is greater than the specified bytes.
- Response Header Field Count—Match packets where the number of header fields in the response is greater than the specified count. You can match the field header type to a regular expression or to a predefined type. The predefined types are: accept-ranges, age, allow, cache-control, connection, content-encoding, content-language, content-length, content-location, content-md5, content-range, content-type, date, etag, expires, last-modified, location, pragma, proxy-authenticate, retry-after, server, set-cookie, trailer, transfer-encoding, upgrade, vary, via, warning, www-authenticate.
- Response Header Field Length—Match packets where the length of the header field in the response is greater than the specified bytes. You can match the field header type to a regular expression or to a predefined type. The predefined types are listed above for Response Header Field Count.
- Response Header Field—Match the content of the selected header field in the response against the selected regular expression or regular expression class. You can specify a predefined header type or use a regular expression to select the headers.
- Response Header Count—Match packets where the number of headers in the response is greater than the specified number.
- Response Header Length—Match packets where the length of the header in the response is greater than the specified bytes.
- Response Header Non-ASCII—Match packets where the header in the response contains non-ASCII characters.
- Response Status Line—Match the content of the response status line against the selected regular expression or regular expression class.

d) Choose whether to drop the connection, reset it, or log it. For drop connection and reset, you can enable or disable logging.
e) Click **OK** to add the inspection. Repeat the process as needed.

**Step 7** Click **OK** in the HTTP Inspect Map dialog box.

You can now use the inspection map in a HTTP inspection service policy.

### What to do next

You can now configure an inspection policy to use the map. See Configure Application Layer Protocol Inspection.

# ICMP Inspection

The ICMP inspection engine allows ICMP traffic to have a "session" so it can be inspected like TCP and UDP traffic. Without the ICMP inspection engine, we recommend that you do not allow ICMP through the ASA in an ACL. Without stateful inspection, ICMP can be used to attack your network. The ICMP inspection engine ensures that there is only one response for each request, and that the sequence number is correct.

However, ICMP traffic directed to an ASA interface is never inspected, even if you enable ICMP inspection. Thus, a ping (echo request) to an interface can fail under specific circumstances, such as when the echo request comes from a source that the ASA can reach through a backup default route.

**Note** NAT uses ICMP inspection when translating packets even if you disable ICMP inspection.

For information on enabling ICMP inspection, see Configure Application Layer Protocol Inspection.

# ICMP Error Inspection

When ICMP Error inspection is enabled, the ASA creates translation sessions for intermediate hops that send ICMP error messages, based on the NAT configuration. The ASA overwrites the packet with the translated IP addresses.

When disabled, the ASA does not create translation sessions for intermediate nodes that generate ICMP error messages. ICMP error messages generated by the intermediate nodes between the inside host and the ASA reach the outside host without consuming any additional NAT resource. This is undesirable when an outside host uses the traceroute command to trace the hops to the destination on the inside of the ASA. When the ASA does not translate the intermediate hops, all the intermediate hops appear with the mapped destination IP address.

**Note** You should always enable ICMP Error inspection if there is a possibility that NAT will be used on ICMP packets. Because NAT automatically uses ICMP inspection for ICMP packets, even if you have ICMP inspection disabled, the use of the mapped destination address as the source address can make it look like a scanner is examining your network. For example, without ICMP Error inspection also enabled, if the echo request packet has its destination translated, when it is embedded in a ICMP time exceeded response, the outer header of the time exceeded request uses the translated destination as the source address. If you enable ICMP Error inspection, the time exceeded source address will be set to the correct value.

For information on enabling ICMP Error inspection, see Configure Application Layer Protocol Inspection.

# ILS Inspection

The Internet Locator Service (ILS) inspection engine provides NAT support for Microsoft NetMeeting, SiteServer, and Active Directory products that use LDAP to exchange directory information with an ILS server. You cannot use PAT with ILS inspection because only IP addresses are stored by an LDAP database.

For search responses, when the LDAP server is located outside, consider using NAT to allow internal peers to communicate locally while registered to external LDAP servers. If you do not need to use NAT, we recommend that you turn off the inspection engine to provide better performance.

Additional configuration may be necessary when the ILS server is located inside the ASA border. This would require a hole for outside clients to access the LDAP server on the specified port, typically TCP 389.



**Note** Because ILS traffic (H225 call signaling) only occurs on the secondary UDP channel, the TCP connection is disconnected after the TCP inactivity interval. By default, this interval is 60 minutes and can be adjusted using the TCP **timeout** command. In ASDM, this is on the **Configuration > Firewall > Advanced > Global Timeouts** pane.

ILS inspection has the following limitations:

- Referral requests and responses are not supported.
- Users in multiple directories are not unified.
- Single users having multiple identities in multiple directories cannot be recognized by NAT.

For information on enabling ILS inspection, see Configure Application Layer Protocol Inspection.

# Instant Messaging Inspection

The Instant Messaging (IM) inspect engine lets you control the network usage of IM and stop leakage of confidential data, propagation of worms, and other threats to the corporate network.

IM inspection is not enabled in the default inspection policy, so you must enable it if you need this inspection. However, the default inspect class does include the default IM ports, so you can simply edit the default global inspection policy to add IM inspection. You can alternatively create a new service policy as desired, for example, an interface-specific policy.

If you decide to implement IM inspection, you can also configure an IM inspection policy map to specify actions when a message violates a parameter. The following procedure explains IM inspection policy maps.

You can optionally create an IM inspection class map to define the traffic class for IM inspection. The other option is to define the traffic classes directly in the IM inspection policy map. The difference between creating a class map and defining the traffic match directly in the inspection map is that you can create more complex match criteria and you can reuse class maps. This procedure explains inspection maps, but class maps are essentially the same, except that you do not specify the actions for matching traffic. You can configure IM class maps by selecting **Configuration > Firewall > Objects > Class Maps > Instant Messaging (IM)**.

**Tip** You can configure inspection maps while creating service policies, in addition to the procedure explained below. The contents of the map are the same regardless of how you create it.

### Before you begin

Some traffic matching options use regular expressions for matching purposes. If you intend to use one of those techniques, first create the regular expression or regular expression class map.

### Procedure

**Step 1** Choose **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **Instant Messaging (IM)**.

**Step 2** Do one of the following:

- Click **Add** to add a new map.
- Select a map and click **Edit**.

**Step 3** For new maps, enter a name (up to 40 characters) and description. When editing a map, you can change the description only.

**Step 4** Define the specific inspections you want to implement based on traffic characteristics.

You can define traffic matching criteria based on IM class maps, by configuring matches directly in the inspection map, or both.

a) Do any of the following:

- Click **Add** to add a new criterion.
- Select an existing criterion and click **Edit**.

b) Choose **Single Match** to define the criterion directly, or **Multiple Match**, in which case you select the IM class map that defines the criteria. Click **Manage** to create new class maps.

c) If you are defining the criterion here, choose the match type for the criteria: **Match** (traffic must match the criterion) or **No Match** (traffic must not match the criterion). For example, if No Match is selected on the string “example.com,” then any traffic that contains “example.com” is excluded from the class map. Then, configure the criterion.

- Protocol—Match traffic of a specific IM protocol, such as Yahoo Messenger or MSN Messenger.
- Service—Match a specific IM service, such as chat, file transfer, web cam, voice chat, conference, or games.
- Version—Match the version of the IM message against the selected regular expression or regular expression class.
- Client Login Name—Match the source client login name of the IM message against the selected regular expression or regular expression class.
- Client Peer Login Name—Match the destination peer login name of the IM message against the selected regular expression or regular expression class.

- Source IP Address—Match the source IP address and mask.
- Destination IP Address—Match the destination IP address and mask.
- Filename—Match the filename of the IM message against the selected regular expression or regular expression class.

d) Choose whether to drop the connection, reset it, or log it. For drop connection and reset, you can enable or disable logging.
e) Click **OK** to add the inspection. Repeat the process as needed.

**Step 5** Click **OK** in the IM Inspect Map dialog box.

You can now use the inspection map in a IM inspection service policy.

### What to do next

You can now configure an inspection policy to use the map. See Configure Application Layer Protocol Inspection.

# IP Options Inspection

You can configure IP Options inspection to control which IP packets are allowed based on the contents of the IP Options field in the packet header. You can drop packets that have unwanted options, clear the options (and allow the packet), or allow the packet without change.

IP options provide control functions that are required in some situations but unnecessary for most common communications. In particular, IP options include provisions for time stamps, security, and special routing. Use of IP Options is optional, and the field can contain zero, one, or more options.

For a list of IP options, with references to the relevant RFCs, see the IANA page, http://www.iana.org/assignments/ip-parameters/ip-parameters.xhtml.

IP options inspection is enabled by default, but for RSVP traffic only. You need to configure it only if you want to allow additional options than the default map allows, or if you want to apply it to other types of traffic by using a non-default inspection traffic class map.

**Note** IP options inspection does not work on fragmented packets. For example, options are not cleared from fragments.

The following sections describe IP Options inspection.

## Defaults for IP Options Inspection

IP Options inspection is enabled by default for RSVP traffic only, using the _default_ip_options_map inspection policy map.

- The Router Alert option is allowed.

This option notifies transit routers to inspect the contents of the packet even when the packet is not destined for that router. This inspection is valuable when implementing RSVP and similar protocols that require relatively complex processing from the routers along the packet's delivery path. Dropping RSVP packets containing the Router Alert option can cause problems in VoIP implementations.

- Packets that contain any other options are dropped.

  Each time a packet is dropped due to inspection, syslog 106012 is issued. The message shows which option caused the drop. Use the **show service-policy inspect ip-options** command to view statistics for each option.

# Configure an IP Options Inspection Policy Map

If you want to perform non-default IP options inspection, create an IP options inspection policy map to specify how you want to handle each option type.

**Tip** You can configure inspection maps while creating service policies, in addition to the procedure explained below. The contents of the map are the same regardless of how you create it.

### Procedure

**Step 1** Choose **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **IP Options**.

**Step 2** Do one of the following:

- Click **Add** to add a new map.
- Select a map and click **Edit**.

**Step 3** For new maps, enter a name (up to 40 characters) and description. When editing a map, you can change the description only.

**Step 4** Choose which options you want to allow by moving them from the Drop list to the Allow list.

Consider the following tips:

- The "default" option sets the default behavior for options not included in the map. If you move it to the Allowed list, even options shown in the Drop list will be allowed.
- For any option you allow, you can check the Clear box to remove the option from the packet header before transmitting the packet.
- Some options are listed by option type number. The number is the whole option type octet (copy, class, and option number), not just the option number portion of the octet. These option types might not represent real options. Non-standard options must be in the expected type-length-value format defined in the Internet Protocol RFC 791, http://tools.ietf.org/html/rfc791.
- If a packet includes more than one type of option, it is dropped so long as the action for one of those types is to drop the packet.

For a list of IP options, with references to the relevant RFCs, see the IANA page, http://www.iana.org/assignments/ip-parameters/ip-parameters.xhtml.

**Step 5** Click **OK**.

You can now use the inspection map in an IP options inspection service policy.

### What to do next

You can now configure an inspection policy to use the map. See Configure Application Layer Protocol Inspection.

# IPsec Pass Through Inspection

IPsec Pass Through inspection is not enabled in the default inspection policy, so you must enable it if you need this inspection. However, the default inspect class does include the default IPsec ports, so you can simply edit the default global inspection policy to add IPsec inspection. You can alternatively create a new service policy as desired, for example, an interface-specific policy.

The following sections describe the IPsec Pass Through inspection engine.

## IPsec Pass Through Inspection Overview

Internet Protocol Security (IPsec) is a protocol suite for securing IP communications by authenticating and encrypting each IP packet of a data stream. IPsec also includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used to protect data flows between a pair of hosts (for example, computer users or servers), between a pair of security gateways (such as routers or firewalls), or between a security gateway and a host.

IPsec Pass Through application inspection provides convenient traversal of ESP (IP protocol 50) and AH (IP protocol 51) traffic associated with an IKE UDP port 500 connection. It avoids lengthy ACL configuration to permit ESP and AH traffic and also provides security using timeout and max connections.

Configure a policy map for IPsec Pass Through to specify the restrictions for ESP or AH traffic. You can set the per client max connections and the idle timeout.

NAT and non-NAT traffic is permitted. However, PAT is not supported.

## Configure an IPsec Pass Through Inspection Policy Map

An IPsec Pass Through map lets you change the default configuration values used for IPsec Pass Through application inspection. You can use an IPsec Pass Through map to permit certain flows without using an ACL.

The configuration includes a default map, _default_ipsec_passthru_map, that sets no maximum limit on ESP connections per client, and sets the ESP idle timeout at 10 minutes. You need to configure an inspection policy map only if you want different values, or if you want to set AH values.

**Tip** You can configure inspection maps while creating service policies, in addition to the procedure explained below. The contents of the map are the same regardless of how you create it.

**Procedure**

**Step 1** Choose **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **IPsec Pass Through**.

**Step 2** Do one of the following:

- Click **Add** to add a new map.
- Select a map to view its contents. You can change the security level directly, or click **Customize** to edit the map. The remainder of the procedure assumes you are customizing or adding a map.

**Step 3** For new maps, enter a name (up to 40 characters) and description. When editing a map, you can change the description only.

**Step 4** In the **Security Level** view of the IPsec Pass Through Inspect Map dialog box, select the level that best matches your desired configuration.

If one of the preset levels matches your requirements, you are now done. Just click **OK**, skip the rest of this procedure, and use the map in a service policy rule for IPsec Pass Through inspection.

If you need to customize the settings further, click **Details**, and continue with the procedure.

**Step 5** Choose whether to allow ESP and AH tunnels.

For each protocol, you can also set the maximum number of connections allowed per client, and the idle timeout.

**Step 6** Click **OK**.

You can now use the inspection map in an IPsec Pass Through inspection service policy.

# IPv6 Inspection

IPv6 inspection lets you selectively log or drop IPv6 traffic based on the extension header. In addition, IPv6 inspection can check conformance to RFC 2460 for type and order of extension headers in IPv6 packets.

IPv6 inspection is not enabled in the default inspection policy, so you must enable it if you need this inspection. You can simply edit the default global inspection policy to add IPv6 inspection. You can alternatively create a new service policy as desired, for example, an interface-specific policy.

## Defaults for IPv6 Inspection

If you enable IPv6 inspection and do not specify an inspection policy map, then the default IPv6 inspection policy map is used, and the following actions are taken:

- Allows only known IPv6 extension headers. Non-conforming packets are dropped and logged.
- Enforces the order of IPv6 extension headers as defined in the RFC 2460 specification. Non-conforming packets are dropped and logged.
- Drops any packet with a routing type header.

# Configure an IPv6 Inspection Policy Map

To identify extension headers to drop or log, or to disable packet verification, create an IPv6 inspection policy map to be used by the service policy.

**Procedure**

**Step 1** Choose **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **IPv6**.

**Step 2** Do one of the following:

- Click **Add** to add a new map.
- Select a map and click **Edit**.

**Step 3** For new maps, enter a name (up to 40 characters) and description. When editing a map, you can change the description only.

**Step 4** Click the **Enforcement** tab and choose whether to permit only known IPv6 extension headers or to enforce the order of IPv6 extension headers as defined in RFC 2460. Non-conforming packets are dropped and logged.

**Step 5** (Optional) Click the **Header Matches** tab to identify traffic to drop or log based on the headers in IPv6 messages.

a) Do any of the following:

- Click **Add** to add a new criterion.
- Select an existing criterion and click **Edit**.

b) Choose the IPv6 extension header to match:

- Authentication (AH) header.
- Destination Options header.
- Encapsulating Security Payload (ESP) header.
- Fragment header.
- Hop-by-Hop Options header.
- Routing header—Specify either a single header type number or a range of numbers.
- Header count—Specify the maximum number of extension headers you will allow without dropping or logging the packet.
- Routing header address count—Specify the maximum number of addresses in the type 0 routing header you will allow without dropping or logging the packet.

c) Choose whether to drop or log the packet. If you drop the packet, you can also enable logging.
d) Click **OK** to add the inspection. Repeat the process as needed.

**Step 6** Click **OK** in the IPv6 Inspect Map dialog box.

You can now use the inspection map in an IPv6 inspection service policy.

### What to do next

You can now configure an inspection policy to use the map. See Configure Application Layer Protocol Inspection.

# NetBIOS Inspection

NetBIOS application inspection performs NAT for the embedded IP address in the NetBIOS name service (NBNS) packets and NetBIOS datagram services packets. It also enforces protocol conformance, checking the various count and length fields for consistency.

NetBIOS inspection is enabled by default. You can optionally create a policy map to drop or log NetBIOS protocol violations. The following procedure explains how to configure a NetBIOS inspection policy map.

### Procedure

**Step 1** Choose **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **NetBIOS**.

**Step 2** Do one of the following:

- Click **Add** to add a new map.
- Select a map and click **Edit**.

**Step 3** For new maps, enter a name (up to 40 characters) and description. When editing a map, you can change the description only.

**Step 4** Select **Check for Protocol Violations**. There is no reason to create a map if you do not select this option.

**Step 5** Select the action to take, either to drop the packet or log it. If you drop the packet, you can also enable logging.

**Step 6** Click **OK**.

You can now use the inspection map in a NetBIOS inspection service policy.

### What to do next

You can now configure an inspection policy to use the map. See Configure Application Layer Protocol Inspection.

# PPTP Inspection

PPTP is a protocol for tunneling PPP traffic. A PPTP session is composed of one TCP channel and usually two PPTP GRE tunnels. The TCP channel is the control channel used for negotiating and managing the PPTP GRE tunnels. The GRE tunnels carry PPP sessions between the two hosts.

When enabled, PPTP application inspection inspects PPTP protocol packets and dynamically creates the GRE connections and xlates necessary to permit PPTP traffic.

Specifically, the ASA inspects the PPTP version announcements and the outgoing call request/response sequence. Only PPTP Version 1, as defined in RFC 2637, is inspected. Further inspection on the TCP control channel is disabled if the version announced by either side is not Version 1. In addition, the outgoing-call request and reply sequence are tracked. Connections and xlates are dynamically allocated as necessary to permit subsequent secondary GRE data traffic.

The PPTP inspection engine must be enabled for PPTP traffic to be translated by PAT. Additionally, PAT is only performed for a modified version of GRE (RFC2637) and only if it is negotiated over the PPTP TCP control channel. PAT is not performed for the unmodified version of GRE (RFC 1701 and RFC 1702).

For information on enabling PPTP inspection, see Configure Application Layer Protocol Inspection.

# RSH Inspection

RSH inspection is enabled by default. The RSH protocol uses a TCP connection from the RSH client to the RSH server on TCP port 514. The client and server negotiate the TCP port number where the client listens for the STDERR output stream. RSH inspection supports NAT of the negotiated port number if necessary.

For information on enabling RSH inspection, see Configure Application Layer Protocol Inspection.

# SMTP and Extended SMTP Inspection

ESMTP inspection detects attacks, including spam, phising, malformed message attacks, and buffer overflow/underflow attacks. It also provides support for application security and protocol conformance, which enforces the sanity of the ESMTP messages as well as block senders/receivers, and block mail relay.

ESMTP inspection is enabled by default. You need to configure it only if you want different processing than that provided by the default inspection map.

The following sections describe the ESMTP inspection engine.

## SMTP and ESMTP Inspection Overview

Extended SMTP (ESMTP) application inspection provides improved protection against SMTP-based attacks by restricting the types of SMTP commands that can pass through the ASA and by adding monitoring capabilities. ESMTP is an enhancement to the SMTP protocol and is similar is most respects to SMTP.

ESMTP application inspection controls and reduces the commands that the user can use as well as the messages that the server returns. ESMTP inspection performs three primary tasks:

- Restricts SMTP requests to seven basic SMTP commands and eight extended commands. Supported commands are the following:

- Extended SMTP—AUTH, EHLO, ETRN, HELP, SAML, SEND, SOML, STARTTLS, and VRFY.
- SMTP (RFC 821)—DATA, HELO, MAIL, NOOP, QUIT, RCPT, RSET.

- Monitors the SMTP command-response sequence.
- Generates an audit trail—Audit record 108002 is generated when an invalid character embedded in the mail address is replaced. For more information, see RFC 821.

ESMTP inspection monitors the command and response sequence for the following anomalous signatures:

- Truncated commands.
- Incorrect command termination (not terminated with <CR><LR>).
- The MAIL and RCPT commands specify who are the sender and the receiver of the mail. Mail addresses are scanned for strange characters. The pipeline character (|) is deleted (changed to a blank space) and “<” ,”>” are only allowed if they are used to define a mail address (“>” must be preceded by “<”).
- Unexpected transition by the SMTP server.
- For unknown or unsupported commands, the inspection engine changes all the characters in the packet to X, which are rejected by the internal server. This results in a message such as “500 Command unknown: 'XXX'.” Incomplete commands are discarded

  Unsupported ESMTP commands are ATRN, ONEX, VERB, CHUNKING, and private extensions..
- TCP stream editing.
- Command pipelining.



**Note** With ESMTP inspection enabled, a Telnet session used for interactive SMTP may hang if the following rules are not observed: SMTP commands must be at least four characters in length; they must be terminated with carriage return and line feed; and you must wait for a response before issuing the next reply.

# Defaults for ESMTP Inspection

ESMTP inspection is enabled by default, using the _default_esmtp_map inspection policy map.

- The server banner is masked. The ESMTP inspection engine changes the characters in the server SMTP banner to asterisks except for the “2”, “0”, “0” characters. Carriage return (CR) and linefeed (LF) characters are ignored.
- Encrypted connections are allowed but not inspected.
- Special characters in sender and receiver address are not noticed, no action is taken.
- Connections with command line length greater than 512 are dropped and logged.
- Connections with more than 100 recipients are dropped and logged.
- Messages with body length greater than 998 bytes are logged.
- Connections with header line length greater than 998 are dropped and logged.

- Messages with MIME filenames greater than 255 characters are dropped and logged.
- EHLO reply parameters matching “others” are masked.

# Configure an ESMTP Inspection Policy Map

To specify actions when a message violates a parameter, create an ESMTP inspection policy map. You can then apply the inspection policy map when you enable ESMTP inspection.

### Before you begin

Some traffic matching options use regular expressions for matching purposes. If you intend to use one of those techniques, first create the regular expression or regular expression class map.

### Procedure

**Step 1** Choose **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **ESMTP**.

**Step 2** Do one of the following:

- Click **Add** to add a new map.
- Select a map to view its contents. You can change the security level directly, or click **Customize** to edit the map. The remainder of the procedure assumes you are customizing or adding a map.

**Step 3** For new maps, enter a name (up to 40 characters) and description. When editing a map, you can change the description only.

**Step 4** In the **Security Level** view of the ESMTP Inspect Map dialog box, select the level that best matches your desired configuration.

If one of the preset levels matches your requirements, you are now done. Just click **OK**, skip the rest of this procedure, and use the map in a service policy rule for ESMTP inspection.

If you need to customize the settings further, click **Details**, and continue with the procedure.

**Tip**
The **MIME File Type Filtering** button is a shortcut to configure file type inspection, which is explained later in this procedure.

**Step 5** Click the **Parameters** tab and configure the desired options.

- **Mask Server Banner**—Whether to mask the banner from the ESMTP server.
- **Encrypted Packet Inspection**—Whether to allow ESMTP over TLS (encrypted connections) without inspection. You can optionally log encrypted connections. The default is to allow TLS sessions without inspection. If you deselect the option, the system strips the STARTTLS indication from any encrypted session connection attempt and forces a plain-text connection.

**Step 6** Click the **Filtering** tab and configure the desired options.

- **Configure mail relay**—Identifies a domain name for mail relay. You can either drop the connection and optionally log it, or log it.

- **Check for special characters**—Identifies the action to take for messages that include the special characters pipe (|), back quote, and NUL in the sender or receiver email addresses. You can either drop the connection and optionally log it, or log it.

**Step 7** Click the **Inspections** tab and define the specific inspections you want to implement based on traffic characteristics.

a) Do any of the following:

- Click **Add** to add a new criterion.
- Select an existing criterion and click **Edit**.

b) Choose the match type for the criteria: **Match** (traffic must match the criterion) or **No Match** (traffic must not match the criterion). For example, if No Match is selected on the string "example.com," then any traffic that contains "example.com" is excluded from the class map. Then, configure the criterion:

- **Body Length**—Matches messages where the length of an ESMTP body message is greater than the specified number of bytes.
- **Body Line Length**—Matches messages where the length of a line in an ESMTP body message is greater than the specified number of bytes.
- **Commands**—Matches the command verb in the message. You can specify one or more of the following commands: auth, data, ehlo, etrn, helo, help, mail, noop, quit, rcpt, rset, saml, soml, vrfy.
- **Command Recipient Count**—Matches messages where the number of recipients is greater than the specified count.
- **Command Line Length**—Matches messages where the length of a line in the command verb is greater than the specified number of bytes.
- **EHLO Reply Parameters**—Matches ESMTP EHLO reply parameters. You can specify one or more of the following parameters: 8bitmime, auth, binaryname, checkpoint, dsn, etrn, others, pipelining, size, vrfy.
- **Header Length**—Matches messages where the length of an ESMTP header is greater than the specified number of bytes.
- **Header Line Length**—Matches messages where the length of a line in an ESMTP header is greater than the specified number of bytes.
- **Header To: Fields Count**—Matches messages where the number of To fields in the header is greater than the specified number.
- **Invalid Recipients Count**—Matches messages where the number of invalid recipients is greater than the specified count.
- **MIME File Type**—Matches the MIME or media file type against the specified regular expression or regular expression class.
- **MIME Filename Length**—Matches messages where a file name is longer than the specified number of bytes.
- **MIME Encoding**—Matches the MIME encoding type. You can specify one or more of the following types: 7bit, 8bit, base64, binary, others, quoted-printable.

- **Sender Address**—Matches the sender email address against the specified regular expression or regular expression class.
- **Sender Address Length**—Matches messages where the sender address is greater than the specified number of bytes.

c) Choose whether to drop the connection, reset it, or log it. For drop connection and reset, you can enable or disable logging. For command and EHLO reply parameter matching, you can also mask the command. For command matching, you can also apply a rate limit in packets per second.
d) Click **OK** to add the inspection. Repeat the process as needed.

**Step 8** Click **OK** in the ESMTP Inspect Map dialog box.

You can now use the inspection map in a ESMTP inspection service policy.

### What to do next

You can now configure an inspection policy to use the map. See Configure Application Layer Protocol Inspection.

# SNMP Inspection

SNMP application inspection is applied to both to-the-device and through-the-device traffic. This inspection is necessary if you configure SNMP v3 where users are limited to specific SNMP hosts. Without the inspection, a defined v3 user can poll the device from any allowed host. SNMP inspection is enabled by default for the default ports, so you need to configure it only if you use non-default ports. The default ports are UDP/161, 162 (for all device types) and UDP/4161 for devices that also run FXOS, as FXOS listens on UDP/161.

By default, the SNMP inspection limits the polling to the configured version.

**Note** If you configure SNMP on a device that also runs FXOS, SNMP inspection is mandatory, and is re-enabled if you disable it. SNMP inspection is enabled on a traffic class map that includes port UDP/4161.

Optionally, you can further restrict SNMP traffic to a specific version of SNMP. Earlier versions of SNMP are less secure; therefore, denying certain SNMP versions may be required by your security policy. The system can deny SNMP versions 1, 2, 2c, or 3. You control the versions permitted by creating an SNMP map, as explained below. If you do not need to control the versions, simply enable SNMP inspection without a map.

### Procedure

**Step 1** Choose **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **SNMP**.
**Step 2** Click **Add**, or select a map and click **Edit**. When adding a map, enter a map name.
**Step 3** Select the SNMP versions to disallow.

**Step 4** Click **OK**.

### What to do next

You can now configure an inspection policy to use the map. See Configure Application Layer Protocol Inspection.

# SQL*Net Inspection

SQL*Net inspection is enabled by default. The inspection engine supports SQL*Net versions 1 and 2, but only the Transparent Network Substrate (TNS) format. Inspection does not support the Tabular Data Stream (TDS) format. SQL*Net messages are scanned for embedded addresses and ports, and NAT rewrite is applied when necessary.

The default port assignment for SQL*Net is 1521. This is the value used by Oracle for SQL*Net, but this value does not agree with IANA port assignments for Structured Query Language (SQL). If your application uses a different port, apply the SQL*Net inspection to a traffic class that includes that port.

Disable SQL*Net inspection when:

- SQL data transfer occurs on the same port as the SQL control TCP port 1521. The security appliance acts as a proxy when SQL*Net inspection is enabled and reduces the client window size from 65000 to about 16000 causing data transfer issues.
- The inspection of high rates of SQL traffic causes unacceptable spikes in CPU usage.

After disabling SQL*Net inspection, use the **clear conn port 1521** command so that connections can be rebuilt without inspection.

For information on enabling SQL*Net inspection, see Configure Application Layer Protocol Inspection.

# Sun RPC Inspection

This section describes Sun RPC application inspection.

## Sun RPC Inspection Overview

Sun RPC protocol inspection is enabled by default. You simply need to manage the Sun RPC server table to identify which services are allowed to traverse the firewall. However, pinholing for NFS is done for any server even without the server table configuration.

Sun RPC is used by NFS and NIS. Sun RPC services can run on any port. When a client attempts to access a Sun RPC service on a server, it must learn the port that service is running on. It does this by querying the port mapper process, usually rpcbind, on the well-known port of 111.

The client sends the Sun RPC program number of the service and the port mapper process responds with the port number of the service. The client sends its Sun RPC queries to the server, specifying the port identified by the port mapper process. When the server replies, the ASA intercepts this packet and opens both embryonic TCP and UDP connections on that port.

NAT or PAT of Sun RPC payload information is not supported.

# Manage Sun RPC Services

Use the Sun RPC services table to control Sun RPC traffic based on established Sun RPC sessions.

**Procedure**

**Step 1** Choose **Configuration** > **Firewall** > **Advanced** > **SUNRPC Server**.

**Step 2** Do one of the following:

- Click **Add** to add a new server.
- Select a server and click **Edit**.

**Step 3** Configure the service properties:

- **Interface Name**—The interface through which traffic to the server flows.
- **IP Address/Mask**—The address of the Sun RPC server.
- **Service ID**—The service type on the server. To determine the service type (for example, 100003), use the **sunrpcinfo** command at the UNIX or Linux command line on the Sun RPC server machine.
- **Protocol**—Whether the service uses TCP or UDP.
- **Port/Port Range**—The port or range of ports used by the service.
- **Timeout**—The idle timeout for the pinhole opened for the connection by Sun RPC inspection.

**Step 4** Click **OK**.

**Step 5** (Optional.) Monitor the pinholes created for these services.

To display the pinholes open for Sun RPC services, enter the **show sunrpc-server active** command. Select **Tools** > **Command Line Interface** to enter the command. For example:

```
hostname# show sunrpc-server active
LOCAL FOREIGN SERVICE TIMEOUT
-----------------------------------------------
1 209.165.200.5/0 192.168.100.2/2049 100003 0:30:00
2 209.165.200.5/0 192.168.100.2/2049 100003 0:30:00
3 209.165.200.5/0 192.168.100.2/647 100005 0:30:00
4 209.165.200.5/0 192.168.100.2/650 100005 0:30:00
```

The entry in the LOCAL column shows the IP address of the client or server on the inside interface, while the value in the FOREIGN column shows the IP address of the client or server on the outside interface.

If necessary, you can clear these services using the **clear sunrpc-server active**

# TFTP Inspection

TFTP inspection is enabled by default.

TFTP, described in RFC 1350, is a simple protocol to read and write files between a TFTP server and client.

The inspection engine inspects TFTP read request (RRQ), write request (WRQ), and error notification (ERROR), and dynamically creates connections and translations, if necessary, to permit file transfer between a TFTP client and server.

A dynamic secondary channel and a PAT translation, if necessary, are allocated on a reception of a valid read (RRQ) or write (WRQ) request. This secondary channel is subsequently used by TFTP for file transfer or error notification.

Only the TFTP server can initiate traffic over the secondary channel, and at most one incomplete secondary channel can exist between the TFTP client and server. An error notification from the server closes the secondary channel.

TFTP inspection must be enabled if static PAT is used to redirect TFTP traffic.

For information on enabling TFTP inspection, see Configure Application Layer Protocol Inspection.

# XDMCP Inspection

XDMCP is a protocol that uses UDP port 177 to negotiate X sessions, which use TCP when established.

For successful negotiation and start of an XWindows session, the ASA must allow the TCP back connection from the Xhosted computer. To permit the back connection, you can use access rules to allow the TCP ports. Alternatively, you can use the **established** command on the ASA. Once XDMCP negotiates the port to send the display, the **established** command is consulted to verify if this back connection should be permitted.

During the XWindows session, the manager talks to the display Xserver on the well-known port 6000 | n. Each display has a separate connection to the Xserver, as a result of the following terminal setting.

```
setenv DISPLAY Xserver:n
```

where *n* is the display number.

When XDMCP is used, the display is negotiated using IP addresses, which the ASA can NAT if needed. XDCMP inspection does not support PAT.

For information on enabling XDMCP inspection, see Configure Application Layer Protocol Inspection.

# VXLAN Inspection

Virtual Extensible Local Area Network (VXLAN) inspection works on VXLAN encapsulated traffic that passes through the ASA. It ensures that the VXLAN header format conforms to standards, dropping any malformed packets. VXLAN inspection is not done on traffic for which the ASA acts as a VXLAN Tunnel End Point (VTEP) or a VXLAN gateway, as those checks are done as a normal part of decapsulating VXLAN packets.

VXLAN packets are UDP, normally on port 4789. This port is part of the default-inspection-traffic class, so you can simply add VXLAN inspection to the inspection_default service policy rule. Alternatively, you can create a class for it using port or ACL matching.

# History for Basic Internet Protocol Inspection

| Feature Name | Releases | Feature Information |
|---|---|---|
| DCERPC inspection support for ISystemMapper UUID message RemoteGetClassObject opnum3. | 9.4(1) | The ASA started supporting non-EPM DCERPC messages in release 8.3, supporting the ISystemMapper UUID message RemoteCreateInstance opnum4. This change extends support to the RemoteGetClassObject opnum3 message.<br>We did not modify any ASDM screens. |
| VXLAN packet inspection | 9.4(1) | The ASA can inspect the VXLAN header to enforce compliance with the standard format.<br>We modified the following screen: **Configuration > Firewall > Service Policy Rules > Add Service Policy Rule > Rule Actions > Protocol Inspection**. |
| ESMTP inspection change in default behavior for TLS sessions. | 9.4(1) | The default for ESMTP inspection was changed to allow TLS sessions, which are not inspected. However, this default applies to new or reimaged systems. If you upgrade a system that includes **no allow-tls**, the command is not changed.<br>The change in default behavior was also made in these older versions: 8.4(7.25), 8.5(1.23), 8.6(1.16), 8.7(1.15), 9.0(4.28), 9.1(6.1), 9.2(3.2) 9.3(1.2), 9.3(2.2). |
| IP Options inspection improvements. | 9.5(1) | IP Options inspection now supports all possible IP options. You can tune the inspection to allow, clear, or drop any standard or experimental options, including those not yet defined. You can also set a default behavior for options not explicitly defined in an IP options inspection map.<br>We changed the IP Options Inspect Map dialog box to include additional options. You now select which options to allow and optionally clear. |
| DCERPC inspection improvements and UUID filtering | 9.5(2) | DCERPC inspection now supports NAT for OxidResolver ServerAlive2 opnum5 messages. You can also now filter on DCERPC message universally unique identifiers (UUIDs) to reset or log particular message types. There is a new DCERPC inspection class map for UUID filtering.<br>We added the following screen: **Configuration** > **Firewall** > **Objects** > **Class Maps** > **DCERPC**. We modified the following screen: **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **DCERPC**. |

| Feature Name | Releases | Feature Information |
|---|---|---|
| DNS over TCP inspection. | 9.6(2) | You can now inspect DNS over TCP traffic (TCP/53).<br>We modified the following page: **Configuration** > **Firewall** > **Objects** > **Inspection Maps** > **DNS** Add/Edit dialog box. |
| Cisco Umbrella support. | 9.10(1) | You can configure the device to redirect DNS requests to Cisco Umbrella, so that your Enterprise Security policy defined in Cisco Umbrella can be applied to user connections. You can allow or block connections based on FQDN, or for suspicious FQDNs, you can redirect the user to the Cisco Umbrella intelligent proxy, which can perform URL filtering. The Umbrella configuration is part of the DNS inspection policy.<br>We added or modified the following screens: **Configuration** > **Firewall** > **Objects** > **Umbrella**, **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **DNS**. |
| Cisco Umbrella Enhancements. | 9.12(1) | You can now identify local domain names that should bypass Cisco Umbrella. DNS requests for these domains go directly to the DNS servers without Umbrella processing. You can also identify which Umbrella servers to use for resolving DNS requests. Finally, you can define the Umbrella inspection policy to fail open, so that DNS requests are not blocked if the Umbrella server is unavailable.<br>We modified the following screens: **Configuration** > **Firewall** > **Objects** > **Umbrella**, **Configuration** > **Firewall** > **Objects** > **Inspect Maps** > **DNS**. |
| XDMCP inspection disabled by default in new installations. | 9.15(1) | Previously, XDMCP inspection was enabled by default for all traffic. Now, on new installations, which includes new systems and reimaged systems, XDMCP is off by default. If you need this inspection, please enable it. Note that on upgrades, your current settings for XDMCP inspection are retained, even if you simply had it enabled by way of the default inspection settings. |