Getting Started with Application Layer Protocol Inspection

This chapter describes how to configure application layer protocol inspection. Inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the ASA to do a deep packet inspection instead of passing the packet through the fast path (see the general operations configuration guide for more information about the fast path). As a result, inspection engines can affect overall throughput. Several common inspection engines are enabled on the ASA by default, but you might need to enable others depending on your network.

This chapter includes the following sections:

- Information about Application Layer Protocol Inspection, page 9-1
- Guidelines and Limitations, page 9-3
- Default Settings and NAT Limitations, page 9-4
- Configuring Application Layer Protocol Inspection, page 9-7

Information about Application Layer Protocol Inspection

This section includes the following topics:

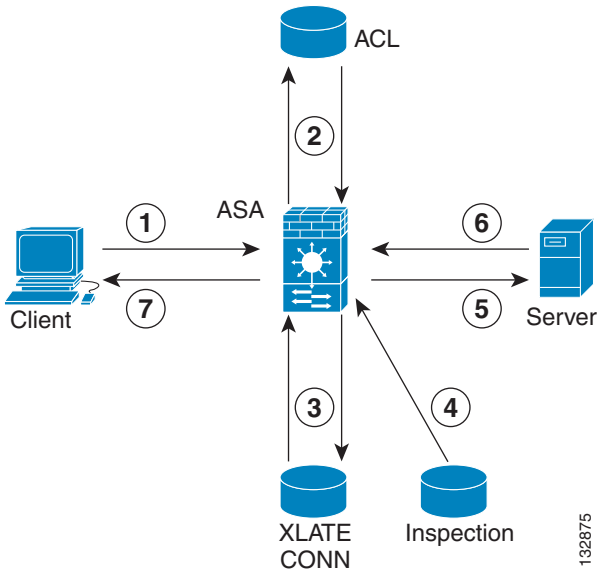
- How Inspection Engines Work, page 9-1
- When to Use Application Protocol Inspection, page 9-2

How Inspection Engines Work

As illustrated in Figure 9-1, the ASA uses three databases for its basic operation:

- ACLs—Used for authentication and authorization of connections based on specific networks, hosts, and services (TCP/UDP port numbers).
- Inspections—Contains a static, predefined set of application-level inspection functions.
- Connections (XLATE and CONN tables)—Maintains state and other information about each established connection. This information is used by the Adaptive Security Algorithm and cut-through proxy to efficiently forward traffic within established sessions.

Figure 9-1 How Inspection Engines Work



In Figure 9-1, operations are numbered in the order they occur, and are described as follows:

1. A TCP SYN packet arrives at the ASA to establish a new connection.
2. The ASA checks the ACL database to determine if the connection is permitted.
3. The ASA creates a new entry in the connection database (XLATE and CONN tables).
4. The ASA checks the Inspections database to determine if the connection requires application-level inspection.
5. After the application inspection engine completes any required operations for the packet, the ASA forwards the packet to the destination system.
6. The destination system responds to the initial request.
7. The ASA receives the reply packet, looks up the connection in the connection database, and forwards the packet because it belongs to an established session.

The default configuration of the ASA includes a set of application inspection entries that associate supported protocols with specific TCP or UDP port numbers and that identify any special handling required.

When to Use Application Protocol Inspection

When a user establishes a connection, the ASA checks the packet against ACLs, creates an address translation, and creates an entry for the session in the fast path, so that further packets can bypass time-consuming checks. However, the fast path relies on predictable port numbers and does not perform address translations inside a packet.

Many protocols open secondary TCP or UDP ports. The initial session on a well-known port is used to negotiate dynamically assigned port numbers.

Other applications embed an IP address in the packet that needs to match the source address that is normally translated when it goes through the ASA.

If you use applications like these, then you need to enable application inspection.

When you enable application inspection for a service that embeds IP addresses, the ASA translates embedded addresses and updates any checksum or other fields that are affected by the translation.

When you enable application inspection for a service that uses dynamically assigned ports, the ASA monitors sessions to identify the dynamic port assignments, and permits data exchange on these ports for the duration of the specific session.

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single and multiple context mode.

Firewall Mode Guidelines

Supported in routed and transparent firewall mode.

Failover Guidelines

State information for multimedia sessions that require inspection are not passed over the state link for stateful failover. The exception is GTP, which is replicated over the state link.

IPv6 Guidelines

Supports IPv6 for the following inspections:

- DNS
- FTP
- HTTP
- ICMP
- SIP
- SMTP
- IPsec pass-through
- IPv6

Supports NAT64 for the following inspections:

- DNS
- FTP
- HTTP
- ICMP

Additional Guidelines and Limitations

Some inspection engines do not support PAT, NAT, outside NAT, or NAT between same security interfaces. See "Default Settings and NAT Limitations" for more information about NAT support.

For all the application inspections, the ASA limits the number of simultaneous, active data connections to 200 connections. For example, if an FTP client opens multiple secondary connections, the FTP inspection engine allows only 200 active connections and the 201 connection is dropped and the adaptive security appliance generates a system error message.

Inspected protocols are subject to advanced TCP-state tracking, and the TCP state of these connections is not automatically replicated. While these connections are replicated to the standby unit, there is a best-effort attempt to re-establish a TCP state.

Default Settings and NAT Limitations

By default, the configuration includes a policy that matches all default application inspection traffic and applies inspection to the traffic on all interfaces (a global policy). Default application inspection traffic includes traffic to the default ports for each protocol. You can only apply one global policy, so if you want to alter the global policy, for example, to apply inspection to non-standard ports, or to add inspections that are not enabled by default, you need to either edit the default policy or disable it and apply a new one.

Table 9-1 lists all inspections supported, the default ports used in the default class map, and the inspection engines that are on by default, shown in bold. This table also notes any NAT limitations.

Table 9-1 Supported Application Inspection Engines

Application[1]	Default Port	NAT Limitations	Standards[2]	Comments
CTIQBE	TCP/2748	No extended PAT. No NAT64. (Clustering) No static PAT.	—	—
DCERPC	TCP/135	No NAT64.	—	—
DNS over UDP	UDP/53	No NAT support is available for name resolution through WINS.	RFC 1123	—
FTP	TCP/21	(Clustering) No static PAT.	RFC 959	—
GTP	UDP/3386 UDP/2123	No extended PAT. No NAT64.	—	Requires a special license.
H.323 H.225 and **RAS**	TCP/1720 UDP/1718 UDP (RAS) 1718-1719	No dynamic NAT or PAT. Static PAT may not work. (Clustering) No static PAT. No extended PAT. No per-session PAT. No NAT on same security interfaces. No outside NAT. No NAT64.	ITU-T H.323, H.245, H225.0, Q.931, Q.932	—
HTTP	TCP/80	—	RFC 2616	Beware of MTU limitations stripping ActiveX and Java. If the MTU is too small to allow the Java or ActiveX tag to be included in one packet, stripping may not occur.
ICMP	—	—	—	—

Table 9-1 Supported Application Inspection Engines (continued)

Application[1]	Default Port	NAT Limitations	Standards[2]	Comments
ICMP ERROR	—	—	—	—
ILS (LDAP)	TCP/389	No extended PAT. No NAT64.	—	—
Instant Messaging (IM)	Varies by client	No extended PAT. No NAT64.	RFC 3860	—
IP Options	—	No NAT64.	RFC 791, RFC 2113	—
IPsec Pass Through	UDP/500	No PAT. No NAT64.	—	—
IPv6	—	No NAT64.	RFC 2460	—
MGCP	UDP/2427, 2727	No extended PAT. No NAT64. (Clustering) No static PAT.	RFC 2705bis-05	—
MMP	TCP 5443	No extended PAT. No NAT64.	—	—
NetBIOS Name Server over IP	UDP/137, 138 (Source ports)	No extended PAT. No NAT64.	—	NetBIOS is supported by performing NAT of the packets for NBNS UDP port 137 and NBDS UDP port 138.
PPTP	TCP/1723	No NAT64. (Clustering) No static PAT.	RFC 2637	—
RADIUS Accounting	1646	No NAT64.	RFC 2865	—
RSH	TCP/514	No PAT. No NAT64. (Clustering) No static PAT.	Berkeley UNIX	—
RTSP	TCP/554	No extended PAT. No outside NAT. No NAT64. (Clustering) No static PAT.	RFC 2326, 2327, 1889	No handling for HTTP cloaking.
ScanSafe (Cloud Web Security)	TCP/80 TCP/413	—	—	These ports are not included in the default-inspection-traffic class for the ScanSafe inspection.

Table 9-1 Supported Application Inspection Engines (continued)

Application[1]	Default Port	NAT Limitations	Standards[2]	Comments
SIP	TCP/5060 UDP/5060	No outside NAT. No NAT on same security interfaces. No extended PAT. No per-session PAT. No NAT64. (Clustering) No static PAT.	RFC 2543	—
SKINNY (SCCP)	TCP/2000	No outside NAT. No NAT on same security interfaces. No extended PAT. No per-session PAT. No NAT64. (Clustering) No static PAT.	—	Does not handle TFTP uploaded Cisco IP Phone configurations under certain circumstances.
SMTP and **ESMTP**	TCP/25	No NAT64.	RFC 821, 1123	—
SNMP	UDP/161, 162	No NAT or PAT.	RFC 1155, 1157, 1212, 1213, 1215	v.2 RFC 1902-1908; v.3 RFC 2570-2580.
SQL*Net	TCP/1521	No extended PAT. No NAT64. (Clustering) No static PAT.	—	v.1 and v.2.
Sun RPC over UDP and TCP	UDP/111	No extended PAT. No NAT64.	—	The default rule includes UDP port 111; if you want to enable Sun RPC inspection for TCP port 111, you need to create a new rule that matches TCP port 111 and performs Sun RPC inspection.
TFTP	UDP/69	No NAT64. (Clustering) No static PAT.	RFC 1350	Payload IP addresses are not translated.
WAAS	—	No extended PAT. No NAT64.	—	—
XDCMP	UDP/177	No extended PAT. No NAT64. (Clustering) No static PAT.	—	—

1. Inspection engines that are enabled by default for the default port are in bold.
2. The ASA is in compliance with these standards, but it does not enforce compliance on packets being inspected. For example, FTP commands are supposed to be in a particular order, but the ASA does not enforce the order.

The default policy configuration includes the following commands:

```
class-map inspection_default
 match default-inspection-traffic
policy-map type inspect dns preset_dns_map
 parameters
    message-length maximum client auto
    message-length maximum 512
    dns-guard
    protocol-enforcement
    nat-rewrite
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225 _default_h323_map
  inspect h323 ras _default_h323_map
  inspect ip-options _default_ip_options_map
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp _default_esmtp_map
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
```

Configuring Application Layer Protocol Inspection

This feature uses Modular Policy Framework to create a service policy. Service policies provide a consistent and flexible way to configure ASA features. For example, you can use a service policy to create a timeout configuration that is specific to a particular TCP application, as opposed to one that applies to all TCP applications. See Chapter 1, "Configuring a Service Policy Using the Modular Policy Framework," for more information. For some applications, you can perform special actions when you enable inspection. See Chapter 1, "Configuring a Service Policy Using the Modular Policy Framework," for more information.

Inspection is enabled by default for some applications. See the "Default Settings and NAT Limitations" section for more information. Use this section to modify your inspection policy.

Detailed Steps

Step 1 To identify the traffic to which you want to apply inspections, add either a Layer 3/4 class map for through traffic or a Layer 3/4 class map for management traffic. See the "Creating a Layer 3/4 Class Map for Through Traffic" section on page 1-12 and "Creating a Layer 3/4 Class Map for Management Traffic" section on page 1-14 for detailed information. The management Layer 3/4 class map can be used only with the RADIUS accounting inspection.

The default Layer 3/4 class map for through traffic is called "inspection_default." It matches traffic using a special **match** command, **match default-inspection-traffic**, to match the default ports for each application protocol. This traffic class (along with **match any**, which is not typically used for inspection) matches both IPv4 and IPv6 traffic for inspections that support IPv6. See the "Guidelines and Limitations" section on page 9-3 for a list of IPv6-enabled inspections.

You can specify a **match access-list** command along with the **match default-inspection-traffic** command to narrow the matched traffic to specific IP addresses. Because the **match default-inspection-traffic** command specifies the ports to match, any ports in the ACL are ignored.

Tip We suggest that you only inspect traffic on ports on which you expect application traffic; if you inspect all traffic, for example using **match any**, the ASA performance can be impacted.

If you want to match non-standard ports, then create a new class map for the non-standard ports. See the "Default Settings and NAT Limitations" section on page 9-4 for the standard ports for each inspection engine. You can combine multiple class maps in the same policy if desired, so you can create one class map to match certain traffic, and another to match different traffic. However, if traffic matches a class map that contains an inspection command, and then matches another class map that also has an inspection command, only the first matching class is used. For example, SNMP matches the inspection_default class. To enable SNMP inspection, enable SNMP inspection for the default class in Step 5. Do not add another class that matches SNMP.

For example, to limit inspection to traffic from 10.1.1.0 to 192.168.1.0 using the default class map, enter the following commands:

```
ciscoasa(config)# access-list inspect extended permit ip 10.1.1.0 255.255.255.0
192.168.1.0 255.255.255.0
ciscoasa(config)# class-map inspection_default
ciscoasa(config-cmap)# match access-list inspect
```

View the entire class map using the following command:

```
ciscoasa(config-cmap)# show running-config class-map inspection_default
!
class-map inspection_default
 match default-inspection-traffic
 match access-list inspect
!
```

To inspect FTP traffic on port 21 as well as 1056 (a non-standard port), create an ACL that specifies the ports, and assign it to a new class map:

```
ciscoasa(config)# access-list ftp_inspect extended permit tcp any any eq 21
ciscoasa(config)# access-list ftp_inspect extended permit tcp any any eq 1056
ciscoasa(config)# class-map new_inspection
ciscoasa(config-cmap)# match access-list ftp_inspect
```

Step 2 (Optional) Some inspection engines let you control additional parameters when you apply the inspection to the traffic. See the following sections to configure an inspection policy map for your application:

- DCERPC—See the "Configuring a DCERPC Inspection Policy Map for Additional Inspection Control" section on page 13-2
- DNS—See the "(Optional) Configuring a DNS Inspection Policy Map and Class Map" section on page 10-3
- ESMTP—See the "Configuring an ESMTP Inspection Policy Map for Additional Inspection Control" section on page 10-33
- FTP—See the "Configuring an FTP Inspection Policy Map for Additional Inspection Control" section on page 10-12.
- GTP—See the "Configuring a GTP Inspection Policy Map for Additional Inspection Control" section on page 13-4.

- H323—See the "Configuring an H.323 Inspection Policy Map for Additional Inspection Control" section on page 11-6
- HTTP—See the "Configuring an HTTP Inspection Policy Map for Additional Inspection Control" section on page 10-16.
- Instant Messaging—See the "Configuring an Instant Messaging Inspection Policy Map for Additional Inspection Control" section on page 10-21
- IP Options—See the "Configuring an IP Options Inspection Policy Map for Additional Inspection Control" section on page 10-25
- IPsec Pass Through—See the "IPsec Pass Through Inspection" section on page 10-25
- IPv6—See the "(Optional) Configuring an IPv6 Inspection Policy Map" section on page 10-27
- MGCP—See the "Configuring an MGCP Inspection Policy Map for Additional Inspection Control" section on page 11-12.
- NetBIOS—See the "Configuring a NetBIOS Inspection Policy Map for Additional Inspection Control" section on page 10-30
- RADIUS Accounting—See the "Configuring a RADIUS Inspection Policy Map for Additional Inspection Control" section on page 13-9
- RTSP—See the "Configuring an RTSP Inspection Policy Map for Additional Inspection Control" section on page 11-16
- ScanSafe (Cloud Web Security)—See the "Configuring a Service Policy to Send Traffic to Cloud Web Security" section on page 25-10
- SIP—See the "Configuring a SIP Inspection Policy Map for Additional Inspection Control" section on page 11-20
- Skinny—See the "Configuring a Skinny (SCCP) Inspection Policy Map for Additional Inspection Control" section on page 11-26
- SNMP—See the "Configuring an SNMP Inspection Policy Map for Additional Inspection Control" section on page 13-10.

Step 3 To add or edit a Layer 3/4 policy map that sets the actions to take with the class map traffic, enter the following command:

```
ciscoasa(config)# policy-map name
ciscoasa(config-pmap)#
```

The default policy map is called "global_policy." This policy map includes the default inspections listed in the "Default Settings and NAT Limitations" section on page 9-4. If you want to modify the default policy (for example, to add or delete an inspection, or to identify an additional class map for your actions), then enter **global_policy** as the name.

Step 4 To identify the class map from Step 1 to which you want to assign an action, enter the following command:

```
ciscoasa(config-pmap)# class class_map_name
ciscoasa(config-pmap-c)#
```

If you are editing the default policy map, it includes the inspection_default class map. You can edit the actions for this class by entering **inspection_default** as the name. To add an additional class map to this policy map, identify a different name. You can combine multiple class maps in the same policy if desired, so you can create one class map to match certain traffic, and another to match different traffic. However, if traffic matches a class map that contains an inspection command, and then matches another class map that also has an inspection command, only the first matching class is used. For example, SNMP matches the inspection_default class map.To enable SNMP inspection, enable SNMP inspection for the default

class in Step 5. Do not add another class that matches SNMP.

Step 5 Enable application inspection by entering the following command:

```
ciscoasa(config-pmap-c)# inspect protocol
```

The *protocol* is one of the following values:

Table 9-2 Protocol Keywords

Keywords	Notes
ctiqbe	—
dcerpc [*map_name*]	If you added a DCERPC inspection policy map according to "Configuring a DCERPC Inspection Policy Map for Additional Inspection Control" section on page 13-2, identify the map name in this command.
dns [*map_name*] [**dynamic-filter-snoop**]	If you added a DNS inspection policy map according to "(Optional) Configuring a DNS Inspection Policy Map and Class Map" section on page 10-3, identify the map name in this command. The default DNS inspection policy map name is "preset_dns_map." The default inspection policy map sets the maximum DNS packet length to 512 bytes. To enable DNS snooping for the Botnet Traffic Filter, enter the **dynamic-filter-snoop** keyword. See the "Enabling DNS Snooping" section on page 26-10 for more information.
esmtp [*map_name*]	If you added an ESMTP inspection policy map according to "Configuring an ESMTP Inspection Policy Map for Additional Inspection Control" section on page 10-33, identify the map name in this command.
ftp [**strict** [*map_name*]]	Use the **strict** keyword to increase the security of protected networks by preventing web browsers from sending embedded commands in FTP requests. See the "Using the strict Option" section on page 10-11 for more information. If you added an FTP inspection policy map according to "Configuring an FTP Inspection Policy Map for Additional Inspection Control" section on page 10-12, identify the map name in this command.
gtp [*map_name*]	If you added a GTP inspection policy map according to the "Configuring a GTP Inspection Policy Map for Additional Inspection Control" section on page 13-4, identify the map name in this command.
h323 h225 [*map_name*]	If you added an H323 inspection policy map according to "Configuring an H.323 Inspection Policy Map for Additional Inspection Control" section on page 11-6, identify the map name in this command.
h323 ras [*map_name*]	If you added an H323 inspection policy map according to "Configuring an H.323 Inspection Policy Map for Additional Inspection Control" section on page 11-6, identify the map name in this command.

Table 9-2 Protocol Keywords

Keywords	Notes
http [*map_name*]	If you added an HTTP inspection policy map according to the "Configuring an HTTP Inspection Policy Map for Additional Inspection Control" section on page 10-16, identify the map name in this command.
icmp	—
icmp error	—
ils	—
im [*map_name*]	If you added an Instant Messaging inspection policy map according to "Configuring an Instant Messaging Inspection Policy Map for Additional Inspection Control" section on page 10-21, identify the map name in this command.
ip-options [*map_name*]	If you added an IP Options inspection policy map according to "Configuring an IP Options Inspection Policy Map for Additional Inspection Control" section on page 10-25, identify the map name in this command.
ipsec-pass-thru [*map_name*]	If you added an IPsec Pass Through inspection policy map according to "IPsec Pass Through Inspection" section on page 10-25, identify the map name in this command.
ipv6 [*map_name*]	If you added an IP Options inspection policy map according to "(Optional) Configuring an IPv6 Inspection Policy Map" section on page 10-27, identify the map name in this command.
mgcp [*map_name*]	If you added an MGCP inspection policy map according to "Configuring an MGCP Inspection Policy Map for Additional Inspection Control" section on page 11-12, identify the map name in this command.
netbios [*map_name*]	If you added a NetBIOS inspection policy map according to "Configuring a NetBIOS Inspection Policy Map for Additional Inspection Control" section on page 10-30, identify the map name in this command.
pptp	—
radius-accounting [*map_name*]	The **radius-accounting** keyword is only available for a management class map. See the "Creating a Layer 3/4 Class Map for Management Traffic" section on page 1-14 for more information about creating a management class map. If you added a RADIUS accounting inspection policy map according to "Configuring a RADIUS Inspection Policy Map for Additional Inspection Control" section on page 13-9, identify the map name in this command.
rsh	—
rtsp [*map_name*]	If you added a RTSP inspection policy map according to "Configuring an RTSP Inspection Policy Map for Additional Inspection Control" section on page 11-16, identify the map name in this command.

Table 9-2 Protocol Keywords

Keywords	Notes
scansafe [*map_name*]	If you added a ScanSafe (Cloud Web Security) inspection policy map according to "Configuring a Service Policy to Send Traffic to Cloud Web Security" section on page 25-10, identify the map name in this command.
sip [*map_name*]	If you added a SIP inspection policy map according to "Configuring a SIP Inspection Policy Map for Additional Inspection Control" section on page 11-20, identify the map name in this command.
skinny [*map_name*]	If you added a Skinny inspection policy map according to "Configuring a Skinny (SCCP) Inspection Policy Map for Additional Inspection Control" section on page 11-26, identify the map name in this command.
snmp [*map_name*]	If you added an SNMP inspection policy map according to "Configuring an SNMP Inspection Policy Map for Additional Inspection Control" section on page 13-10, identify the map name in this command.
sqlnet	—
sunrpc	The default class map includes UDP port 111; if you want to enable Sun RPC inspection for TCP port 111, you need to create a new class map that matches TCP port 111, add the class to the policy, and then apply the **inspect sunrpc** command to that class.
tftp	—
waas	—
xdmcp	—

Step 6 To activate the policy map on one or more interfaces, enter the following command:

```
ciscoasa(config)# service-policy policymap_name {global | interface interface_name}
```

Where **global** applies the policy map to all interfaces, and **interface** applies the policy to one interface. By default, the default policy map, "global_policy," is applied globally. Only one global policy is allowed. You can override the global policy on an interface by applying a service policy to that interface. You can only apply one policy map to each interface.