Inspection of Database and Directory Protocols

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. 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:

ILS Inspection

The 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.

The ASA supports NAT for ILS, which is used to register and locate endpoints in the ILS or SiteServer Directory. PAT cannot be supported because only IP addresses are stored by an LDAP database.

For search responses, when the LDAP server is located outside, NAT should be considered to allow internal peers to communicate locally while registered to external LDAP servers. For such search responses, xlates are searched first, and then DNAT entries to obtain the correct address. If both of these searches fail, then the address is not changed. For sites using NAT 0 (no NAT) and not expecting DNAT interaction, we recommend that the inspection engine be turned off 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.

Because ILS traffic 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 **timeout** command.

ILS/LDAP follows a client/server model with sessions handled over a single TCP connection. Depending on the client's actions, several of these sessions may be created.

During connection negotiation time, a BIND PDU is sent from the client to the server. Once a successful BIND RESPONSE from the server is received, other operational messages may be exchanged (such as ADD, DEL, SEARCH, or MODIFY) to perform operations on the ILS Directory. The ADD REQUEST and SEARCH RESPONSE PDUs may contain IP addresses of NetMeeting peers, used by H.323 (SETUP and CONNECT messages) to establish the NetMeeting sessions. Microsoft NetMeeting v2.X and v3.X provides ILS support.

The ILS inspection performs the following operations:

- Decodes the LDAP REQUEST/RESPONSE PDUs using the BER decode functions
- Parses the LDAP packet
- Extracts IP addresses
- Translates IP addresses as necessary
- Encodes the PDU with translated addresses using BER encode functions
- Copies the newly encoded PDU back to the TCP packet
- Performs incremental TCP checksum and sequence number adjustment

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

Note   Because H.225 call signalling traffic only occurs on the secondary UDP channel, the TCP connection is disconnected after the interval specified by the TCP option in the Configuration > Firewall > Advanced > Global Timeouts pane. By default, this interval is set at 60 minutes.

SQL*Net Inspection

SQL*Net inspection is enabled by default.

The SQL*Net protocol consists of different packet types that the ASA handles to make the data stream appear consistent to the Oracle applications on either side of the ASA.

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).

Note   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 ASA translates all addresses and looks in the packets for all embedded ports to open for SQL*Net Version 1.

For SQL*Net Version 2, all DATA or REDIRECT packets that immediately follow REDIRECT packets with a zero data length will be fixed up.

The packets that need fix-up contain embedded host/port addresses in the following format:

```
(ADDRESS=(PROTOCOL=tcp)(DEV=6)(HOST=a.b.c.d)(PORT=a))
```

SQL*Net Version 2 TNSFrame types (Connect, Accept, Refuse, Resend, and Marker) will not be scanned for addresses to NAT nor will inspection open dynamic connections for any embedded ports in the packet.

SQL*Net Version 2 TNSFrames, Redirect, and Data packets will be scanned for ports to open and addresses to NAT, if preceded by a REDIRECT TNSFrame type with a zero data length for the payload. When the Redirect message with data length zero passes through the ASA, a flag will be set in the connection data structure to expect the Data or Redirect message that follows to be translated and ports to be dynamically opened. If one of the TNS frames in the preceding paragraph arrive after the Redirect message, the flag will be reset.

The SQL*Net inspection engine will recalculate the checksum, change IP, TCP lengths, and readjust Sequence Numbers and Acknowledgment Numbers using the delta of the length of the new and old message.

SQL*Net Version 1 is assumed for all other cases. TNSFrame types (Connect, Accept, Refuse, Resend, Marker, Redirect, and Data) and all packets will be scanned for ports and addresses. Addresses will be translated and port connections will be opened.

Sun RPC Inspection

This section describes Sun RPC application inspection. This section includes the following topics:

Sun RPC Inspection Overview

The Sun RPC inspection engine enables or disables application inspection for the Sun RPC protocol. Sun RPC is used by NFS and NIS. Sun RPC services can run on any port. When a client attempts to access an 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.

The following limitations apply to Sun RPC inspection:

- NAT or PAT of Sun RPC payload information is not supported.
- Sun RPC inspection supports inbound ACLs only. Sun RPC inspection does not support outbound ACLs because the inspection engine uses dynamic ACLs instead of secondary connections. Dynamic ACLs are always added on the ingress direction and not on egress; therefore, this inspection engine does not support outbound ACLs. To view the dynamic ACLs configured for the ASA, use the **show asp table classify domain permit** command. For information about the **show asp table classify domain permit** command, see the CLI configuration guide.

SUNRPC Server

Configuration > Properties > SUNRPC Server

The Configuration > Firewall > Advanced > SUNRPC Server pane shows which SunRPC services can traverse the ASA and their specific timeout, on a per server basis.

Fields

- Interface—Displays the interface on which the SunRPC server resides.
- IP address—Displays the IP address of the SunRPC server.
- Mask—Displays the subnet mask of the IP Address of the SunRPC server.
- Service ID—Displays the SunRPC program number, or service ID, allowed to traverse the ASA.
- Protocol—Displays the SunRPC transport protocol (TCP or UDP).
- Port—Displays the SunRPC protocol port range.
- Timeout—Displays the idle time after which the access for the SunRPC service traffic is closed.

Add/Edit SUNRPC Service

Configuration > Properties > SUNRPC Server > Add/Edit SUNRPC Service

The Configuration > Firewall > Advanced > SUNRPC Server > Add/Edit SUNRPC Service dialog box lets you specify what SunRPC services are allowed to traverse the ASA and their specific timeout, on a per-server basis.

Fields

- Interface Name—Specifies the interface on which the SunRPC server resides.
- Protocol—Specifies the SunRPC transport protocol (TCP or UDP).
- IP address—Specifies the IP address of the SunRPC server.
- Port—Specifies the SunRPC protocol port range.
- Mask—Specifies the subnet mask of the IP Address of the SunRPC server.
- Timeout—Specifies the idle time after which the access for the SunRPC service traffic is closed. Format is HH:MM:SS.
- Service ID—Specifies the SunRPC program number, or service ID, allowed to traverse the ASA.