

# ARP Inspection and the MAC Address Table

This chapter describes how to customize the MAC address table and configure ARP Inspection for bridge groups.


- About ARP Inspection and the MAC Address Table, on page 1
- Default Settings, on page 2
- Guidelines for ARP Inspection and the MAC Address Table, on page 2
- Configure ARP Inspection and Other ARP Parameters, on page 3
- Customize the MAC Address Table for Bridge Groups, on page 5
- Monitoring ARP Inspection and the MAC Address Table, on page 7
- History for ARP Inspection and the MAC Address Table, on page 7


# About ARP Inspection and the MAC Address Table

For interfaces in a bridge group, ARP inspection prevents a “man-in-the-middle” attack. You can also customize other ARP settings. You can customize the MAC address table for bridge groups, including adding a static ARP entry to guard against MAC spoofing.

## ARP Inspection for Bridge Group Traffic

By default, all ARP packets are allowed between bridge group members. You can control the flow of ARP packets by enabling ARP inspection.

ARP inspection prevents malicious users from impersonating other hosts or routers (known as ARP spoofing). ARP spoofing can enable a “man-in-the-middle” attack. For example, a host sends an ARP request to the gateway router; the gateway router responds with the gateway router MAC address. The attacker, however, sends another ARP response to the host with the attacker MAC address instead of the router MAC address. The attacker can now intercept all the host traffic before forwarding it on to the router.

ARP inspection ensures that an attacker cannot send an ARP response with the attacker MAC address, so long as the correct MAC address and the associated IP address are in the static ARP table.

When you enable ARP inspection, the ASA compares the MAC address, IP address, and source interface in all ARP packets to static entries in the ARP table, and takes the following actions:

- If the IP address, MAC address, and source interface match an ARP entry, the packet is passed through.
- If there is a mismatch between the MAC address, the IP address, or the interface, then the ASA drops the packet.

- If the ARP packet does not match any entries in the static ARP table, then you can set the ASA to either forward the packet out all interfaces (flood), or to drop the packet.

**Note** The dedicated Management interface never floods packets even if this parameter is set to flood.

## MAC Address Table

When you use bridge groups, the ASA learns and builds a MAC address table in a similar way as a normal bridge or switch: when a device sends a packet through the bridge group, the ASA adds the MAC address to its table. The table associates the MAC address with the source interface so that the ASA knows to send any packets addressed to the device out the correct interface. Because traffic between bridge group members is subject to the ASA security policy, if the destination MAC address of a packet is not in the table, the ASA does not flood the original packet on all interfaces as a normal bridge does. Instead, it generates the following packets for directly-connected devices or for remote devices:

- Packets for directly-connected devices—The ASA generates an ARP request for the destination IP address, so that it can learn which interface receives the ARP response.
- Packets for remote devices—The ASA generates a ping to the destination IP address so that it can learn which interface receives the ping reply.

The original packet is dropped.

For routed mode, you can optionally enable flooding of non-IP packets on all interfaces.

# Default Settings

- If you enable ARP inspection, the default setting is to flood non-matching packets.
- The default timeout value for dynamic MAC address table entries is 5 minutes.
- By default, each interface automatically learns the MAC addresses of entering traffic, and the ASA adds corresponding entries to the MAC address table.

**Note** ASA generates a reset packet to reset a connection that is denied by a stateful inspection engine. Here, the destination MAC address of the packet is not determined based on the ARP table lookup but instead it is taken directly from the packets (connections) that are being denied.

# Guidelines for ARP Inspection and the MAC Address Table

- ARP inspection is only supported for bridge groups.

- MAC address table configuration is only supported for bridge groups.

# Configure ARP Inspection and Other ARP Parameters

For bridge groups, you can enable ARP inspection. You can also configure other ARP parameters for both bridge groups and for routed mode interfaces.

**Procedure**

**Step 1** Add static ARP entries according to Add a Static ARP Entry and Customize Other ARP Parameters, on page 3. ARP inspection compares ARP packets with static ARP entries in the ARP table, so static ARP entries are required for this feature. You can also configure other ARP parameters.

**Step 2** Enable ARP inspection according to Enable ARP Inspection, on page 4.

# Add a Static ARP Entry and Customize Other ARP Parameters

By default for bridge groups, all ARP packets are allowed between bridge group member interfaces. You can control the flow of ARP packets by enabling ARP inspection. ARP inspection compares ARP packets with *static* ARP entries in the ARP table.

For routed interfaces, you can enter static ARP entries, but normally dynamic entries are sufficient. For routed interfaces, the ARP table is used to deliver packets to directly-connected hosts. Although senders identify a packet destination by an IP address, the actual delivery of the packet on Ethernet relies on the Ethernet MAC address. When a router or host wants to deliver a packet on a directly connected network, it sends an ARP request asking for the MAC address associated with the IP address, and then delivers the packet to the MAC address according to the ARP response. The host or router keeps an ARP table so it does not have to send ARP requests for every packet it needs to deliver. The ARP table is dynamically updated whenever ARP responses are sent on the network, and if an entry is not used for a period of time, it times out. If an entry is incorrect (for example, the MAC address changes for a given IP address), the entry needs to time out before it can be updated with the new information.

For transparent mode, the ASA only uses dynamic ARP entries in the ARP table for traffic to and from the ASA, such as management traffic.

You can also set the ARP timeout and other ARP behavior.

**Procedure**

**Step 1** Add a static ARP entry:

**arp** *interface_name ip_address mac_address* [**alias**]

**Example:**

```
ciscoasa(config)# arp outside 10.1.1.1 0009.7cbe.2100
```

This example allows ARP responses from the router at 10.1.1.1 with the MAC address 0009.7cbe.2100 on the outside interface.

Specify **alias** in routed mode to enable proxy ARP for this mapping. If the ASA receives an ARP request for the specified IP address, then it responds with the ASA MAC address. This keyword is useful if you have devices that do not perform ARP, for example. In transparent firewall mode, this keyword is ignored; the ASA does not perform proxy ARP.

**Step 2** Set the ARP timeout for dynamic ARP entries:

**arp timeout** *seconds*

**Example:**

```
ciscoasa(config)# arp timeout 5000
```

This field sets the amount of time before the ASA rebuilds the ARP table, between 60 to 4294967 seconds. The default is 14400 seconds. Rebuilding the ARP table automatically updates new host information and removes old host information. You might want to reduce the timeout because the host information changes frequently.

**Step 3** Allow non-connected subnets:

**arp permit-nonconnected**

The ASA ARP cache only contains entries from directly-connected subnets by default. You can enable the ARP cache to also include non-directly-connected subnets. We do not recommend enabling this feature unless you know the security risks. This feature could facilitate denial of service (DoS) attack against the ASA; a user on any interface could send out many ARP replies and overload the ASA ARP table with false entries.

You may want to use this feature if you use:

- Secondary subnets.
- Proxy ARP on adjacent routes for traffic forwarding.

**Step 4** Set the ARP rate limit to control the number of ARP packets per second:

**arp rate-limit** *seconds*

**Example:**

```
ciscoasa(config)# arp rate-limit 1000
```

Enter a value between 10 and 32768. The default value depends on your ASA model. You can customize this value to prevent an ARP storm attack.

# Enable ARP Inspection

This section describes how to enable ARP inspection for bridge groups.

### Procedure

Enable ARP inspection:

**arp-inspection** *interface_name* **enable** [**flood** | **no-flood**]

**Example:**

```
ciscoasa(config)# arp-inspection outside enable no-flood
```

The **flood** keyword forwards non-matching ARP packets out all interfaces, and **no-flood** drops non-matching packets.

The default setting is to flood non-matching packets. To restrict ARP through the ASA to only static entries, then set this command to **no-flood**.

# Customize the MAC Address Table for Bridge Groups

This section describes how you can customize the MAC address table for bridge groups.

## Add a Static MAC Address for Bridge Groups

Normally, MAC addresses are added to the MAC address table dynamically as traffic from a particular MAC address enters an interface. You can add static MAC addresses to the MAC address table. One benefit to adding static entries is to guard against MAC spoofing. If a client with the same MAC address as a static entry attempts to send traffic to an interface that does not match the static entry, then the ASA drops the traffic and generates a system message. When you add a static ARP entry (see Add a Static ARP Entry and Customize Other ARP Parameters, on page 3), a static MAC address entry is automatically added to the MAC address table.

To add a static MAC address to the MAC address table, perform the following steps.

### Procedure

Add a static MAC address entry:

**mac-address-table static** *interface_name mac_address*

**Example:**

```
ciscoasa(config)# mac-address-table static inside 0009.7cbe.2100
```

The *interface_name* is the source interface.

# Set the MAC Address Timeout

The default timeout value for dynamic MAC address table entries is 5 minutes, but you can change the timeout. To change the timeout, perform the following steps.

### Procedure

Set the MAC address entry timeout:

**mac-address-table aging-time** *timeout_value*

**Example:**

```
ciscoasa(config)# mac-address-table aging-time 10
```

The *timeout_value* (in minutes) is between 5 and 720 (12 hours). 5 minutes is the default.

# Configure MAC Address Learning

By default, each interface automatically learns the MAC addresses of entering traffic, and the ASA adds corresponding entries to the MAC address table. You can disable MAC address learning if desired, however, unless you statically add MAC addresses to the table, no traffic can pass through the ASA. In routed mode, you can enable flooding of non-IP packets on all interfaces.

To configure MAC address learning, perform the following steps:

### Procedure

**Step 1** Disable MAC address learning:

**mac-learn** *interface_name* **disable**

**Example:**

```
ciscoasa(config)# mac-learn inside disable
```

The **no** form of this command reenables MAC address learning.

The **clear configure mac-learn** command reenables MAC address learning on all interfaces.

**Step 2** (Routed mode only) Enable flooding of non-IP packets.

**mac-learn flood**

**Example:**

```
ciscoasa(config)# mac-learn flood
```

# Monitoring ARP Inspection and the MAC Address Table

- **show arp-inspection**

  Monitors ARP Inspection. Shows the current settings for ARP inspection on all interfaces.

- **show mac-address-table [*interface_name*]**

  Monitors the MAC address table. You can view the entire MAC address table (including static and dynamic entries for both interfaces), or you can view the MAC address table for an interface.

  The following is sample output from the **show mac-address-table** command that shows the entire table:

```
ciscoasa# show mac-address-table
interface        mac address           type         Time Left
-----------------------------------------------------------------------
outside     0009.7cbe.2100       static    -
inside     0010.7cbe.6101       static    -
inside     0009.7cbe.5101       dynamic    10
```

  The following is sample output from the **show mac-address-table** command that shows the table for the inside interface:

```
ciscoasa# show mac-address-table inside
interface        mac address       type         Time Left
-----------------------------------------------------------------------
inside     0010.7cbe.6101       static    -
inside     0009.7cbe.5101       dynamic    10
```

  The following is sample output from the **show mac-address-table** command that shows the total number of static and dynamic bridge group entries:

```
ciscoasa# show mac-address-table count
Static     mac-address bridges (curr/max): 0/16384
Dynamic    mac-address bridges (curr/max): 0/16384
```

# History for ARP Inspection and the MAC Address Table

| Feature Name | Platform Releases | Feature Information |
|---|---|---|
| ARP inspection | 7.0(1) | ARP inspection compares the MAC address, IP address, and source interface in all ARP packets to static entries in the ARP table. This feature is available for Transparent Firewall Mode, and for interfaces in a bridge group in both Transparent and Routed modes starting in 9.7(1).<br>We introduced the following commands: **arp**, **arp-inspection**, and **show arp-inspection**. |

| Feature Name | Platform Releases | Feature Information |
|---|---|---|
| MAC address table | 7.0(1) | You might want to customize the MAC address table for transparent mode, and for interfaces in a bridge group in both Transparent and Routed modes starting in 9.7(1).<br><br>We introduced the following commands: **mac-address-table static**, **mac-address-table aging-time**, **mac-learn disable**, and **show mac-address-table**. |
| ARP cache additions for non-connected subnets | 8.4(5)/9.1(2) | The ASA ARP cache only contains entries from directly-connected subnets by default. You can now enable the ARP cache to also include non-directly-connected subnets. We do not recommend enabling this feature unless you know the security risks. This feature could facilitate denial of service (DoS) attack against the ASA; a user on any interface could send out many ARP replies and overload the ASA ARP table with false entries.<br><br>You may want to use this feature if you use:<br>• Secondary subnets.<br>• Proxy ARP on adjacent routes for traffic forwarding.<br><br>We introduced the following command: **arp permit-nonconnected**. |
| Customizable ARP rate limiting | 9.6(2) | You can set the maximum number of ARP packets allowed per second. The default value depends on your ASA model. You can customize this value to prevent an ARP storm attack.<br><br>We added the following commands: **arp rate-limit, show arp rate-limit** |

| Feature Name | Platform Releases | Feature Information |
|---|---|---|
| Integrated Routing and Bridging | 9.7(1) | Integrated Routing and Bridging provides the ability to route between a bridge group and a routed interface. A bridge group is a group of interfaces that the ASA bridges instead of routes. The ASA is not a true bridge in that the ASA continues to act as a firewall: access control between interfaces is controlled, and all of the usual firewall checks are in place. Previously, you could only configure bridge groups in transparent firewall mode, where you cannot route between bridge groups. This feature lets you configure bridge groups in routed firewall mode, and to route between bridge groups and between a bridge group and a routed interface. The bridge group participates in routing by using a Bridge Virtual Interface (BVI) to act as a gateway for the bridge group. Integrated Routing and Bridging provides an alternative to using an external Layer 2 switch if you have extra interfaces on the ASA to assign to the bridge group. In routed mode, the BVI can be a named interface and can participate separately from member interfaces in some features, such as access rules and DHCP server.<br><br>The following features that are supported in transparent mode are not supported in routed mode: multiple context mode, ASA clustering. The following features are also not supported on BVIs: dynamic routing and multicast routing.<br><br>We modified the following commands: **access-group, access-list ethertype, arp-inspection, dhcpd, mac-address-table static, mac-address-table aging-time, mac-learn, route, show arp-inspection, show bridge-group, show mac-address-table, show mac-learn** |