

Advanced Interface Configuration

This chapter describes how to configure MAC addresses for interfaces, how to set the maximum transmission unit (MTU), and set the TCP maximum segment size (TCP MSS), and how to allow same security level communication. Setting the correct MTU and maximum TCP segment size is essential for the best network performance.

- About advanced interface configuration, on page 1
- Manually Configure the MAC Address, on page 6
- Automatically Assign MAC Addresses, on page 7
- Configure the MTU and TCP MSS, on page 8
- Allow Same Security Level Communication, on page 9
- History for Advanced Interface Configuration, on page 10

About advanced interface configuration

This section describes advanced interface settings.

MAC addresses

Media Access Control (MAC) address is a unique physical address of a network device.

You can manually assign MAC addresses to override the default. For multiple context mode, you can automatically generate unique MAC addresses (for all interfaces assigned to a context) and single context mode (for subinterfaces).. For container instances, the FXOS chassis automatically generates unique MAC addresses for all interfaces.

Note You might want to assign unique MAC addresses to subinterfaces defined on the ASA, because they use the same burned-in MAC address of the parent interface. For example, your service provider might perform access control based on the MAC address. Also, because IPv6 link-local addresses are generated based on the MAC address, assigning unique MAC addresses to subinterfaces allows for unique IPv6 link-local addresses, which can avoid traffic disruption in certain instances on the ASA device.

Note For container instances, even if you are not sharing a subinterface, if you manually configure MAC addresses, make sure you use unique MAC addresses for all subinterfaces on the same parent interface to ensure proper classification.

Default MAC addresses

This reference provides information about default MAC addresses used in network devices and their configurations.

Default MAC address assignments depend on the type of interface.

- Physical interfaces—The physical interface uses the burned-in MAC address.
- VLAN interfaces (Firepower 1010 and Secure Firewall 1210/1220)—Routed firewall mode: All VLAN interfaces share a MAC address. Ensure that any connected switches can support this scenario. If the connected switches require unique MAC addresses, you can manually assign MAC addresses. See Manually Configure the MAC Address, on page 6.

  Transparent firewall mode: Each VLAN interface has a unique MAC address. You can override the generated MAC addresses if desired by manually assigning MAC addresses. See Manually Configure the MAC Address, on page 6.
- EtherChannels—For an EtherChannel, all interfaces that are part of the channel group share the same MAC address. This feature makes the EtherChannel transparent to network applications and users, because they only see the one logical connection; they have no knowledge of the individual links. The port-channel interface uses a unique MAC address from a pool; interface membership does not affect the MAC address.
- EtherChannels (ASA Models)—The port-channel interface uses the lowest-numbered channel group interface MAC address as the port-channel MAC address. Alternatively you can configure a MAC address for the port-channel interface. We recommend configuring a unique MAC address in case the group channel interface membership changes. If you remove the interface that was providing the port-channel MAC address, then the port-channel MAC address changes to the next lowest numbered interface, thus causing traffic disruption.
- Subinterfaces—All subinterfaces of a physical interface use the same burned-in MAC address. You might want to assign unique MAC addresses to subinterfaces. For example, your service provider might perform access control based on the MAC address. Also, because IPv6 link-local addresses are generated based on the MAC address, assigning unique MAC addresses to subinterfaces allows for unique IPv6 link-local addresses, which can avoid traffic disruption in certain instances on the ASA.

Automatic MAC Addresses

In multiple context mode, auto-generation assigns unique MAC addresses to all interfaces assigned to a context.

If you manually assign a MAC address and also enable auto-generation, then the manually assigned MAC address is used. If you later remove the manual MAC address, the auto-generated address is used, if enabled.

In the rare circumstance that the generated MAC address conflicts with another private MAC address in your network, you can manually set the MAC address for the interface.

Because auto-generated addresses (when using a prefix) start with A2, you cannot start manual MAC addresses with A2 if you also want to use auto-generation.

The ASA generates the MAC address using the following format:

A2*xx.yyzz.zzzz*

Where *xx.yy* is a user-defined prefix or an autogenerated prefix based on the last two bytes of the interface MAC address, and *zz.zzzz* is an internal counter generated by the ASA. For the standby MAC address, the address is identical except that the internal counter is increased by 1.

For an example of how the prefix is used, if you set a prefix of 77, then the ASA converts 77 into the hexadecimal value 004D (*yyxx*). When used in the MAC address, the prefix is reversed (*xxyy*) to match the ASA native form:

A2**4D.00***zz.zzzz*

For a prefix of 1009 (03F1), the MAC address is:

A2**F1.03***zz.zzzz*



Note The MAC address format without a prefix is a legacy version. See the **mac-address auto** command in the command reference for more information about the legacy format.

MTU

Maximum Transmission Unit (MTU) specifies the maximum frame *payload* size that the ASA can transmit on a given Ethernet interface. The MTU value *excludes* Ethernet headers, VLAN tags, or other overhead. For example, if you set the MTU to 1500, the expected frame size is 1518 bytes including the headers, or 1522 when VLAN tagging is used. Do not set the MTU value higher to accommodate these headers.

VXLAN and geneve encapsulation requirements

For VXLAN or Geneve, the entire Ethernet datagram is encapsulated, so the new IP packet is larger and requires a larger MTU. Set the ASA VTEP source interface MTU to the network MTU plus 54 bytes for VXLAN or plus 306 bytes for Geneve.

Path MTU discovery

Path MTU Discovery enables all devices in a network path between two hosts to coordinate the MTU so they can standardize on the lowest MTU in the path.

Path MTU discovery support

The ASA supports Path MTU Discovery (as defined in RFC 1191).

Default MTU

The default MTU on the ASA is 1500 bytes. This value does not include the 18-22 bytes for the Ethernet header, VLAN tagging, or other overhead.

When you enable VXLAN on the VTEP source interface, if the MTU is less than 1554 bytes, then the ASA automatically raises the MTU to 1554 bytes. In this case, the entire Ethernet datagram is being encapsulated,

so the new packet is larger and requires a larger MTU. In general, you should set the ASA source interface MTU to the network MTU plus 54 bytes.

MTU and fragmentation

IPv4 packets that are larger than the specified MTU are fragmented into multiple frames.

Packet handling details

For IPv4, if an outgoing IP packet is larger than the specified MTU, it is fragmented into 2 or more frames. Fragments are reassembled at the destination (and sometimes at intermediate hops), and fragmentation can cause performance degradation. For IPv6, packets are typically not allowed to be fragmented at all. Therefore, your IP packets should fit within the MTU size to avoid fragmentation.

For TCP packets, the endpoints typically use their MTU to determine the TCP maximum segment size (MTU - 40, for example). If additional TCP headers are added along the way, for example for site-to-site VPN tunnels, then the TCP MSS might need to be adjusted down by the tunneling entity. See TCP MSS, on page 5.

For UDP or ICMP, the application should take the MTU into account to avoid fragmentation.

Note The ASA can receive frames larger than the configured MTU as long as there is room in memory.

Full reassembly for inspected fragmented packets

The ASA performs virtual reassembly by default. However, when features such as DNS or voice media inspection are active, the firewall performs full reassembly before forwarding. This ensures packets are processed correctly by inspection engines.

For example, SIP packets subject to inspection require full reassembly. If SIP inspection is disabled, the system maintains virtual reassembly. To avoid issues, we recommend reducing SIP message size or using TCP or TLS for large messages rather than sending large SIP INVITE packets over UDP.

The firewall merges fragments in memory. If the egress interface MTU is smaller than the reassembled packet, the device will refragment the data to fit. To verify behavior, use packet captures on the egress interface and adjust MTU settings as needed.

MTU and jumbo frames configuration

A larger MTU lets you send larger packets. Larger packets might be more efficient for your network.

- Matching MTUs on the traffic path—We recommend that you set the MTU on all ASA interfaces and other device interfaces along the traffic path to be the same. Matching MTUs prevents intermediate devices from fragmenting the packets.
- Accommodating jumbo frames—You can set the MTU 9000 bytes or higher when you enable jumbo frames. The maximum depends on the model.

TCP MSS

The TCP maximum segment size (MSS) is the size of the TCP payload *before* any TCP and IP headers are added. TCP MSS values are exchanged between client and server during the three-way handshake when establishing connections. UDP packets are not affected.

TCP MSS configuration and behavior

You can set the TCP MSS on the ASA for through traffic. By default, the maximum TCP MSS is set to 1380 bytes. This setting is useful if the ASA must increase the packet size for IPsec VPN encapsulation. However, for non-IPsec endpoints, you should disable the maximum TCP MSS on the ASA.

If you set a maximum TCP MSS and either endpoint of a connection requests a TCP MSS larger than this value, the ASA overwrites the TCP MSS in the request packet with the ASA maximum. If the host or server does not request a TCP MSS, the ASA assumes the RFC 793-default value of 536 bytes for IPv4 or 1220 bytes for IPv6, but does not modify the packet.

You can also configure the minimum TCP MSS. If a host or server requests a very small TCP MSS, the ASA can increase the value. By default, the minimum TCP MSS is not enabled.

For to-the-box traffic, including for SSL VPN connections, this setting does not apply. The ASA derives the TCP MSS from MTU: MTU minus 40 for IPv4 or MTU minus 60 for IPv6.

TCP MSS adjustment scenario

For example, you leave the default MTU as 1500 bytes. A host requests an MSS of 1500 minus the TCP and IP header length, which sets the MSS to 1460. If the ASA maximum TCP MSS is 1380 (the default), then the ASA changes the MSS value in the TCP request packet to 1380. The server then sends packets with 1380-byte payloads. The ASA can add up to 120 bytes of headers to the packet, and the resulting packet fits within the MTU size of 1500 bytes.

Default TCP MSS

A default TCP MSS is a network configuration setting that sets the maximum TCP MSS on the ASA to 1380 bytes by default, accommodates IPv4 IPsec VPN connections where the headers can equal up to 120 bytes, and fits within the default MTU of 1500 bytes.

Suggested maximum TCP MSS setting

The default TCP MSS assumes the ASA acts as an IPv4 IPsec VPN endpoint and has an MTU of 1500. When the ASA acts as an IPv4 IPsec VPN endpoint, it needs to accommodate up to 120 bytes for TCP and IP headers.

If you change the MTU value, use IPv6, or do not use the ASA as an IPsec VPN endpoint, then you should change the TCP MSS setting.

- Normal traffic—Disable the TCP MSS limit and accept the value established between connection endpoints. Because connection endpoints typically derive the TCP MSS from the MTU, non-IPsec packets usually fit this TCP MSS.
- IPv4 IPsec endpoint traffic—Set the maximum TCP MSS to the MTU - 120. For example, if you use jumbo frames and set the MTU to 9000, then you need to set the TCP MSS to 8880 to take advantage of the new MTU.
- IPv6 IPsec endpoint traffic—Set the maximum TCP MSS to the MTU - 140.

Inter-Interface Communication

Allowing interfaces on the same security level to communicate with each other provides the following benefits:

- You can configure more than 101 communicating interfaces.

  If you use different levels for each interface and do not assign any interfaces to the same security level, you can configure only one interface per level (0 to 100).

- You want traffic to flow freely between all same security interfaces without ACLs.

If you enable same security interface communication, you can still configure interfaces at different security levels as usual.

Intra-Interface Communication (Routed Firewall Mode)

Intra-interface communication might be useful for VPN traffic that enters an interface, but is then routed out the same interface. The VPN traffic might be unencrypted in this case, or it might be reencrypted for another VPN connection. For example, if you have a hub and spoke VPN network, where the ASA is the hub, and remote VPN networks are spokes, for one spoke to communicate with another spoke, traffic must go into the ASA and then out again to the other spoke.



Note All traffic allowed by this feature is still subject to firewall rules. Be careful not to create an asymmetric routing situation that can cause return traffic not to traverse the ASA.

Manually Configure the MAC Address

If you need to manually assign the MAC address, you can do so using this procedure.

You might want to assign unique MAC addresses to subinterfaces defined on the ASA, because they use the same burned-in MAC address of the parent interface. For example, your service provider might perform access control based on the MAC address. Also, because IPv6 link-local addresses are generated based on the MAC address, assigning unique MAC addresses to subinterfaces allows for unique IPv6 link-local addresses, which can avoid traffic disruption in certain instances on the ASA.

Before you begin

In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter the **changeto context** *name* command.

Procedure

Step 1 Enter interface configuration mode:

interface *id*

Example:

```
ciscoasa(config)# interface gigabithethernet 0/0
```

Step 2 Assign a private MAC address to this interface:

mac-address *mac_address* [**standby** *mac_address*]

Example:

```
ciscoasa(config-if)# mac-address 000C.F142.4CDE
```

The *mac_address* is in H.H.H format, where H is a 16-bit hexadecimal digit. For example, the MAC address 00-0C-F1-42-4C-DE is entered as 000C.F142.4CDE. The MAC address must not have the multicast bit set, that is, the second hexadecimal digit from the left cannot be an odd number.

The first two bytes of a manual MAC address cannot be A2 if you also want to use auto-generated MAC addresses.

For use with failover, set the **standby** MAC address. If the active unit fails over and the standby unit becomes active, the new active unit starts using the active MAC addresses to minimize network disruption, while the old active unit uses the standby address.

Automatically Assign MAC Addresses

This section describes how to configure auto-generation of MAC addresses. For multiple context mode, this feature assigns unique MAC addresses to all interface types that are assigned to a context. For single mode, this feature assigns unique MAC addresses to VLAN subinterfaces.

Before you begin

- When you configure a **nameif** command for the interface, the new MAC address is generated immediately. If you enable this feature after you configure interfaces, then MAC addresses are generated for all interfaces immediately after you enable it. If you disable this feature, the MAC address for each interface reverts to the default MAC address. For example, subinterfaces of GigabitEthernet 0/1 revert to using the MAC address of GigabitEthernet 0/1.
- In the rare circumstance that the generated MAC address conflicts with another private MAC address in your network, you can manually set the MAC address for the interface.
- For multiple context mode, complete this procedure in the system execution space. To change from the context to the system execution space, enter the **changeto system** command.

Procedure

Automatically assign private MAC addresses to each interface:

mac-address auto [**prefix** *prefix*]

If you do not enter a prefix, then the ASA autogenerates the prefix based on the last two bytes of the interface MAC address.

If you manually enter a prefix, then the *prefix* is a decimal value between 0 and 65535. This prefix is converted to a four-digit hexadecimal number, and used as part of the MAC address.

Example:

```
ciscoasa(config)# mac-address auto prefix 19
```

Configure the MTU and TCP MSS

Before you begin

- In multiple context mode, complete this procedure in the context execution space. To change from the system to a context configuration, enter the **changeto context** *name* command.
- To increase the MTU above 1500, enable jumbo frames according to Enable Jumbo Frame Support (ASA Virtual, ISA 3000).

Procedure

Step 1 Set the MTU. The minimum and maximum depend on your platform.

mtu *interface_name bytes*

Example:

```
ciscoasa(config-if)# mtu inside ?

configure mode commands/options:
  <64-9198>  MTU bytes
ciscoasa(config)# mtu inside 9000
```

The default is 1500 bytes.

Note When you set the MTU for a port-channel interface, the ASA applies the setting to all member interfaces.

For some models that support jumbo frames, if you enter a value for any interface that is greater than 1500, then you need to enable jumbo frame support. See Enable Jumbo Frame Support (ASA Virtual, ISA 3000).

Step 2 Configure the maximum TCP segment size in bytes, between 48 and any maximum number:

sysopt connection tcpmss [**minimum**] *bytes*

Example:

```
ciscoasa(config)# sysopt connection tcpmss 8500
ciscoasa(config)# sysopt connection tcpmss minimum 1290
```

The default value is 1380 bytes. You can disable this feature by setting bytes to **0**.

For the **minimum** keyword, sets the maximum segment size to be no less than *bytes*, between 48 and 65535. The minimum feature is disabled by default (set to 0).

Step 3 Configure the maximum number of unprocessed TCP segments.

sysopt connection tcp-max-unprocessed-seg *unprocessed segments*

Example:

```
ciscoasa(config)# sysopt connection tcp-max-unprocessed-seg 7
```

The default value is 6. The range is from 6 to 24.

Examples

The following example enables jumbo frames, increases the MTU on all interfaces, and disables the TCP MSS for non-VPN traffic (by setting the TCP MSS to 0, which means there is no limit):

```
mtu inside 9198
mtu outside 9198
sysopt connection tcpmss 0
```

The following example enables jumbo frames, increases the MTU on all interfaces, and changes the TCP MSS for VPN traffic to 9078 (the MTU minus 120):

```
mtu inside 9198
mtu outside 9198
sysopt connection tcpmss 9078
```

Allow Same Security Level Communication

By default, interfaces on the same security level cannot communicate with each other, and packets cannot enter and exit the same interface. This section describes how to enable inter-interface communication when interfaces are on the same security level, and how to enable intra-interface communication.

Procedure

Step 1 Enable interfaces on the same security level so that they can communicate with each other:

same-security-traffic permit inter-interface

Step 2 Enable communication between hosts connected to the same interface:

same-security-traffic permit intra-interface

History for Advanced Interface Configuration

Table 1: History for Advanced Interface Configuration

Feature Name	Releases	Feature Information
Maximum MTU is now 9198 bytes	9.1(6), 9.2(1)	The maximum MTU that the ASA can use is 9198 bytes (check for your model's exact limit at the CLI help). This value does not include the Layer 2 header. Formerly, the ASA let you specify the maximum MTU as 65535 bytes, which was inaccurate and could cause problems. If your MTU was set to a value higher than 9198, then the MTU is automatically lowered when you upgrade. In some cases, this MTU change can cause an MTU mismatch; be sure to set any connecting equipment to use the new MTU value. We modified the following command: **mtu**
Increased MTU size for the ASA on the Firepower 4100/9300 chassis	9.6(2)	You can set the maximum MTU to 9184 bytes on the Firepower 4100 and 9300; formerly, the maximum was 9000 bytes. This MTU is supported with FXOS 2.0.1.68 and later. We modified the following command: **mtu**
Unique MAC address generation for single context mode	9.8(3), 9.8(4), 9.9(2)	You can now enable unique MAC address generation for VLAN subinterfaces in single context mode. Normally, subinterfaces share the same MAC address with the main interface. Because IPv6 link-local addresses are generated based on the MAC address, this feature allows for unique IPv6 link-local addresses. New or modified command: **mac-address auto**
Secure Firewall 3100 auto-negotiation can be enabled or disabled for 1Gigabit and higher interfaces.	9.17(1)	Secure Firewall 3100 auto-negotiation can be enabled or disabled for 1Gigabit and higher interfaces. For other model SFP ports, the **no speed nonegotiate** option sets the speed to 1000 Mbps; the new command means you can set auto-negotiation and speed independently. New/Modified commands: **negotiate-auto**