Configuring IPv6

This chapter describes how to enable and configure IPv6 on the security appliance. IPv6 is available in Routed firewall mode only.

This chapter includes the following sections:

IPv6-enabled Commands

The following security appliance commands can accept and display IPv6 addresses:

- capture
- configure
- copy
- http
- name
- object-group
- ping
- show conn
- show local-host
- show tcpstat
- ssh

- telnet
- tftp-server
- who
- write

Note Failover does not support IPv6. The **ipv6 address** command does not support setting standby addresses for failover configurations. The **failover interface ip** command does not support using IPv6 addresses on the failover and Stateful Failover interfaces.

When entering IPv6 addresses in commands that support them, simply enter the IPv6 address using standard IPv6 notation, for example **`ping fe80::2e0:b6ff:fe01:3b7a`**. The security appliance correctly recognizes and processes the IPv6 address. However, you must enclose the IPv6 address in square brackets ([ ]) in the following situations:

- You need to specify a port number with the address, for example **`[fe80::2e0:b6ff:fe01:3b7a]:8080`**.
- The command uses a colon as a separator, such as the **write net** and **config net** commands. For example, **`configure net [fe80::2e0:b6ff:fe01:3b7a]:/tftp/config/pixconfig`**.

The following commands were modified to work for IPv6:

- debug
- fragment
- ip verify
- mtu
- icmp (entered as **ipv6 icmp**)

The following inspection engines support IPv6:

- FTP
- HTTP
- ICMP
- SMTP
- TCP
- UDP

Configuring IPv6 on an Interface

At a minimum, each interface needs to be configured with an IPv6 link-local address. Additionally, you can add a site-local and global address to the interface.

Note The security appliance does not support IPv6 anycast addresses.

You can configure both IPv6 and IPv4 addresses on an interface.

To configure IPv6 on an interface, perform the following steps:

Step 1 Enter interface configuration mode for the interface for which you are configuring the IPv6 addresses:

```
hostname(config)# interface if
```

Step 2 Configure an IPv6 address for the interface. You can assign several IPv6 addresses to an interface, such as an IPv6 link-local, site-local, and global address. However, at a minimum, you must configure a link-local address.

There are several methods for configuring IPv6 addresses for an interface. Pick the method that suits your needs from the following:

- The simplest method is to enable stateless autoconfiguration on the interface. Enabling stateless autoconfiguration on the interface configures IPv6 addresses based on prefixes received in Router Advertisement messages. A link-local address, based on the Modified EUI-64 interface ID, is automatically generated for the interface when stateless autoconfiguration is enabled. To enable stateless autoconfiguration, enter the following command:

  ```
  hostname(config-if)# ipv6 address autoconfig
  ```

- If you only need to configure a link-local address on the interface and are not going to assign any other IPv6 addresses to the interface, you have the option of manually defining the link-local address or generating one based on the interface MAC address (Modified EUI-64 format).

  Enter the following command to manually specify the link-local address:

  ```
  hostname(config-if)# ipv6 address ipv6-address link-local
  ```

Enter the following command to enable IPv6 on the interface and automatically generate the link-local address using the Modified EUI-64 interface ID based on the interface MAC address:

```
hostname(config-if)# ipv6 enable
```

Note You do not need to use the **ipv6 enable** command if you enter any other **ipv6 address** commands on an interface; IPv6 support is automatically enabled as soon as you assign an IPv6 address to the interface.

- Assign a site-local or global address to the interface. When you assign a site-local or global address, a link-local address is automatically created. Enter the following command to add a global or site-local address to the interface. Use the optional **eui-64** keyword to use the Modified EUI-64 interface ID in the low order 64 bits of the address.

  ```
  hostname(config-if)# ipv6 address ipv6-address [eui-64]
  ```

Step 3 (Optional) Suppress Router Advertisement messages on an interface. By default, Router Advertisement messages are automatically sent in response to router solicitation messages. You may want to disable these messages on any interface for which you do not want the security appliance to supply the IPv6 prefix (for example, the outside interface).

Enter the following command to suppress Router Advertisement messages on an interface:

```
hostname(config-if)# ipv6 nd suppress-ra
```

See the "IPv6 Configuration Example" section on page 9-7 for an example IPv6 addresses applied to an interface.

Configuring IPv6 Default and Static Routes

IPv6 unicast routing is always enabled. The security appliance routes IPv6 traffic between interfaces as long as the interfaces are enabled for IPv6 and the IPv6 ACLs allow the traffic. You can add a default route and static routes using the **ipv6 route** command.

To configure an IPv6 default route and static routes, perform the following steps:

Step 1 To add the default route, use the following command:

```
hostname(config)# ipv6 route interface_name ::/0 next_hop_ipv6_addr
```

The address ::/0 is the IPv6 equivalent of "any."

Step 2 (Optional) Define IPv6 static routes. Use the following command to add an IPv6 static route to the IPv6 routing table:

```
hostname(config)# ipv6 route if_name destination next_hop_ipv6_addr [admin_distance]
```

Note The **ipv6 route** command works like the **route** command used to define IPv4 static routes.

See the "IPv6 Configuration Example" section on page 9-7 for an example of the **ipv6 route** command used to configure the default route.

Configuring IPv6 Access Lists

Configuring an IPv6 access list is similar configuring an IPv4 access, but with IPv6 addresses.

To configure an IPv6 access list, perform the following steps:

Step 1 Create an access entry. To create an access list, use the **ipv6 access-list** command to create entries for the access list. There are two main forms of this command to choose from, one for creating access list entries specifically for ICMP traffic, and one to create access list entries for all other types of IP traffic.

- To create an IPv6 access list entry specifically for ICMP traffic, enter the following command:

  ```
  hostname(config)# ipv6 access-list id [line num] {permit | deny} icmp source
  destination [icmp_type]
  ```

- To create an IPv6 access list entry, enter the following command:

  ```
  hostname(config)# ipv6 access-list id [line num] {permit | deny} protocol source
  [src_port] destination [dst_port]
  ```

The following describes the arguments for the **ipv6 access-list** command:

- *id*—The name of the access list. Use the same *id* in each command when you are entering multiple entries for an access list.
- **line** *num*—When adding an entry to an access list, you can specify the line number in the list where the entry should appear.
- **permit** | **deny**—Determines whether the specified traffic is blocked or allowed to pass.
- **icmp**—Indicates that the access list entry applies to ICMP traffic.

- *protocol*—Specifies the traffic being controlled by the access list entry. This can be the name (**ip**, **tcp**, or **udp**) or number (1-254) of an IP protocol. Alternatively, you can specify a protocol object group using **object-group** *grp_id*.
- *source* and *destination*—Specifies the source or destination of the traffic. The source or destination can be an IPv6 prefix, in the format *prefix/length*, to indicate a range of addresses, the keyword **any**, to specify any address, or a specific host designated by **host** `host_ipv6_addr`.
- *src_port* and *dst_port*—The source and destination port (or service) argument. Enter an operator (**lt** for less than, **gt** for greater than, **eq** for equal to, **neq** for not equal to, or **range** for an inclusive range) followed by a space and a port number (or two port numbers separated by a space for the **range** keyword).
- *icmp_type*—Specifies the ICMP message type being filtered by the access rule. The value can be a valid ICMP type number (from 0 to 155) or one of the ICMP type literals as shown in Appendix D, "Addresses, Protocols, and Ports". Alternatively, you can specify an ICMP object group using **object-group** *id*.

Step 2 To apply the access list to an interface, enter the following command:

```
hostname(config)# access-group access_list_name {in | out} interface if_name
```

See the "IPv6 Configuration Example" section on page 9-7 for an example IPv6 access list.

Verifying the IPv6 Configuration

This section describes how to verify your IPv6 configuration. You can use various show commands to verify your IPv6 settings.

This section includes the following topics:

The show ipv6 interface Command

To display the IPv6 interface settings, enter the following command:

```
hostname# show ipv6 interface [if_name]
```

Including the interface name, such as "outside", displays the settings for the specified interface. Excluding the name from the command displays the setting for all interfaces that have IPv6 enabled on them. The output for the command shows the following:

- The name and status of the interface.
- The link-local and global unicast addresses.
- The multicast groups the interface belongs to.
- ICMP redirect and error message settings.
- Neighbor discovery settings.

The following is sample output from the **show ipv6 interface** command:

```
hostname# show ipv6 interface
```

```
ipv6interface is down, line protocol is down
  IPv6 is enabled, link-local address is fe80::20d:88ff:feee:6a82 [TENTATIVE]
  No global unicast address is configured
  Joined group address(es):
    ff02::1
    ff02::1:ffee:6a82
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
```

Note The **show interface** command only displays the IPv4 settings for an interface. To see the IPv6 configuration on an interface, you need to use the **show ipv6 interface** command. The **show ipv6 interface** command does not display any IPv4 settings for the interface (if both are configured on the interface).

The show ipv6 route Command

To display the routes in the IPv6 routing table, enter the following command:

```
hostname# show ipv6 route
```

The output from the **show ipv6 route** command is similar to the IPv4 **show route** command. It displays the following information:

- The protocol that derived the route.
- The IPv6 prefix of the remote network.
- The administrative distance and metric for the route.
- The address of the next-hop router.
- The interface through which the next hop router to the specified network is reached.

The following is sample output from the **show ipv6 route** command:

```
hostname# show ipv6 route

IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
L   fe80::/10 [0/0]
     via ::, inside
L   fec0::a:0:0:a0a:a70/128 [0/0]
     via ::, inside
C   fec0:0:0:a::/64 [0/0]
     via ::, inside
L   ff00::/8 [0/0]
     via ::, inside
```

Configuring a Dual IP Stack on an Interface

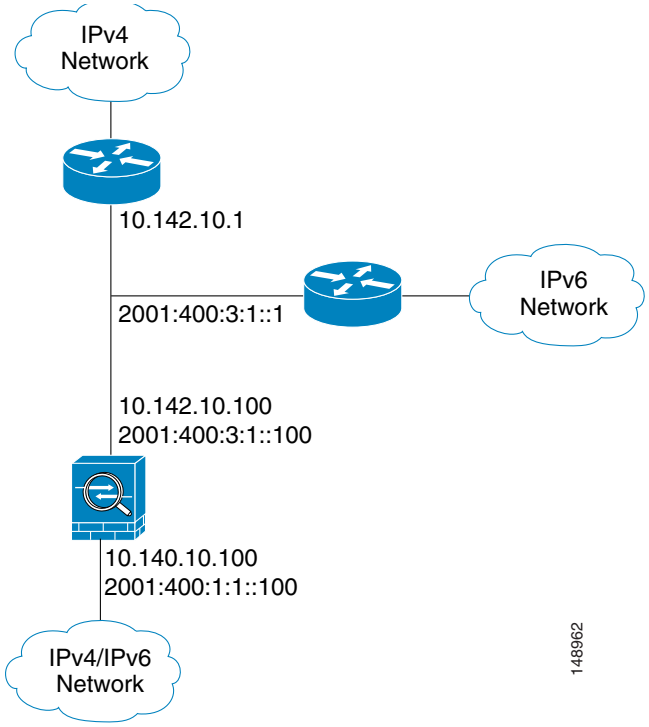
The security appliance supports the configuration of both IPv6 and IPv4 on an interface. You do not need to enter any special commands to do so; simply enter the IPv4 configuration commands and IPv6 configuration commands as you normally would. Make sure you configure the default route for both IPv4 and IPv6.

IPv6 Configuration Example

Example 9-1 shows several features of IPv6 configuration:

- Each interface is configured with both IPv6 and IPv4 addresses.
- The IPv6 default route is set with the **ipv6 route** command.
- An IPv6 access list is applied to the outside interface.

Figure 9-1 IPv6 Dual Stack Configuration



Example 9-1 IPv6 Configuration Example

```
interface Ethernet0
 speed auto
 duplex auto
 nameif outside
 security-level 0
 ip address 16.142.10.100 255.255.255.0
 ipv6 address 2001:400:3:1::100/64
 ipv6 nd suppress-ra
```

```
 ospf mtu-ignore auto
!
interface Ethernet1
 speed auto
 duplex auto
 nameif inside
 security-level 100
 ip address 16.140.10.100 255.255.255.0
 ipv6 address 2001:400:1:1::100/64
 ospf mtu-ignore auto
!
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname coyupix
boot system flash:/cdisk.7.0.0.16
ftp mode passive
names
access-list allow extended permit icmp any any
pager lines 24
logging enable
logging buffered debugging
mtu outside 1500
mtu inside 1500
ipv6 route outside ::/0 2001:400:3:1::1
ipv6 access-list outacl permit icmp6 2001:400:2:1::/64 2001:400:1:1::/64
ipv6 access-list outacl permit tcp 2001:400:2:1::/64 2001:400:1:1::/64 eq telnet
ipv6 access-list outacl permit tcp 2001:400:2:1::/64 2001:400:1:1::/64 eq ftp
ipv6 access-list outacl permit tcp 2001:400:2:1::/64 2001:400:1:1::/64 eq www
no failover
monitor-interface outside
monitor-interface inside
asdm image
no asdm history enable
arp timeout 14400
access-group allow in interface outside
access-group outacl in interface outside
route outside 0.0.0.0 0.0.0.0 16.142.10.1 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:02:00 rpc 0:10:00 h323 0:05:00
h225 1:00:00 mgcp 0:05:00 mgcp-pat 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
no snmp-server location
no snmp-server contact
snmp-server enable traps snmp
fragment size 200 outside
fragment chain 24 outside
fragment size 200 inside
fragment chain 24 inside
sysopt nodnsalias inbound
sysopt nodnsalias outbound
telnet timeout 5
ssh timeout 5
console timeout 0
!
class-map inspection_default
 match default-inspection-traffic
!
!
policy-map global_policy
 class inspection_default
  inspect dns
  inspect ftp
  inspect h323 h225
  inspect h323 ras
```

```
  inspect rsh
  inspect smtp
  inspect sqlnet
  inspect sip
  inspect skinny
  inspect rpc
  inspect xdmcp
  inspect netbios
  inspect mgcp
  inspect tftp
  inspect snmp
!
terminal width 80
service-policy global_policy global
Cryptochecksum:00000000000000000000000000000000
: end
```