

Cisco Applied Mitigation Bulletin: Identifying and Mitigating Exploitation of the IPv6 Routing Header Vulnerability

Document ID: 81722

<http://www.cisco.com/warp/public/707/cisco-amb-20070124-IOS-IPv6.shtml>

Revision 1.0

For Public Release 2007 January 24 1600 UTC (GMT)

Please provide your [feedback](#) on this document.

[Cisco Response](#)
[Device-Specific Mitigation and Identification](#)
[Additional Information](#)
[Revision History](#)
[Cisco Security Procedures](#)
[Related Information](#)

Cisco Response

Vulnerability Characteristics

A vulnerability exists in Cisco IOS software when processing a specially crafted IPv6 ([RFC2460](#)) packet with a Type 0 Routing Header present. Devices running vulnerable IOS software affected by this vulnerability can be exploited remotely by a unauthenticated attacker. Successful exploitation of this vulnerability may allow for arbitrary code execution, the affected device may crash or repeated attempts to exploit this vulnerability could result in a sustained Denial of Service (DoS) condition. The threat vector used to exploit this vulnerability is through the IPv6 protocol via the IPv6 Extension Header – Routing Header, Type 0 (Source Route) which is also susceptible to exploitation through spoofed attacks. This vulnerability is not exploitable through IPv6 multicast packets. The IPv6 Extension Header – Routing Header is identified by a Next Header (NH) value of 43 in the immediately preceding header. This vulnerability is not covered by a CVE ID.

This document contains information to assist Cisco customers in identifying and mitigating attempts to exploit the "**IPv6 Routing Header Vulnerability**". The vulnerability described in this document affects devices running Cisco IOS software and having IPv6 enabled on at least one of its interfaces.

Vulnerable, non-affected and fixed software information is available in the PSIRT Security Advisory at <http://www.cisco.com/warp/public/707/cisco-sa-20070124-IOS-IPv6.shtml>.

Mitigation Technique Overview

Cisco devices provide several countermeasures for the "**IPv6 Routing Header Vulnerability**". The most effective means of exploit prevention is provided by Cisco IOS software through [Infrastructure Protection Access Control Lists \(iACLs\)](#) that filter IPv6 packets with a Type 0 – Routing Header present within the IPv6 header chaining or by disabling the processing of IPv6 Source-Route packets (**no ipv6 source-route**). Exploitation of this vulnerability can also be mitigated through IPv6 access control lists (ACLs) applied to

deny IPv6 packets with any IPv6 Routing Header Type (0 through 255) present within the packet, however if Mobile IPv6 is deployed within the infrastructure this may break and/or disrupt its operations as Mobile IPv6 uses Type 2 – Routing Header (Extension Header) for its operations. Detective controls can be performed by Cisco IOS devices, ASA, PIX, and Firewall Service Module Firewalls through syslog messages and the counter values displayed in output from show commands.

Device–Specific Mitigation and Identification

Specific information on mitigation and identification is available for these devices:

- [Cisco IOS Routers](#)
- [Cisco IOS Switches](#)
- [Cisco ASA, PIX and FWSM Firewalls](#)

Cisco IOS Routers



Caution: The effectiveness of any mitigation technique is dependent on specific customer situations

such as product mix, network architecture, traffic behavior, and organizational mission. As with any configuration change, evaluate the impact of this configuration prior to applying the change.

Mitigation

Infrastructure Protection Access Control Lists (iACL)

The following IPv6 access control list (ACL) policy denies IPv6 Type 0 (Source Route) Routing Header packets sent to the addresses configured on an affected device. All other IPv6 packets sent to the affected device are permitted only from known trusted source networks (ie: management networks, security operations center, network operations center). Added access list entries (ACEs) should be implemented as part of an Infrastructure Protection Access Control List (iACL) policy that is used to minimize the risk and effectiveness of direct infrastructure attacks by explicitly permitting only authorized traffic to the infrastructure equipment while permitting all other transit traffic in accordance with existing security policies and configurations.

Additional information about iACLs is available at [Protecting Your Core: Infrastructure Protection Access Control Lists](#).

The example ACL policy below uses addresses configured on network infrastructure devices from the following IPv6 prefix – 3ffe:ffff::/64. The IPv6 prefix 3ffe:ffff::/64 is further subnetted into /96, /112 and /127 bit prefixes for use on interfaces (management, loopbacks, point-to-point links) and network segments.

Note: Cisco IOS releases prior to 12.4(2)T do not have the ability to filter on specific IPv6 Routing Header Type values using IPv6 ACLs. IOS releases prior to 12.4(2)T can only filter IPv6 packets with the presence of a Routing Header in the IPv6 header chain. In IOS 12.4(2)T a new keyword of "**routing-type**" added the ability to filter on the presence of specific IPv6 Extension Header – Routing Header Type values.

```
!-- If device is running Cisco IOS software release 12.4(2)T or later
!-- Deny all IPv6 Extension Header - Routing Header Type 0 (Source Route)
!-- packets sent to any IPv6 address configured on interfaces of the affected
!-- device (management, loopback, access links and network/user segments),
!-- IPv6 link-local addresses, or for IPv6 packets transiting through the
!-- IPv6 enabled router targeting other IPv6 enabled devices within the
```

```

!-- network infrastructure.

ipv6 access-list ipv6-infrastructure-acl-policy
  deny ipv6 any any routing-type 0

!
!-- Permit/Deny all other IPv6 Layer3 and Layer4 traffic in accordance
!-- with existing security policies and configurations.
!
!-- Apply IPv6 iACL to interface(s) in the inbound direction.

interface GigabitEthernet0/0
  ipv6 address 3ffe:ffff::0:1:0:1111/96
  ipv6 enable
  ipv6 traffic-filter ipv6-infrastructure-acl-policy in

!
!-- If device is running Cisco IOS software release 12.4(2)T or later
!-- Deny all IPv6 Extension Header - Routing Header Type 0 (Source Route)
!-- packets sent to IPv6 addresses configured on interfaces of the affected
!-- device (management, loopback, access links and network/user segments)
!-- or IPv6 link-local addresses.

ipv6 access-list ipv6-infrastructure-acl-policy
  deny ipv6 any host 3ffe:ffff::0:1:0:1111 routing-type 0
  deny ipv6 any host 3ffe:ffff::0:2:0:2222 routing-type 0
  deny ipv6 any host 3ffe:ffff::0:3:0:3333 routing-type 0
  deny ipv6 any host 3ffe:ffff::0:4:0:4444 routing-type 0
  deny ipv6 any host fe80::218:74ff:feb5:a41b routing-type 0
  deny ipv6 any host fe80::218:74ff:feb5:a41b routing-type 0
  deny ipv6 any host fe80::218:74ff:feb5:a419 routing-type 0

!
!-- The following IPv6 Addresses are configured on Loopback interfaces
!-- for management and BGP peering using /128 prefixes.

  deny ipv6 any host 3ffe:ffff::0:f:0:ffff routing-type 0
  deny ipv6 any host 3ffe:ffff::0:f:0:f00d routing-type 0

!
!-- Deny all other IPv6 Type 0 Routing Header traffic sent to the IPv6
!-- prefix used in the configuration of network infrastructure devices.

  deny ipv6 any 3ffe:ffff::/64 routing-type 0

!
!-- Permit/Deny all other IPv6 Layer3 and Layer4 traffic in accordance
!-- with existing security policies and configurations.
!
!-- Apply IPv6 iACL to interface(s) in the inbound direction.

interface GigabitEthernet0/0
  ipv6 address 3ffe:ffff::0:1:0:1111/96
  ipv6 enable
  ipv6 traffic-filter ipv6-infrastructure-acl-policy in

!

```



Caution: If Mobile IPv6 is deployed within the infrastructure, the following ACL policy may disrupt and/or break its operations. Therefore a workaround does not exist for Mobile IPv6.

```
!-- If device is running Cisco IOS software release prior to 12.4(2)T  
!-- Deny all IPv6 Extension Header - Routing Header Type values 0 - 255  
!-- (all type values) packets sent to IPv6 addresses configured on  
!-- interfaces of the affected device (management, loopback, access links  
!-- and network/user segments) or IPv6 link-local addresses.
```

```
ipv6 access-list ipv6-infrastructure-acl-policy  
  deny ipv6 any host 3ffe:ffff::0:1:0:1111 routing  
  deny ipv6 any host 3ffe:ffff::0:2:0:2222 routing  
  deny ipv6 any host 3ffe:ffff::0:3:0:3333 routing  
  deny ipv6 any host 3ffe:ffff::0:4:0:4444 routing  
  deny ipv6 any host fe80::218:74ff:feb5:a41b routing  
  deny ipv6 any host fe80::218:74ff:feb5:a41b routing  
  deny ipv6 any host fe80::218:74ff:feb5:a419 routing
```

```
!-- The following IPv6 Addresses are configured on Loopback interfaces  
!-- for management and BGP peering using /128 prefixes.
```

```
  deny ipv6 any host 3ffe:ffff::0:f:0:ffff routing  
  deny ipv6 any host 3ffe:ffff::0:f:0:f00d routing
```

```
!-- Deny all other IPv6 Type 0 Routing Header traffic sent to the IPv6  
!-- prefix used in the configuration of network infrastructure devices.
```

```
  deny ipv6 any 3ffe:ffff::/64 routing
```

```
!-- Permit/Deny all other IPv6 Layer3 and Layer4 traffic in accordance  
!-- with existing security policies and configurations.
```

```
!  
!-- Apply IPv6 iACL to interface(s) in the inbound direction.
```

```
interface GigabitEthernet0/0  
  ipv6 address 3ffe:ffff::0:1:0:1111/96  
  ipv6 enable  
  ipv6 traffic-filter ipv6-infrastructure-acl-policy in
```

```
!
```

Control Plane Policing

Cisco IOS software releases 12.0S, 12.2SX, 12.2S, 12.3T, 12.4, and 12.4T support the Control Plane Policing (CoPP) feature. CoPP may be configured on a device to protect the management and control planes to minimize the risk and effectiveness of direct infrastructure attacks by explicitly permitting only authorized traffic sent to the infrastructure device in accordance with existing security policies and configurations. The following example can be adapted to your network. This example assumes that IPv6 packets sent to the IPv6 addresses configured on the affected device are to be fully restricted from receiving any IPv6 Type 0 (Source Route) Routing Header packets:

```

!-- Permit all IPv6 Routing Header Type 0 (Source Route) packets sent to
!-- any IPv6 address configured on interfaces of the affected device
!-- (management, loopback, access links and network/user segments) or IPv6
!-- link-local addresses. This traffic will be policed by the CoPP feature.

ipv6 access-list ipv6-deny-routingheader-type0
 permit ipv6 any any routing-type 0

!-- Permit/Deny all other IPv6 Layer3 and Layer4 traffic in accordance
!-- with existing security policies and configurations for traffic that
!-- is authorized to be sent to infrastructure devices on the management
!-- and control planes.
!
!-- Create a Class-Map where the defined IPv6 access-list which permits all
!-- IPv6 Routing Header Type 0 (Source Route) packets will be applied. This
!-- Class-Map will be applied to a Policy-Map used to police traffic via the
!-- CoPP feature.

class-map match-all control-plane-class
 match access-group name ipv6-deny-routingheader-type0

!
!-- Create a Policy-Map where the defined Class-Map will be applied. This
!-- Policy-Map will be applied to the Control-Plane of the affected device
!-- for traffic to be policed by the CoPP feature.

policy-map drop-unauthorized-infra-traffic
 class control-plane-class
  drop

!
!-- Apply the defined Policy-Map to the Control-Plane of the device for
!-- traffic sent to the management and control planes to be policed by the
!-- CoPP feature.

control-plane
 service-policy input drop-unauthorized-infra-traffic

!

```

Note: In the above CoPP examples, the access control list entries (ACEs) which match the potential exploit packets with the "**permit**" action result in these packets being discarded by the policy-map "**drop**" function, while packets that match the "**deny**" action are not affected by the policy-map drop function.

Anti-Spoof Protection via Unicast Reverse Path Forwarding (uRPF) for IPv6

The vulnerability described within this document can be exploited by a spoofed packet. Protection mechanisms for anti-spoofing exist through the proper deployment and configuration of Unicast Reverse Path Forwarding (uRPF) for IPv6. uRPF for IPv6 can detect and drop (discard) IPv6 packets transiting through a router that lack a verifiable IPv6 source addresses. The use of uRPF for IPv6 should not be relied upon to provide 100% protection as spoofed packets may still enter the network through a uRPF enabled interface for which there is a return route to the IPv6 source address within the packet or allowed by anti-spoofing access-lists. Additional information about uRPF for IPv6 is available at [Unicast RPF \(uRPF\) for IPv6 on the Cisco 12000 Series](#) and configuration information for [ipv6 verify unicast reverse-path](#) and [ipv6 verify](#)

[unicast source reachable-via](#) is available at [Cisco IOS 12.3 IPv6 Command Reference](#). iACLs coupled with anti-spoofing protection mechanisms via uRPF for IPv6 provides an added layer of threat mitigation for this vulnerability.

Identification

Infrastructure Protection Access Control Lists (iACL) – Routing Header 'Type 0' Filtering

With an iACL, once the access list is applied, the **show ipv6 access-list** command can be used to identify the number of IPv6 packets being filtered. Filtered packets should be investigated to determine if they are attempts to exploit this vulnerability. Example output for **show ipv6 access-list ipv6-infrastructure-acl-policy**:

```
ios-router#
ios-router#show ipv6 access-list ipv6-infrastructure-acl-policy
IPv6 access list ipv6-infrastructure-acl-policy
  deny ipv6 any any routing-type 0 (156 matches) sequence 10
  --          ACL Policy Truncated          --
  -- Permit/Deny all other IPv6 Layer3 and Layer4 --
  -- traffic in accordance with existing security --
  -- policies and configurations.          --
ios-router#
```

In the above example, access list **ipv6-infrastructure-acl-policy** dropped 156 – IPv6 Type 0 Routing Header packets on ACE sequence-id 10, which is applied in the inbound direction on interface GigabitEthernet0/0.

```
ios-router#
ios-router#show ipv6 access-list ipv6-infrastructure-acl-policy
IPv6 access list ipv6-infrastructure-acl-policy
  deny ipv6 any host 3ffe:ffff::0:1:0:1111 routing-type 0 (9 matches) sequence 10
  deny ipv6 any host 3ffe:ffff::0:2:0:2222 routing-type 0 sequence 20
  deny ipv6 any host 3ffe:ffff::0:3:0:3333 routing-type 0 sequence 30
  deny ipv6 any host 3ffe:ffff::0:4:0:4444 routing-type 0 (127 matches) sequence 40
  deny ipv6 any host FE80::218:74FF:FEB5:A41B routing-type 0 sequence 50
  deny ipv6 any host FE80::218:74FF:FEB5:A41A routing-type 0 sequence 60
  deny ipv6 any host FE80::218:74FF:FEB5:A419 routing-type 0 sequence 70
  deny ipv6 any host 3ffe:ffff::0:F:0:FFFF routing-type 0 sequence 80
  deny ipv6 any host 3ffe:ffff::0:F:0:F00D routing-type 0 sequence 90
  deny ipv6 any 3ffe:ffff::/64 routing-type 0 (173 matches) sequence 100
  --          ACL Policy Truncated          --
  -- Permit/Deny all other IPv6 Layer3 and Layer4 --
  -- traffic in accordance with existing security --
  -- policies and configurations.          --
ios-router#
```

In the above example, access list **ipv6-infrastructure-acl-policy** dropped 9 – IPv6 Type 0 Routing Header packets on ACE sequence-id 10, 127 – IPv6 Type 0 Routing Header packets on ACE sequence-id 40, and 173 – IPv6 Type 0 Routing Header packets on ACE sequence-id 100, which is applied in the inbound direction on interface GigabitEthernet0/0.

Infrastructure Protection Access Control Lists (iACL) – Routing Header 'Type 0 through Type 255' Filtering

With an iACL, once the access list is applied, the **show ipv6 access-list** command can be used to identify the number of IPv6 packets being filtered. Filtered packets should be investigated to determine if they are attempts to exploit this vulnerability. Example output for **show ipv6 access-list ipv6-infrastructure-acl-policy**:

```

ios-router#
ios-router#show ipv6 access-list ipv6-infrastructure-acl-policy
IPv6 access list ipv6-infrastructure-acl-policy
  deny ipv6 any host 3ffe:ffff::0:1:0:1111 routing sequence 10
  deny ipv6 any host 3ffe:ffff::0:2:0:2222 routing (17 matches) sequence 20
  deny ipv6 any host 3ffe:ffff::0:3:0:3333 routing sequence 30
  deny ipv6 any host 3ffe:ffff::0:4:0:4444 routing sequence 40
  deny ipv6 any host FE80::218:74FF:FEB5:A41B routing sequence 50
  deny ipv6 any host FE80::218:74FF:FEB5:A41A routing sequence 60
  deny ipv6 any host FE80::218:74FF:FEB5:A419 routing sequence 70
  deny ipv6 any host 3ffe:ffff::0:F:0:FFFF routing (29 matches) sequence 80
  deny ipv6 any host 3ffe:ffff::0:F:0:F00D routing (77 matches) sequence 90
  deny ipv6 any 3ffe:ffff::/64 routing (137 matches) sequence 100
  --          ACL Policy Truncated          --
  -- Permit/Deny all other IPv6 Layer3 and Layer4 --
  -- traffic in accordance with existing security --
  -- policies and configurations.          --
ios-router#

```

In the above example, access list **ipv6-infrastructure-acl-policy** dropped 17 – IPv6 Routing Header packets on ACE sequence-id 20, 29 – IPv6 Routing Header packets on ACE sequence-id 80, 77 – IPv6 Routing Header packets on ACE sequence-id 90, and 137 – IPv6 Routing Header packets on ACE sequence-id 100, which is applied in the inbound direction on interface GigabitEthernet0/0.

Control Plane Policing (CoPP)

With Control Plane Policing (CoPP), once the policy-map is applied to the **control-plane**, the **show policy-map control-plane** and **show ipv6 access-list** commands can be used to identify the number of packets that have been sent to the management and control planes and dropped by the CoPP policy. Packets dropped by CoPP should be investigated to determine if they are attempts to exploit this vulnerability.

Example output for **show policy-map control-plane** and **show ipv6 access-list ipv6-deny-routingheader-type0**:

```

ios-router#
ios-router#show policy-map control-plane
Control Plane

Service-policy input: drop-unauthorized-infra-traffic

Class-map: control-plane-class (match-all)
  41 packets, 14846 bytes
  5 minute offered rate 3000 bps, drop rate 3000 bps
  Match: access-group name ipv6-deny-routingheader-type0
  drop

Class-map: class-default (match-any)
  1804 packets, 144288 bytes
  5 minute offered rate 4000 bps, drop rate 0 bps
  Match: any
ios-router#
ios-router#show ipv6 access-list ipv6-deny-routingheader-type0
IPv6 access list ipv6-deny-routingheader-type0
  permit ipv6 any any routing-type 0 (41 matches) sequence 10
ios-router#

```

In the above example, the CoPP policy dropped 41 (total) IPv6 packets by access control list (ACL) "**ipv6-deny-routingheader-type0**" which is associated with CoPP.

Anti-Spoof Protection via Unicast Reverse Path Forwarding (uRPF) for IPv6

With uRPF for IPv6 properly deployed and configured throughout the network infrastructure, the **show ipv6 interface**, **show cef drop**, and **show cef interface <type> <slot/port> internal** commands can be used to identify the number of IPv6 packets that uRPF for IPv6 has dropped (discarded).

```
ios-router#
ios-router#show ipv6 interface GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::218:74FF:FEB5:A41B
  No Virtual link-local address(es):
  Global unicast address(es):
    3FFE:FFFF::0:1:0:1111, subnet is 3FFE:FFFF::1:0:0/96
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:7301
    FF02::1:FFB5:A41B
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachable are sent
  Input features: Ingress-Netflow RPF ACL
  Output features: Post-Ingress-Netflow Egress-Netflow
Unicast RPF
  Process Switching:
    0 verification drops
    0 suppressed verification drops
CEF Switching:
  12 verification drops
  0 suppressed verification drops
  Inbound access list infrastructure-acl-policy
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
ios-router#
ios-router#show cef drop
CEF Drop Statistics
Slot  Encap_fail  Unresolved  Unsupported  No_route  No_adj  ChkSum_Err
RP      158           0           0           4         0       0
IPv6 CEF Drop Statistics
Slot  Encap_fail  Unresolved  Unsupported  No_route  No_adj
RP      51         0           0           12       0
ios-router#
ios-router#show cef interface GigabitEthernet 0/0 internal
GigabitEthernet0/0 is up (if_number 2)
  Corresponding hwidb fast_if_number 2
  Corresponding hwidb firstsw->if_number 2
  ICMP redirects are never sent
  Per packet load-sharing is disabled
  IP unicast RPF check is disabled
  Inbound access list is not set
  Outbound access list is not set
  Hardware idb is GigabitEthernet0/0
  Fast switching type 1, interface type 27
  IP CEF switching enabled
  IP Selective flow switching turbo vector
  IP Flow CEF switching turbo vector
  Input fast flags 0x0, Input fast flags2 0x8, Output fast flags 0x0, Output fast flags2 0
  ifindex 2(2)
  Slot 0 Slot unit 0 Unit 0 VC -1
  Transmit limit accumulator 0x0 (0x0)
```

```

IP MTU 1500
Subblocks:
  IPv6 unicast RPF: acl=None, drop=12, sdrop=0
  IPv6: enabled 1 unreachable TRUE redirect TRUE mtu 1500 flags 0x0
  Switching mode is CEF
  Input features: Ingress-Netflow RPF ACL
  Output features: Post-Ingress-Netflow Egress-Netflow
  Inbound access list: infrastructure-acl-policy
ios-router#

```

In the above examples, **uRPF for IPv6** has dropped 12 – IPv6 packets received on interface GigabitEthernet0/0 due to the inability to verify the source address of the IPv6 packets within the Cisco Express Forwarding (CEF) Forwarding Information Base (FIB).

Cisco IOS Switches



Caution: The effectiveness of any mitigation technique is dependent on specific customer situations such as product mix, network architecture, traffic behavior, and organizational mission. As with any configuration change, evaluate the impact of this configuration prior to applying the change.

Mitigation

Infrastructure Protection Access Control Lists (iACL)

The following IPv6 access control list (ACL) policy denies IPv6 Type 0 through Type 255 Routing Header packets sent to the addresses configured on an affected device as switches running Cisco IOS software do not have the ability to filter traffic on specific Routing Header Type values (ie: Extension Header – Routing Header Type 0 – Source Route). All other IPv6 packets sent to the affected device are permitted only from known trusted source networks (ie: management networks, security operations center, network operations center). Added access list entries (ACEs) should be implemented as part of an Infrastructure Protection Access Control List (iACL) policy that is used to minimize the risk and effectiveness of direct infrastructure attacks by explicitly permitting only authorized traffic to the infrastructure equipment while permitting all other transit traffic in accordance with existing security policies and configurations.

Additional information about iACLs is available at [Protecting Your Core: Infrastructure Protection Access Control Lists](#).

The example ACL policy below uses addresses configured on network infrastructure devices from the following IPv6 prefix – 3ffe:ffff::/64. The IPv6 prefix 3ffe:ffff::/64 is further subnetted into /96, /112 and /127 bit prefixes for use on interfaces (management, loopbacks, point-to-point links) and network segments.



Caution: If Mobile IPv6 is deployed within the infrastructure, the following ACL policy may disrupt and/or break its operations. Therefore a workaround does not exist for Mobile IPv6.

```

!-- If switch is running Cisco IOS software
!-- Deny all IPv6 Extension Header - Routing Header Type values 0 - 255
!-- (all type values) packets sent to IPv6 addresses configured on
!-- interfaces of the affected device (management, loopback, access links
!-- and network/user segments) or IPv6 link-local addresses.

ipv6 access-list ipv6-infrastructure-acl-policy
  deny ipv6 any host 3ffe:ffff::0:1:0:1111 routing
  deny ipv6 any host 3ffe:ffff::0:2:0:2222 routing

```

```

deny ipv6 any host 3ffe:ffff::0:3:0:3333 routing
deny ipv6 any host 3ffe:ffff::0:4:0:4444 routing
deny ipv6 any host fe80::218:74ff:fea2:c00 routing

!-- The following IPv6 Addresses are configured on Loopback interfaces
!-- for management and BGP peering using /128 prefixes.

deny ipv6 any host 3ffe:ffff::0:f:0:ffff routing
deny ipv6 any host 3ffe:ffff::0:f:0:f00d routing

!-- Deny all other IPv6 Type 0 Routing Header traffic sent to the IPv6
!-- prefix used in the configuration of network infrastructure devices.

deny ipv6 any 3ffe:ffff::/64 routing

!-- Permit/Deny all other IPv6 Layer3 and Layer4 traffic in accordance
!-- with existing security policies and configurations.
!
!-- Apply IPv6 iACL to interface(s) in the inbound direction.

interface Vlan100
ipv6 address 3ffe:ffff::0:1:0:1111/96
ipv6 enable
ipv6 traffic-filter ipv6-infrastructure-acl-policy in

!

```

Identification

Infrastructure Protection Access Control Lists (iACL)

With an iACL, once the access list is applied, the **show ipv6 access-list** command can be used to identify the number of IPv6 packets being filtered. Filtered packets should be investigated to determine if they are attempts to exploit this vulnerability. Example output for **show ipv6 access-list ipv6-infrastructure-acl-policy**:

```

ios-switch#
ios-switch#show ipv6 access-list ipv6-infrastructure-acl-policy
IPv6 access list ipv6-infrastructure-acl-policy
  deny ipv6 any host 3ffe:ffff::0:1:0:1111 routing sequence 10
  deny ipv6 any host 3ffe:ffff::0:2:0:2222 routing (1 match) sequence 20
  deny ipv6 any host 3ffe:ffff::0:3:0:3333 routing (77 matches) sequence 30
  deny ipv6 any host 3ffe:ffff::0:4:0:4444 routing sequence 40
  deny ipv6 any host FE80::218:74FF:FEA2:C00 routing sequence 50
  deny ipv6 any host 3ffe:ffff::0:F:0:FFFF routing sequence 60
  deny ipv6 any host 3ffe:ffff::0:F:0:F00D routing sequence 70
  deny ipv6 any 3ffe:ffff::/64 routing (256 matches) sequence 80
  -- ACL Policy Truncated --
  -- Permit/Deny all other IPv6 Layer3 and Layer4 --
  -- traffic in accordance with existing security --
  -- policies and configurations. --
ios-switch#

```

In the above example, access list **ipv6-infrastructure-acl-policy** dropped 1 – IPv6 Routing Header packet on ACE sequence-id 20, 77 – IPv6 Routing Header packets on ACE sequence-id 30, and 256 – IPv6 Routing

Header packets on ACE sequence–id 80, which is applied in the inbound direction on interface Vlan100.

Note: The above hit counts displayed are for those packets processed (dropped) in software. For hardware–based IOS switches, an additional command can be used to determine if packets are being dropped in hardware.

Starting with IOS version 12.2(17a)SX (for Supervisor 720) and version 12.2(17d)SXB (for Supervisor 2), the command **show tcam interface vlan <vlan-id> acl <in/out> ipv6** can be used to provide ACE hit counts for IPv6 packets that have been processed in hardware.

```
ios-switch#show tcam interface vlan 100 acl in ipv6

* Global Defaults shared

-----
ICMP Neighbor Discovery Packet Types:
na - neighbor advertisement      ra - router advertisement
ns - neighbor solicit            rs - router solicit
r  - redirect

IPV6 Address Types:
full - IPv6 Full                 eui - IPv6 EUI
eipv4 - IPv6 embeded IPv4
-----

policy-route ipv6 any(eui) 0:FE80::218:74A2:C00/20(eui) (3 matches)
policy-route ipv6 any(full) 0:FE80::218:74A2:C00/20(eui) (1 match)
policy-route ipv6 any(eui) 0:3ffe:ffff::/64(eui)
policy-route ipv6 any(full) 0:3ffe:ffff::/64(eui)
policy-route ipv6 any(eui) 3ffe:ffff::/64(full)
policy-route ipv6 any(full) 3ffe:ffff::/64(full) (3 matches)
policy-route ipv6 any(eui) host 3ffe:ffff::0:1:0:1111(full)
policy-route ipv6 any(full) host 3ffe:ffff::0:1:0:1111(full) (5 matches)
policy-route ipv6 any(eui) host 3ffe:ffff::0:F:0:F00D(full)
policy-route ipv6 any(full) host 3ffe:ffff::0:F:0:F00D(full) (17 matches)
permit      ipv6 any(eui) any
permit      ipv6 any(full) any (5 matches)

ios-switch#
```

In the above example, access list **ipv6–infrastructure–acl–policy** dropped 3 – IPv6 Routing Header packets sent to 3ffe:ffff::/64, 5 – IPv6 Routing Header packets sent to 3ffe:ffff::0:1:0:1111, and 17 – IPv6 Routing Header packets sent to 3ffe:ffff::0:F:0:F00D in hardware for packets being sent through interface Vlan 100. The **show tcam interface vlan <vlan-id> acl <in/out> ipv6 detail** command can optionally be used to display detailed information.

Cisco ASA, PIX and FWSM Firewalls



Caution: The effectiveness of any mitigation technique is dependent on specific customer situations

such as product mix, network architecture, traffic behavior, and organizational mission. As with any configuration change, evaluate the impact of this configuration prior to applying the change.

Mitigation

ASA, PIX, and FWSM – Transit Access Control Lists (tACL)

The following IPv6 access control list (ACL) policy only permits packets sent to the affected device from known trusted source networks (ie: management networks, security operations center, network operations

center – 3ffe:ffff::/112). Added access list entries (ACEs) should be implemented as part of a Transit Access Control List (tACL) policy which is used for filtering transit and edge traffic at network ingress points in accordance with existing security policies and configurations.

Additional information about tACLs is available at [Transit Access Control Lists: Filtering at Your Edge](#).

The example ACL policy below uses addresses configured on network infrastructure devices from the following IPv6 prefix – 3ffe:ffff::/64. The IPv6 prefix 3ffe:ffff::/64 is further subnetted into /96, /112 and /127 bit prefixes for use on interfaces (management, loopbacks, point-to-point links) and network segments.



Caution: If Mobile IPv6 is deployed within the infrastructure, the following ACL policy may disrupt and/or break its operations. Therefore a workaround does not exist for Mobile IPv6.

```
!-- Deny all other traffic to the affected IPv6 device.
```

```
ipv6 access-list ipv6-transit-policy remark -- Deny all other IPv6 traffic sent
to the affected device
ipv6 access-list ipv6-transit-policy deny ip any host 3ffe:ffff::0:1:0:1111
ipv6 access-list ipv6-transit-policy deny ip any host 3ffe:ffff::0:2:0:2222
ipv6 access-list ipv6-transit-policy deny ip any host 3ffe:ffff::0:3:0:3333
ipv6 access-list ipv6-transit-policy deny ip any host 3ffe:ffff::0:4:0:4444
ipv6 access-list ipv6-transit-policy deny ip any host 3ffe:ffff::0:f:0:ffff
ipv6 access-list ipv6-transit-policy deny ip any host 3ffe:ffff::0:f:0:f00d
```

```
!-- Permit/Deny all other IPv6 Layer3 and Layer4 traffic in accordance
!-- with existing security policies and configurations.
!
!-- Apply IPv6 tACL to the outside interface in the inbound direction.
```

```
access-group ipv6-transit-policy in interface outside
```

```
!
```

Identification

ASA, PIX, and FWSM – Transit Access Control Lists (tACL)

```
firewall#show ipv6 access-list ipv6-transit-policy
ipv6 access-list ipv6-transit-policy remark -- Deny all other IPv6 traffic
sent to the affected device --
ipv6 access-list ipv6-transit-policy deny ip any host
3ffe:ffff::0:1:0:1111 (hitcnt=37)
ipv6 access-list ipv6-transit-policy deny ip any host
3ffe:ffff::0:2:0:2222 (hitcnt=0)
ipv6 access-list ipv6-transit-policy deny ip any host
3ffe:ffff::0:3:0:3333 (hitcnt=0)
ipv6 access-list ipv6-transit-policy deny ip any host
3ffe:ffff::0:4:0:4444 (hitcnt=0)
ipv6 access-list ipv6-transit-policy deny ip any host
3ffe:ffff::0:f:0:ffff (hitcnt=119)
ipv6 access-list ipv6-transit-policy deny ip any host
3ffe:ffff::0:f:0:f00d (hitcnt=3)
-- ACL Policy Truncated --
-- Permit/Deny all other IPv6 Layer3 and Layer4 --
-- traffic in accordance with existing security --
```

```
-- policies and configurations.          --
firewall#
```

In the above example, 159 (total) – IPv6 packets have been received from a non-trusted host or network and denied. In addition, the following **syslog** messages will be logged for any attempts that are denied by IPv6 access list **ipv6-transit-policy**:

```
Jan 01 2007 09:58:10: %ASA|PIX|FWSM-4-106023:
Deny icmp src outside:3ffe:ffff::bad:bad:bad
    dst inside:3ffe:ffff::0:f:0:f00d (type 128, code 0) by
    access-group "ipv6-transit-policy"
Jan 01 2007 09:59:38: %ASA|PIX|FWSM-4-106023:
Deny tcp src outside:3ffe:ffff::bad:bad:bad/35321 dst
inside:3ffe:ffff::0:f:0:ffff/22 by access-group "ipv6-transit-policy"
```

Additional information about **syslog** messages for ASA and PIX security appliances is available at [Cisco Security Appliance System Log Messages](#).

Additional information about syslog messages for the FWSM is available at [Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Logging Configuration and System Log Message](#).

Additional Information

THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS AND DOES NOT IMPLY ANY KIND OF GUARANTEE OR WARRANTY, INCLUDING THE WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR USE. YOUR USE OF THE INFORMATION ON THE DOCUMENT OR MATERIALS LINKED FROM THE DOCUMENT IS AT YOUR OWN RISK. CISCO RESERVES THE RIGHT TO CHANGE OR UPDATE THIS DOCUMENT AT ANY TIME.

Revision History

Revision 1.0	2007 January 24	Initial public release.
--------------	-----------------	-------------------------

Cisco Security Procedures

Complete information on reporting security vulnerabilities in Cisco products, obtaining assistance with security incidents, and registering to receive security information from Cisco, is available on Cisco's worldwide website at http://www.cisco.com/en/US/products/products_security_vulnerability_policy.html. This includes instructions for press inquiries regarding Cisco security notices. All Cisco security advisories are available at <http://www.cisco.com/go/psirt>.

Related Information

- [IPv6 Home Page on Cisco Connection Online \(CCO\)](#)
- [Unicast RPF \(uRPF\) for IPv6 on the Cisco 12000 Series](#)
- [Cisco IOS 12.3 IPv6 Command Reference](#)
- [Unicast Reverse Path Forwarding \(uRPF\) Enhancements for the Internet Service Provider \(ISP\) – ISP Edge](#)

