

IPv6 Fundamentals, Design, and Deployment

Volume 2

Version 3.0

Student Guide

Text Part Number: 97-2946-01



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

 Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

DISCLAIMER WARRANTY: THIS CONTENT IS BEING PROVIDED "AS IS." CISCO MAKES AND YOU RECEIVE NO WARRANTIES IN CONNECTION WITH THE CONTENT PROVIDED HEREUNDER, EXPRESS, IMPLIED, STATUTORY OR IN ANY OTHER PROVISION OF THIS CONTENT OR COMMUNICATION BETWEEN CISCO AND YOU. CISCO SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE. This learning product may contain early release content, and while Cisco believes it to be accurate, it falls subject to the disclaimer above.

Table of Contents

Volume 2

<i>IPv6 Multicast Services</i>	<i>5-1</i>
Overview	5-1
Module Objectives	5-1
<i>Implementing Multicast in an IPv6 Network</i>	<i>5-3</i>
Overview	5-3
Objectives	5-3
IPv6 Multicast Addressing	5-4
PIM for IPv6	5-12
Rendezvous Points	5-22
MP-BGP for the IPv6 Multicast Address Family	5-33
How to Implement Multicasting in an IPv6 Network	5-34
IPv6 Multicast Application Example	5-35
NTP	5-35
VideoLAN	5-36
Summary	5-37
References	5-37
<i>Using IPv6 MLD</i>	<i>5-39</i>
Overview	5-39
Objectives	5-39
Multicast Listener Discovery	5-40
MLD Snooping and MLD Group Limits	5-57
Multicast User Authentication and Group Range Support	5-63
Summary	5-66
References	5-66
Module Summary	5-67
Module Self-Check	5-69
Module Self-Check Answer Key	5-70
<i>IPv6 Transition Mechanisms</i>	<i>6-1</i>
Overview	6-1
Module Objectives	6-1
<i>Implementing Dual-Stack</i>	<i>6-3</i>
Overview	6-3
Objectives	6-3
Dual-Stack Applications	6-4
Dual-Stack Node	6-9
The Dual-Stack Approach	6-11
Summary	6-13
Resources	6-13
<i>Describing IPv6 Tunneling Mechanisms</i>	<i>6-15</i>
Overview	6-15
Objectives	6-15
Overlay Tunnels	6-16
Manually Configured Tunnels	6-19
Automatic Tunnels	6-25
Summary	6-36
Module Summary	6-37
Module Self-Check	6-39
Module Self-Check Answer Key	6-41

IPv6 Security	7-1
Overview	7-1
Module Objectives	7-1
Configuring IPv6 ACLs	7-3
Overview	7-3
Objectives	7-3
IPv6 ACLs	7-4
IPv6 ACL Configuration	7-6
Reflexive and Time-Based ACLs	7-9
Reflexive ACLs	7-9
Time-Based ACLs	7-9
Cisco IOS IPv6 Header Filtering	7-13
Cisco IOS New ICMPv6 Types	7-16
Editing of ACLs	7-19
How to Configure ACLs in an IPv6 Environment	7-21
Summary	7-24
Using IPsec, IKE, and VPNs	7-25
Overview	7-25
Objectives	7-25
IPsec, IKE, and VPNs Basics	7-26
IPsec and IKE	7-29
VPN Connections Using IPv6	7-41
Summary	7-50
Discussing Security Issues in an IPv6 Transition Environment	7-51
Overview	7-51
Objectives	7-51
Dual-Stack Issues	7-52
Tunnel Security Issues	7-58
Summary of Tunnel Security Issues	7-61
NAT-PT Security Issues	7-62
ICMP Traffic Requirements	7-63
Summary	7-64
Understanding IPv6 Security Practices	7-65
Overview	7-65
Objectives	7-65
Threats in IPv6 Networks	7-66
Network Scans	7-67
Spoofing	7-68
Fragmentation	7-69
ARP	7-70
DHCP	7-70
Smurf	7-71
Source Routing	7-72
Rogue Devices	7-73
Traffic Redirection	7-74
Attacks on Services	7-75
Build Distributed Security Capability	7-76
Hide Topology when Possible	7-77
Secure the Local Link	7-78
ICMPv6 at Edge—Manage ICMPv6 Traffic	7-81
Develop Mobility Support Plan	7-82
Use Transition Mechanisms as Transport	7-83
Secure the Routing Plane	7-84
Deploy an Early-Warning System	7-85
Summary	7-87
References	7-87

Configuring Cisco IOS Firewall for IPv6	7-89
Overview	7-89
Objectives	7-89
Cisco IOS Firewall for IPv6	7-90
IPv6 Inspection on ISRs	7-95
Implement IPv6 Inspection on ISRs	7-102
Basic ACL Configuration	7-103
External Interface ACL	7-105
Internal Interface ACL	7-105
Zone-Based Policy Firewall for IPv6 on ISRs	7-113
Configuring Zones and Zone Pairs	7-115
The Default Zone	7-118
Configuring a Basic OSI Layer 3 to 4 Interzone Access Policy	7-121
Troubleshooting the Zone-Based Policy Firewall	7-140
Summary	7-145
Module Summary	7-147
Module Self-Check	7-149
Module Self-Check Answer Key	7-152

IPv6 Multicast Services

Overview

Diverse requirements and leveraging of the network for business purposes continually direct the development of technologies to enhance network performance or to increase network capabilities. Services such as multicasting are requirements for networks. This module describes changes that were made to these services to support IP version 6 (IPv6) operations.

Module Objectives

Upon completing this module, you will be able to understand multicast concepts and IPv6 multicast specifics. This ability includes being able to meet these objectives:

- Implement multicasting in an IPv6 network
- Describe IPv6 multicast addressing options, MAC address mappings, and multicast address scoping

Implementing Multicast in an IPv6 Network

Overview

IP version 6 (IPv6) has advanced multicast capabilities that are used for many more core protocol-dependent functions than in IP version 4 (IPv4). These functions include helping link-local neighbors discover each other by using solicited-node multicast, and improving many services that previously relied on broadcast traffic (such as Routing Information Protocol [RIP] neighbors sharing routing information).

In this lesson, you will describe the IPv6 multicast addresses format, including a real-life multicast example. IP multicast is a foundation technology for entertainment applications, such as rich-media streaming, IPTV, video in general, and other one-to-many applications.

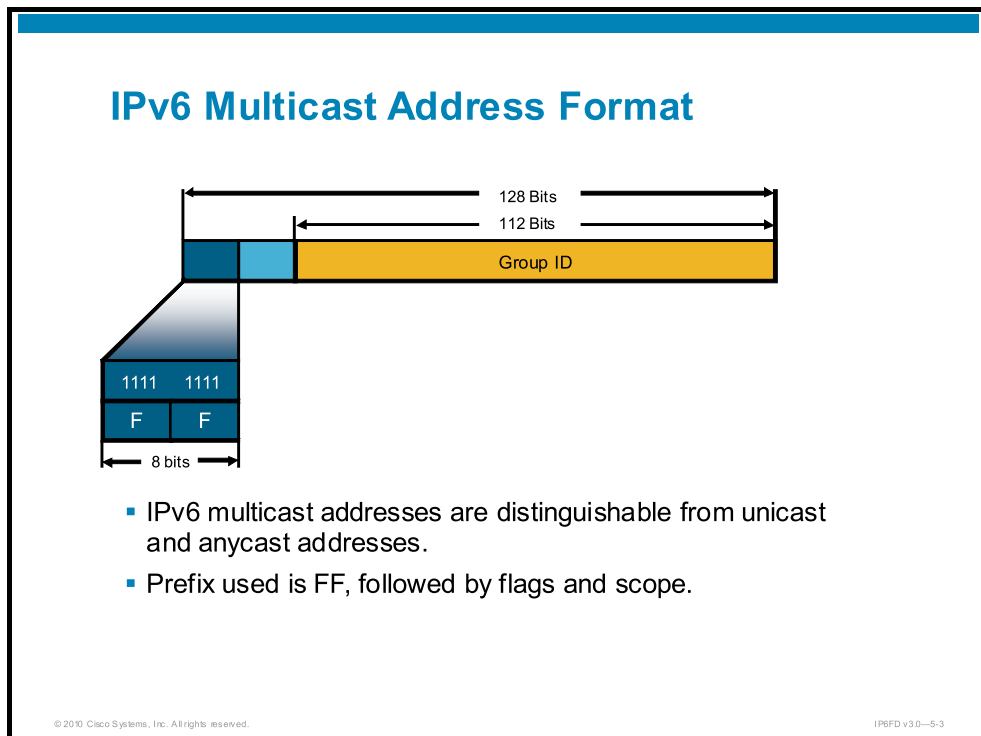
Objectives

Upon completing this lesson, you will be able to describe generic multicast address scoping, identify extended multicast use in IPv6, and describe the principles, protocols, differences, and similarities in IPv4 and IPv6. This ability includes being able to meet these objectives:

- Describe the basic IPv6 multicast addressing and address scoping, and motivations to use IP multicast
- Explain PIM in IPv6 networks
- Describe RP deployments
- Explain MP-BGP for the IPv6 multicast address family
- List the steps for implementing multicasting in an IPv6 network
- Design IPv6 multicast example applications

IPv6 Multicast Addressing

This topic describes the basic IPv6 multicast addressing and address scoping, and motivations to use IP multicast.

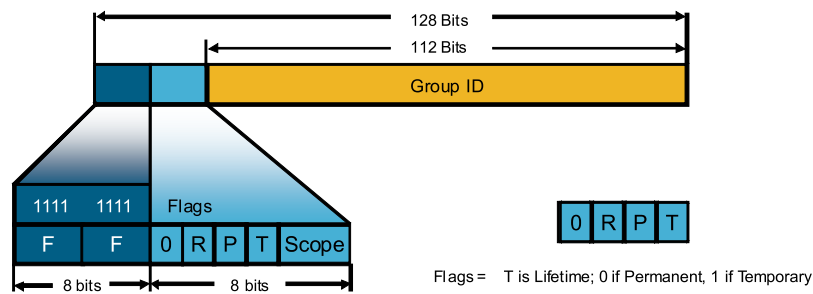


The motivation to use multicasting remains the same as in IPv4: to send only one copy of traffic to multiple receivers. The network devices along the traffic path replicate the data as needed. This provides a great relief of link bandwidth usage compared to several unicast copies of the same stream going to multiple receivers.

The addressing for multicast groups in IPv6 uses a prefix of FF00::/8 (1111 1111). The FF at the start of the address identifies the address as a multicast address.

An IPv6 multicast address is an identifier for a set of interfaces that typically belong to different nodes. A node may belong to any number of multicast groups. A packet that is sent to a multicast address is delivered to all the interfaces that are identified by the multicast group address.

IPv6 Multicast Address Flags



- IPv6 multicast address flags define the lifetime of the address
- Temporary or permanent addresses

The second octet following the prefix defines the lifetime and scope of the multicast address.

Flags is a set of four flags. The high-order flag is reserved, and must be initialized to 0.

T = 0 indicates a permanently assigned (well-known) multicast address that is assigned by the Internet Assigned Numbers Authority (IANA).

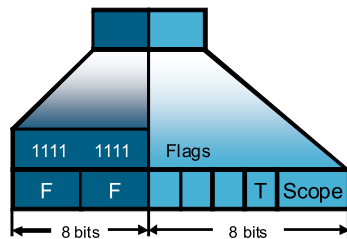
T = 1 indicates a non-permanently assigned (transient) multicast address.

The P flag definition and usage can be found in RFC 3306.

The R flag definition and usage can be found in RFC 3956.

The 112-bit Group ID identifies the multicast group, either permanent or transient, within the given scope.

IPv6 Multicast Address Scope



- IPv6 multicast address scope defines the reach of the multicast address group.
- Scope defines if group traffic should be forwarded across links, routers, domains, etc.

Scope	
1 = Interface-Local	0 0 0 1
2 = Link-Local	0 0 1 0
3 = Reserved	0 0 1 1
4 = Admin-Local	0 1 0 0
5 = Site-Local	0 1 0 1
6 = Unassigned	0 1 1 0
7 = Unassigned	0 1 1 1
8 = Organization-Local	1 0 0 0
9-D = Unassigned	1 0 0 1
	1 1 0 1
E = Global	1 1 1 0
F = Reserved	1 1 1 1

The Scope field uses a single 4-bit field that is embedded in the IPv6 multicast address format.

The Scope field is used to limit the scope of the multicast group. Several values are defined, and others are reserved or unassigned:

- 0 is reserved.
- 1 is interface-local scope.
- 2 is link-local scope.
- 3 is reserved.
- 4 is admin-local scope.
- 5 is site-local scope.
- 6 is unassigned.
- 7 is unassigned.
- 8 is organization-local scope.
- 9 through D are unassigned.
- E is global scope.
- F is reserved.

Consequently, a multicast address that has an interface, link, site, organization, or global scope has a scope parameter of 1, 2, 5, 8, or E, respectively.

For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link-local scope.

The scope values have these various functions:

- Interface-local scope spans only one interface on a node and is useful only for loopback transmission of multicast.

- Link-local multicast scope spans the same topological region as the corresponding unicast scope.
- Admin-local scope is the smallest scope that must be administratively configured—in other words, that is not automatically derived from physical connectivity or other, nonmulticast-related configuration.
- Site-local scope is intended to span a single site.
- Organization-local scope is intended to span multiple sites that belong to a single organization.
- Unassigned scopes are available for administrators to define additional multicast regions.

Routers must not forward any multicast packets beyond the scope that is indicated by the Scope field in the destination multicast address.

Note	The only scope which gets filtered automatically is the link-local scope—traffic to FF02::/8 group addresses does not get routed off the local interface. All other scopes, such as site-local, organization-local, and so on require explicit filters to be applied on the borders. Such filters are easily implemented using the deny statements in access lists.
-------------	---

Nodes must not originate a packet to a multicast address whose Scope field contains the reserved value 0. If such a packet is received, it must be silently dropped. Nodes should not originate a packet to a multicast address whose Scope field contains the reserved value F. If such a packet is sent or received, it must be treated the same as packets that are destined to a global (scope E) multicast address.

Predefined Multicast Addresses

Group of all IPv6 nodes:

- Scope 1:
interface-local
FF01:0:0:0:0:0:0:1
- Scope 2
link-local
FF02:0:0:0:0:0:0:1

Group of all IPv6 routers:

- Scope 1
interface-local
FF01:0:0:0:0:0:0:2
- Scope 2
link-local FF02:0:0:0:0:0:0:2
- Scope 5
site-local
FF05:0:0:0:0:0:0:2

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-6

Two multicast addresses identify the group of all IPv6 nodes within scope 1 (interface-local) or scope 2 (link-local):

FF01::1
FF02::1

Three multicast addresses identify the group of all IPv6 routers within scope 1 (interface-local), scope 2 (link-local), or scope 5 (site-local):

FF01::2
FF02::2
FF05::2

These multicast addresses are reserved and must not be assigned to any multicast group:

FF00::
FF01::
...
FF0E::
FF0F::

IPv6 Solicited-Node Multicast Address Format

- One solicited-node multicast group for every configured or automatic IPv6 address
- Multicast address with a link-local scope
- Formed by a prefix and the right-most 24 bits of every unicast and anycast address
- Used to resolve link-layer address from an IPv6 address

Solicited-node address format:

FF02	0000	0000	0000	0000	0001	FFXX	XXXX
------	------	------	------	------	------	------	------

IPv6 address:

2001	0DB8	0001	0000	0001	0800	200E	8C6C
------	------	------	------	------	------	------	------

Solicited-node multicast address:

FF02	0000	0000	0000	0000	0001	FF0E	8C6C
------	------	------	------	------	------	------	------

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6 v3.0—5-7

Solicited-node multicast addresses are used to discover the link-layer address (Layer 2) from a IPv6 (Layer 3) address. The process itself is very similar as the ARP process in IPv4.

Note For more information on link-layer processes refer to the *IPv6 Operations* module.

The format of the IPv6 solicited-node address is FF02::1:FFXX:XXXX. Solicited-node multicast addresses are computed as a function of the unicast and anycast addresses of a node. A solicited-node multicast address is formed by taking the low-order 24 bits of an address (unicast or anycast) and appending those bits to the prefix FF02::1:FF00::/104. Doing so results in a multicast address in the range FF02::1:FF00:0000 to FF02::1:FFFF:FFFF. For example, the solicited-node multicast address that corresponds to the IPv6 address 2001:0DB8::1:800:200E:8C6C is FF02::1:FF0E:8C6C.

IPv6 addresses that differ only in the high-order bits (for example, because of multiple high-order prefixes that are associated with different aggregations) map to the same solicited-node address, thus reducing the number of multicast addresses that a node must join.

A node is required to compute and join (on the appropriate interface) the associated solicited-node multicast addresses for all unicast and anycast addresses that have been configured (manually or automatically) for the node interfaces.

Generic IPv6 Multicast Application Scope

New IPv6 multicast application address with global scope:

- FF specifies multicast
- 1 specifies a temporary address
- E specifies a global scope
- X is any hexadecimal value

Temporary multicast address with global scope:



- Used by applications that transmit data to a group with active listeners over the IPv6 Internet

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-8

A new multicast application with a global scope can be deployed on the Internet at any address that meets this criteria:

FF1E:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX

The multicast address is composed of the following:

- FF specifies that the address is multicast.
- 1 specifies a temporary address (not a reserved, permanent, IANA-registered multicast address).
- E specifies a global scope.
- X can be any hexadecimal value.

Note For a full explanation of multicast address flags, see the RFC 3306 document. Unicast-based multicast addresses and source-specific multicast addresses have also a reserved range described in this document.

See the RFC 3956 document for details on embedded-RP addressing.

IPv4 and IPv6 Multicast Comparison

IP Service	IPv4 Solution	IPv6 Solution
Addressing Range	32-bit, Class D	128-bit (112-bit group)
Routing	Protocol independent, all IGPs and MGBP	Protocol independent, all IGPs and MBGP with IPv6 multicast SAFI
Forwarding	PIM-DM, PIM-SM, PIM-SSM, PIM-bidirectional	PIM-SM, PIM-SSM, PIM-bidirectional
Group Management	IGMPv1, v2, v3	MLDv1, v2
Domain Control	Boundary, border	Scope identifier
Interdomain Solutions	MSDP across independent PIM domains	Single RP within globally shared domains

© 2010 Cisco Systems, Inc. All rights reserved.

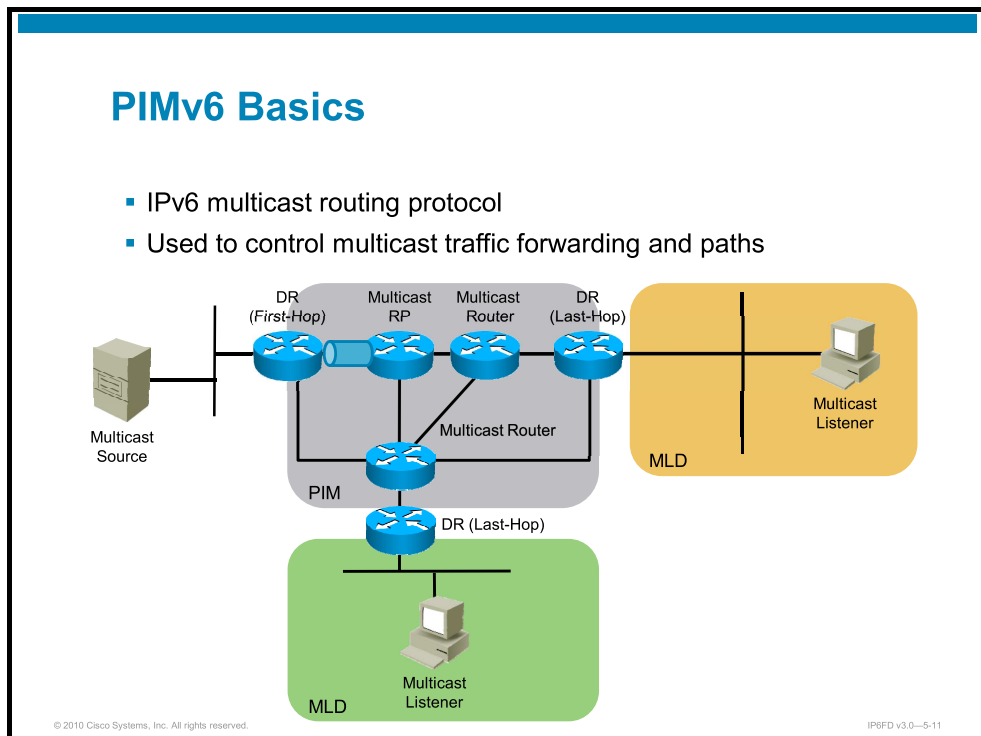
IP6FD v3.0—5-9

These are the most prominent changes in IPv6 multicast:

- The IPv6 multicast address space is much larger than the space in IPv4.
- Multiprotocol Border Gateway Protocol version 4 (MP-BGPv4) multiprotocol support allows the population of separate routing paths for IPv6 unicast and multicast traffic.
- The Cisco IPv6 implementation does not support Protocol Independent Multicast dense mode (PIM-DM) but it is defined as a standard in RFC 3973, *Protocol Independent Multicast - Dense Mode*. However, because of resource consumption, dense mode is rarely used.
- IPv6 multicast uses Multicast Listener Discovery (MLD) for group management (join and leave reports and queries), whereas IPv4 uses Internet Group Management Protocol (IGMP).
- IPv6 multicast is scoped by using IPv6 scoping rules.
- IPv6 uses a single rendezvous point (RP) for interdomain multicast.

PIM for IPv6

This topic describes the Protocol Independent Multicast (PIM) routing protocol in IPv6 networks.



PIM is used between routers so that they can track which multicast packets to forward to each other and to their directly connected LANs. PIM works independently of the unicast routing protocol, to perform send or receive multicast route updates like other protocols do. Regardless of which unicast routing protocols are used in the LAN to populate the unicast routing table, Cisco IOS PIM uses the existing unicast table content to perform the Reverse Path Forwarding (RPF) check, instead of building and maintaining its own separate routing table.

You can configure IPv6 multicast to use either PIM sparse mode (PIM-SM) or PIM source-specific multicast (PIM-SSM) operation, or you can use both PIM-SM and PIM-SSM together in your network.

PIMv6 Terminology

The following terms have special significance for PIM-SM:

- Rendezvous point (RP)
- Designated router (DR)
- Multicast Routing Information Base (MRIB)
- Multicast Forwarding Information Base (MFIB)
- Upstream
- Downstream
- PIM Multicast Border Router (PMBR)

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—5-12

- An RP is a router that has been configured as the root of the nonsource-specific distribution tree for a multicast group. Join messages from receivers for a group are sent toward the RP, and data from senders is sent to the RP so that receivers can discover who the senders are and can start to receive traffic that is destined for the group.
- A shared-media LAN such as Ethernet may have multiple PIM-SM routers that are connected to it. Just one of these routers, the designated router (DR), acts on behalf of directly connected hosts with respect to the PIM-SM protocol. One DR is elected per interface (LAN or otherwise) by using a simple election process.
- The Multicast Routing Information Base (MRIB) is the multicast topology table, which is typically derived from the unicast routing table or routing protocols (such as MP-BGP) that carry multicast-specific topology information. In PIM-SM, the MRIB is used to determine where to send join-prune messages. A secondary function of the MRIB is to provide routing metrics for destination addresses. These metrics are used when sending and processing assert messages.

Routing clients use the services that the MRIB provides to instantiate routing entries and to retrieve changes that other clients make to routing entries. In addition to routing clients, MRIB also has forwarding clients (Multicast Forwarding Information Base [MFIB] instances) and special clients such as MLD. MFIB retrieves its forwarding entries from the MRIB and notifies the MRIB of any events that are related to packet reception. These notifications can either be explicitly requested by routing clients or spontaneously generated by the MFIB.

Another important function of the MRIB is to enable the coordination of multiple routing clients in establishing multicast connectivity within the same multicast session. MRIB also enables coordination between MLD and routing protocols.

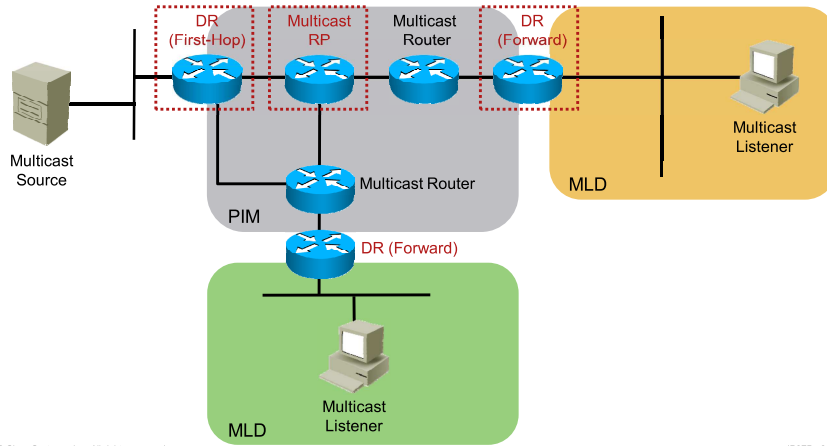
- The MFIB is a platform- and routing-protocol-independent library for IPv6 software. The main purpose of the MFIB is to give a Cisco IOS platform an interface with which to read the IPv6 multicast forwarding table and notifications when the forwarding table changes. The information that the MFIB provides has clearly defined forwarding semantics and is designed to make it easy for the platform to translate to its specific hardware or software forwarding mechanisms.

When routing or topology changes occur in the network, the IPv6 routing table is updated, and those changes are reflected in the MFIB. The MFIB maintains next-hop address information that is based on the information in the IPv6 routing table. Because a one-to-one correlation exists between MFIB entries and routing table entries, the MFIB contains all known routes and eliminates the need for route-cache maintenance that is associated with switching paths such as fast switching and optimum switching.

- The term “upstream” denotes toward the root of the tree. The root of the tree may be either the source or the RP, depending on the context.
- The term “downstream” denotes away from the root of the tree.
- A PIM Multicast Border Router (PMBR) joins a PIM domain with another multicast domain.

PIMv6 Overview

- Protocol-independent: MRIB built on underlying unicast routing table
- DR and RP router discover optimal paths to server group traffic



PIM relies on an underlying topology-gathering protocol to populate the Multicast Routing Information Base (MRIB) routing table. The routes in this table may be taken directly from the unicast routing table, or they may be different routes that are provided by a separate routing protocol such as MP-BGP. Regardless of how the MRIB is created, its primary role in the PIM protocol is to provide the next-hop router along a multicast-capable path to each destination subnet. The MRIB is used to determine the next-hop neighbor to which any PIM join-prune message is sent. Data flows along the reverse path of the join messages. Thus, in contrast to the unicast RIB, which specifies the next hop that a data packet takes to get to some subnet, the MRIB gives reverse-path information and indicates the path that a multicast data packet takes from its origin subnet to the router that has the MRIB.

MLD is used to allow link-local receivers to report interest in receiving specific multicast groups to their link-local router. PIM allows DRs that serve multicast receivers (last-hop routers), DRs that serve multicast sources (first-hop routers), and RPs to discover optimal forwarding paths for multicast control and data packets. The figure shows the regions of the multicast network in which PIM and MLD are active. Note that routers can concurrently be first-hop routers for some multicast addresses and last-hop routers for other multicast addresses.

Protocol-independent means that the multicast routing table can be built from the underlying unicast routing table. So regardless of whether the unicast routing table was built by RIP next generation (RIPng), Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First version 3 (OSPFv3), or any other protocol, PIM can construct the multicast routing table. The only limitation in this case will be that multicast traffic and unicast traffic will follow the same path.

PIM can also run on top of a multicast routing table that is built by a multicast-capable routing protocol such as MP-BGP. In this case, the multicast traffic and unicast traffic can take different paths in the network rather than those that are derived from the unicast routing table.

PIM packets are exchanged between all three of these main components:

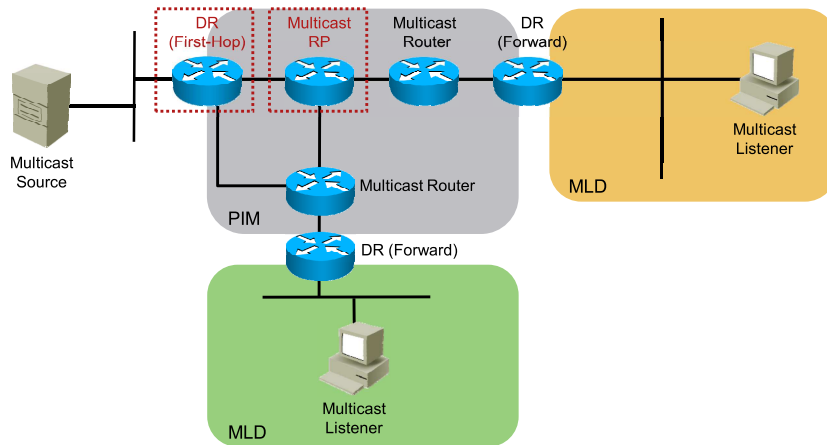
- **DR (first-hop) to and from RP:** This part of PIM enables new multicast sources to register with the RP that is associated with a given multicast address. This registration is done via PIM tunnels, in which the multicast packets are tunneled by the first-hop router to the RP. Afterward, the RP can obtain the multicast stream directly from the source. This process is called building the shortest path tree (SPT) from the RP to the multicast source.
- **DR (last hop) to and from RP:** This part of PIM enables the DR that serves new multicast receivers to build a flooding path back to the RP, through the intervening multicast routers, in such a way that the DR can receive a specific stream. This results in the RP flooding packets toward the DR (last-hop) router which, in turn, results in a completed shared tree.
- **DR (last-hop) to and from DR (first-hop):** This process is called building the SPT, in which the DR that serves the listeners wants to get the multicast stream directly from the DR that serves the multicast source (DR-Source).

Although the PIM process is a straightforward concept, PIM is a complex protocol.

Note	PIM-SM is described in detail in RFC 4601, <i>Protocol Independent Multicast – Sparse Mode (PIM-SM)</i> .
-------------	---

PIMv6 Phase One

- DR (first-hop) router is the first router after source.
- RP routers consolidate groups from different sources.



Phase one of the network configuration for IPv6 multicast is to configure the DR and RP routers.

Except for implementations that use techniques such as a bootstrap router (BSR) or embedded RP, multicast routers are configured to know the location of the RP (IPv6 unicast or anycast address) for each multicast address group. Also, the RP is configured to know for which multicast groups it will serve as RP. This situation is a static RP assignment.

The following example enables multicast routing on all interfaces. Entering this command also enables multicast forwarding for PIM and MLD on all enabled interfaces of the router:

```
Router(config)#ipv6 multicast-routing
```

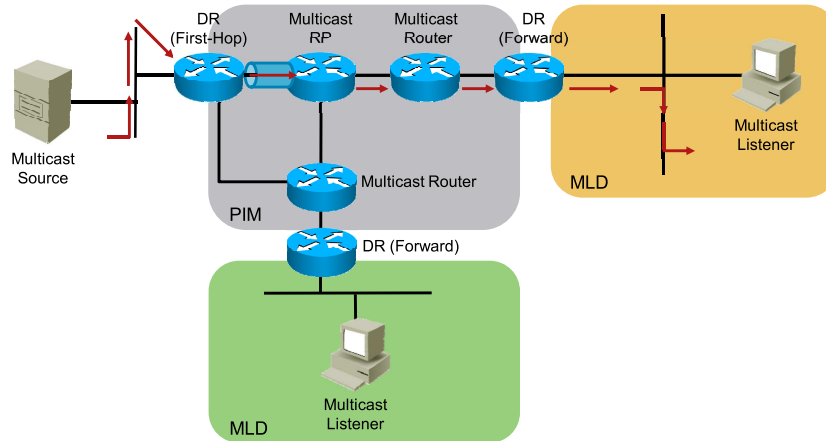
The following example shows how to configure a router to use PIM-SM, using 2001:0DB8::1 as the RP:

```
Router(config)#ipv6 pim rp-address 2001:0DB8::1
```

In phase one of the PIM-SM protocol overview, a multicast receiver expresses its interest in receiving traffic that is destined for a multicast group. Typically, the receiver does this by using IGMP or MLD, but other mechanisms might also serve this purpose. One of the local routers of the receiver is elected as the DR for that subnet. After receiving the expression of interest from the receiver, the DR then sends a PIM join message toward the RP for that multicast group. This join message is known as a (*,G) join message because it joins group G for all sources to that group. The (*,G) join message travels hop-by-hop toward the RP for the group, and in each router through which the message passes, multicast tree state for group G is instantiated. Eventually, the (*,G) join message either reaches the RP or reaches a router that already has (*,G) join state for that group. When many receivers join the group, their join messages converge on the RP and form a distribution tree for group G that is rooted at the RP. This process is known as the RP Tree (RPT) and is also known as the shared tree because it is shared by all sources that send to that group. Join messages are resent periodically, so long as the receiver remains in the group. When all receivers on a leaf-network leave the group, the DR sends a PIM (*,G) prune message toward the RP for that multicast group. However, if the prune message is not sent for any reason, the state eventually times out.

PIMv6 Phase Two

- Interest from the listener (signaled by using MLD to join to a multicast group) starts the forwarding of multicast group data across the multicast tree.



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--5-15

A multicast data sender just starts sending data that is destined for a multicast group. The local router of the sender (DR) takes those data packets, unicast-encapsulates them, and sends them directly to the RP. The RP receives these encapsulated data packets, de-encapsulates them, and forwards them on to the shared tree. The packets then follow the (*,G) multicast tree state in the routers on the RPT, being replicated wherever the RPT branches and eventually reaching all the receivers for that multicast group. The process of encapsulating data packets to the RP is called registering, and the encapsulation packets are known as PIM Register packets. At the end, multicast traffic flows encapsulated to the RP and then natively over the RPT to the multicast receivers.

Phase two of the PIM-SM protocol overview is register-stop.

Register-encapsulation of data packets is inefficient for two reasons:

- Encapsulation and de-encapsulation might be relatively expensive operations for a router to perform, depending on whether the router has appropriate hardware for these tasks.
- Traveling all the way to the RP and then back down the shared tree can result in the packets traveling a relatively long distance to reach receivers that are close to the sender. For some applications, this increased latency or bandwidth consumption is undesirable.

Although register-encapsulation may continue indefinitely, the RP typically chooses to switch to native forwarding for the previous reasons. To do so, when the RP receives a register-encapsulated data packet from source S on group G, it typically initiates an (S,G) source-specific join toward S. This join message travels hop-by-hop toward source S, instantiating (S,G) multicast tree state in the routers along the path. (S,G) multicast tree state is used only to forward packets for group G if those packets come from source S. Eventually the join message reaches the source S subnet or a router that already has (S,G) multicast tree state. Then packets from source S start to flow, following the (S,G) tree state toward the RP. Along the path toward the RP, these data packets may also reach routers with (*,G) state; if so, the packets can shortcut onto the RP tree at that point.

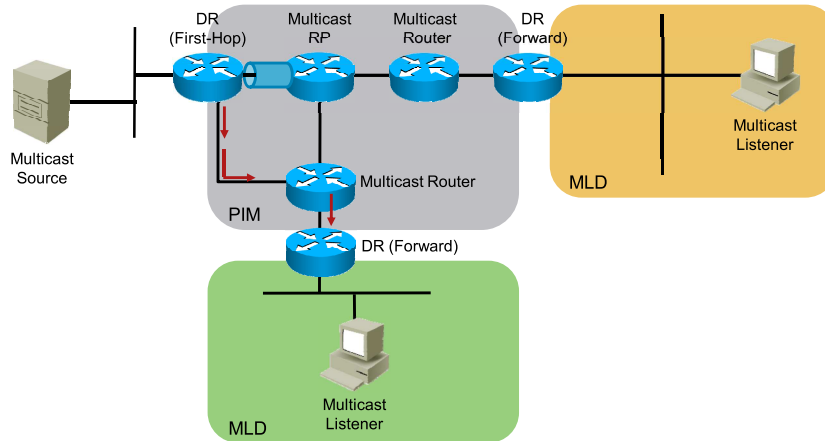
While the RP is in the process of joining the source-specific tree for source S, the data packets continue to be encapsulated to the RP. When packets from source S also start to arrive natively at the RP, the RP receives two copies of each packet. At this point, the RP starts to discard the encapsulated copy of these packets, and it sends a register-stop message back to the source S DR, to prevent the DR from unnecessarily encapsulating the packets.

At the end of phase two, traffic is flowing natively from source S, along a source-specific tree, to the RP, and from there along the shared tree to the receivers. Where the two trees intersect, traffic may transfer from the source-specific tree to the RP tree and so avoid taking a long detour via the RP.

Note A sender may start sending before or after a receiver joins the group, and so phase two may happen before the shared tree to the receiver is built.

PIMv6 Phase Three

- Path through RP might not be optimal.
- Transfer from a shared tree to source-specific, shortest-path tree for lower latency and (possibly) better bandwidth utilization.



Although having the RP join back toward the source removes the encapsulation overhead, it does not completely optimize the forwarding paths. For many receivers, the route via the RP can involve a significant detour when compared with the shortest path from the source to the receiver.

To obtain lower latencies or more-efficient bandwidth utilization, a router (typically the DR) on the receiver LAN may optionally initiate a transfer from the shared tree to a source-specific SPT. To do so, the router issues an (S,G) join message toward source S. This message instantiates state in the routers along the path to source S. Eventually, this join message either reaches the source S subnet or reaches a router that already has (S,G) state. When this happens, data packets from source S start to flow, following the (S,G) state until they reach the receiver.

PIM is used to manage the flow of these packets, pruning back the stream when the last listener has stopped listening to a specific multicast group.

At this point, the receiver (or a router that is upstream of the receiver) is receiving two copies of the data: one from the SPT and one from the RPT. When the first traffic starts to arrive from the SPT, the DR or upstream router starts to drop the packets from source S that are for group G and that arrive via the RP tree. In addition, the DR sends an (S,G) prune message toward the RP. This message is known as an (S,G,rpt) prune. The prune message travels hop by hop, instantiating state along the path toward the RP and indicating that traffic from source S and for group G should not be forwarded in this direction. The prune is propagated until it reaches the RP or a router that still needs the traffic from source S for other receivers.

By now, the receiver is receiving traffic from source S, along the SPT between the receiver and S. In addition, the RP is receiving the traffic from source S, but this traffic is no longer reaching the receiver along the RP tree. As far as the receiver is concerned, this is the final distribution tree.

The following example sets the SPT threshold to infinity, to prevent switchover to the source tree when a source starts sending traffic.

```
Router(config)#ipv6 pim spt-threshold infinity
```

Note	Cisco routers, by default, immediately join the source tree after receiving the first few multicast packets from the shared tree (RP tree).
-------------	---

Rendezvous Points

This topic describes RP deployments.

Rendezvous Points Deployments

- Static RP:
 - For PIM-SM and Bidir-PIM
 - Provides group-to-RP mapping, no RP redundancy (yet)
- BSR:
 - Provides group-to-RP mapping and RP redundancy
- Embedded-RP:
 - Easy to deploy
 - Group-to-RP mapping only, no RP redundancy (yet)
 - PIM-SM only, no Bidir-PIM

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-18

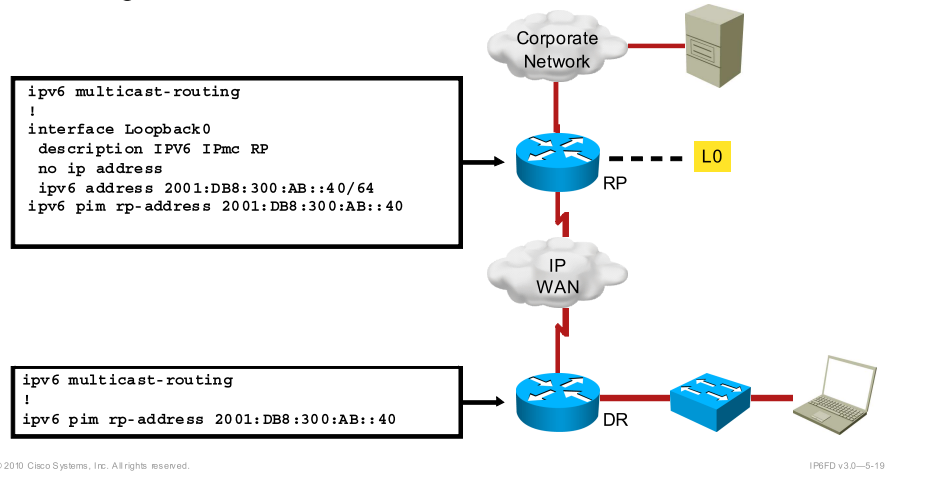
Currently there are three alternatives for RP deployments:

- Static RP configuration
- RP deployment using the BSR protocol
- Embedded-RP

Only deployment using BSR offers RP redundancy.

Static RP

RP and last-hop routers are statically configured with a single RP IPv6 address.



In static RPs, all PIM-enabled routers are statically configured with the RP IPv6 address that should be used for a given multicast group. Whenever the forwarding router discovers a new listener on an attached link, for a new multicast group, the router builds state toward that RP. This implies that the first-hop DR and all other multicast-enabled routers in the domain are configured to register with that specific RP for multicast streams.

There are several drawbacks to this scheme:

- There is no redundancy or failover capability in a static RP configuration. If the RP fails over, multicast fails.

Note There is, however, a stateless RP failover option, to announce the second RP with a shorter prefix.

- Static configurations require you to manually configure each multicast router and maintain the static mappings whenever the RP address changes—a process that is time consuming and error prone.

In effect, the static RP implementation has many of the same limitations of static routes: simple to configure, but with poor scalability and poor availability because of the single-sourcing of the RP address.

Embedded RP

- Embeds the RP address into a multicast group address
- Redefines what was an 8-bit reserved field into a 4-bit reserved and 4-bit R field:
 - R field allows provision of 16 RPs on embedded address.
 - 32-bit group ID field provides for 232 multicast groups per RP.



Flags = 0RPT, R = 1, P = 1, T = 1 = RP Address Embedded

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-20

Embedded RPs eliminates the need for static configuration of RPs on multicast DRs. The RP address for a given multicast group (address) is encoded inside the multicast group.

The figure shows the format for the multicast group address. This address is derived via the mechanism that is described in RFC 3956, *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*, and is also a unicast-prefix multicast group address. A multicast group address uses these special rules, with fields that are described from left to right:

- **FF:** This field is a multicast designator.
- **Flags:** This field contains 4 bits, defined from left to right:
 - **First bit:** This bit is reserved.
 - **Second bit:** This bit is known as the R field. When set to 1, this bit specifies that this multicast address contains an embedded RP address.
 - **Third bit:** This bit is known as the P field. The bit specifies whether the address contains a unicast prefix-based address. The field is set to 1 if it contains a unicast prefix-based address. The unicast prefix that is included in the address is for the multicast source home prefix, making this multicast group globally unique. If the R bit is set to 1, the P bit must be set to 1 as well.
 - **Fourth bit:** This bit is known as the T field. The bit is set to 0 if this is a permanently assigned, well-known multicast address and to 1 if the address is temporary. For all embedded RPs, the T bit must be set to 1.
- **Scope:** This field has the same values as for any multicast address:
 - Link-local scope is 2.
 - Site-local scope is 5.
 - Global scope is E.
- **Rsvd (reserved):** This field is commonly set to all zeros. If this address was not an embedded RP multicast address, this field would be an 8-bit reserved field.

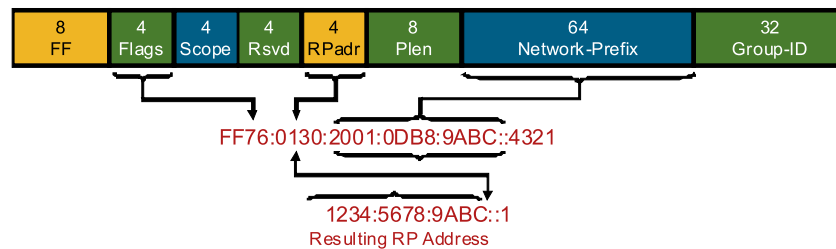
- **RPadr:** This 4-bit field specifies RPs 1 through 16 on the /64 network prefix that is embedded farther along in the address.
- **Plen:** This value must not be 0 (SSM) or greater than 64.
- **Network-Prefix:** This field is the network prefix for the RP.
- **Group-ID:** This field is the multicast group ID, which is either a well-known ID, such as NTP, or a temporary ID that is invented by an enterprise. Remember that the Flags field identifies the address as well known or transient.

Note Embedded-RP is described in detail in RFC 3956.

Embedded RP (Cont.)

Address example:

- 16 RP addresses per network prefix
- 232 multicast groups per RP
- Guaranteed unique because enterprise-assigned /64 network is used in address
- RFC 3956 and RFC 3306



This example shows the resulting RP address that is taken from the multicast group address. The 4 bits that are taken from the previously reserved range are used to specify one of 16 addresses, 1 through 15 (0 is reserved). Therefore, an address that ends in ::17 cannot be an embedded RP.

In the example, the flag bits are set to 0111, which indicates an embedded RP, unicast-based, temporary multicast group.

By using prefix-embedding, any enterprise that has a /64-bit prefix can generate a unique multicast address (or group)—guaranteed to be unique Internet-wide—without consulting any organization or registering with IANA.

Embedded RPs add the ability to include 16 possible addresses for RPs for that multicast group in the group ID (multicast address). Because the RP address is encoded in the address and located within the IPv6 network of the inventing enterprise, that enterprise can set up an RP to service its new multicast group. The enterprise can set up a maximum of 16 RPs for this multicast group.

Embedded RP (Cont.)

Use of embedded RP

- Embedded RP can be considered an automatic replacement to static RP configuration.
- Unsupported routers can be configured statically or via BSR for embedded RPs.
- Embedded RP does not provide RP redundancy, as BSR or anycast RP can.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-22

Embedded RPs are ideal as a replacement for static RPs. No manual configuration of RP addresses is required because the address is embedded within a multicast group. However, if the RP address needs to be changed, then the multicast address needs to change, which requires informing all interested applications. Routers that do not support embedded RPs can be configured statically or can use other methods, such as BSR.

Routers that serve as embedded RPs must be configured for that specific multicast group. This safeguard mechanism prohibits poor selection of RPs. By specifying for which multicast groups an RP will stream traffic, the likelihood that a high-rate group address will be sourced from a low-throughput RP is reduced. Without this safeguard, you could select RPs simply by specifying the multicast address with an embedded RP address, without consulting the network team. Doing so could allow an embedded RP address to specify an RP that could not manage the multicast traffic; for example, an RP that was designed only for low-throughput multicast.

Note The embedded RP solves the problem of helping multicast routers choose the correct RP but does not address redundancy of RPs in case the RP goes down.

The embedded RP is an IPv6-only multicast feature, which simplifies the process of managing mapping between multicast groups and RPs.

Embedded RP (Cont.)

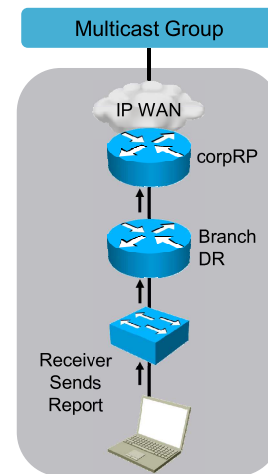
Configuration Example

- RP to be used as an embedded-RP needs to be configured with address/group range
- All other non-RP routers require no special configuration

```
ipv6 pim rp-address 2001:db8:c003:111d::1 ERP
!
ipv6 access-list ERP
permit ipv6 any ff7e:140:2001:db8:c003:111d::/96
```

```
corpRP#show ipv6 pim group

FF7E:140:2001:DB8:45A:31::/96*
RP      : 2001:DB8:45A:31::1    Protocol: SM
Client  : Embedded
Groups  : 1
Info    : RPF: Se0/0.1, FE80::210:7FF:FEDD:40
```



In this example, the multicast address has the RP addresses embedded. The number 1 of the available 15 RPs is used. The branch router is configured as a PIM client and knows that the multicast address contains an embedded RP. The group tag shows the encoding FF7E, in which the “7” signifies to the PIM routers that an embedded RP is being used. The group tag includes the source multicast node unicast home prefix that is the location of the RP.

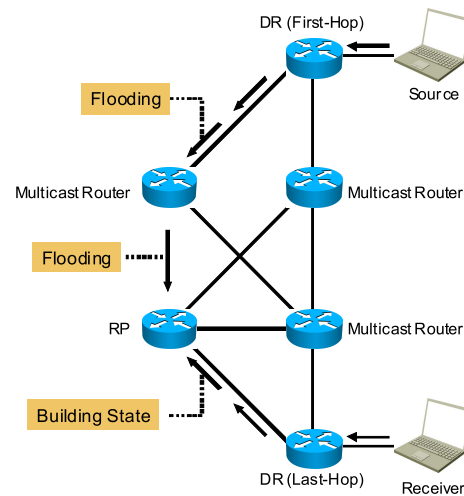
```
corpRP#show ipv6 pim group
```

```
FF7E:140:2001:0DB8:C003:111D::*
RP      :      2001:DB8:C003:111D::1
Protocol:    SM
Client      :      Embedded
Groups      :      1
Info       :    RPF: Se0/0.1, FE80::210:7FF:FEDD:40
```

RP Redundancy

If RP fails, the multicast session fails:

- Reliance on a single point of failure
- Need to provide RP redundancy:
 - Using a manually configured anycast IPv6 RP address
 - Using BSR protocol



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6 v3.0—5-24

Only one RP can be active at one time for each multicast group in any source multicast (ASM). If a DR first-hop router (link-local with a multicast sender) registers with one RP, but multicast receiver routers (link-local with multicast receivers) are seeking the stream from a different RP, then the multicast session fails. Thus, both multicast senders and multicast receivers must be configured to use the same single RP, regardless of whether RP addresses are statically configured or configured automatically via embedded RP. For availability reasons, configuring RP addresses automatically is often unacceptable. ISP and enterprise users cannot, for all applications, accept the service levels that a single machine can support. A mechanism to provide higher availability for multicast streaming is required.

The figure shows how relying on one RP for multicast traffic distribution reduces application availability.

RP Redundancy (Cont.)

- Mechanisms and issues.
- IPv4 RP redundancy solutions for IPv6 implementation:
 - Static RP configuration cannot provide for redundancy by itself.
 - BSR and Auto-RP in IPv4 are not considered as optimal solutions to anycast:
 - Long convergence times
 - Active protocol operations required in all routers
- **BSR and Anycast RP are the only available RP redundancy solutions for IPv6.**

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—5-25

These mechanisms are available for IPv4 RP redundancy:

- Static RPs cannot provide for redundancy. The DR first-hop and last-hop routers must both use a single, configured RP to deliver the stream to receivers. If that RP is lost, the multicast session fails.
- Multicast Source Discovery Protocol (MSDP) is defined for IPv4 but not for IPv6. By using MSDP, IPv4 RPs can find out about other RPs that have active sources and can obtain the stream for distribution from themselves as well. MSDP is most often used for interdomain multicasting but can also be used within a domain to provide RP redundancy.
- BSR (in which one or more routers in the domain provide group-to-RP information for all devices, through message flooding) and Auto-RP (in which RPs multicast the groups for which they provide services to a mapping agent, which in turn multicasts a consolidated set of group-to-RP throughout the domain) are also used in IPv4. However, BSR and Auto-RP in IPv4 have several operational problems:

Both Auto-RP and BSR are slow to converge:

- Auto-RP, by default, can take as long as 3 minutes from the time the active RP fails until the RP Mapping Agent sends information about an alternative RP for a given multicast group destination.
- BSR, by default, can require 150 seconds from the time the active RP fails to fail over to an alternate RP. When this 150-second timer expires, the BSR floods information that specifies another RP throughout the multicast domain.

Auto-RP and BSR are active mechanisms, requiring all PIM routers in the domain to implement specific rules and protocols to make the process work properly.

BSR and Anycast RP are currently the only solutions for IPv6 RP redundancy.

Anycast

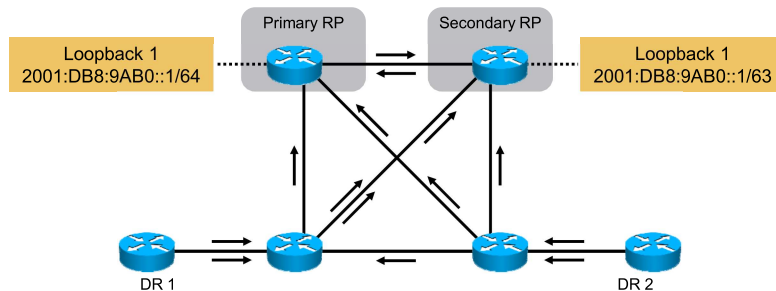
The PIM anycast is an alternative; the solution requires all RPs in an anycast set to be configured with unicast addresses of each other RP and to share an anycast address. When a DR registers with one RP, that RP passes the register information to all other RPs in the set, and each RP creates the state that is needed to satisfy last-hop DR requests.

Anycast RP with prefix-length arbitration is a simple solution for RP redundancy and provides failover as quickly as the routing protocol converges. This approach is a Cisco concept at this point and is covered in detail in the following pages.

RP Redundancy (Cont.)

Alternative RP redundancy solution:

- Designate a primary and a secondary RP (add more if necessary) for an anycast group.
- Configure primary RP with the longest prefix (e.g., /64) on the loopback; secondary RP has a shorter prefix (e.g., /63).



Anycast provides a mechanism for routers to reach the topologically closest RP when multiple RPs exist at the same IPv6 unicast address. This concept keeps a primary RP but also permits multiple backup RPs and smooth failover if the primary RP is lost.

Using the Anycast RP Solution

This solution supports multiple RPs; if one RP fails, the others can pick up the orphaned multicast receivers. The difference is that, on the anycast address configuration (which is configured as loopback interfaces), the RP that is designed to be the primary RP is configured with a longer prefix length than the secondary routers are. Routing protocols use a longest-match-first process, so as long as the primary RP is advertising reachability (in other words, is still working), all multicast DR-Forward routers converge on that address. Only when the primary RP becomes silent does the listening population shift to the backup RP that has the next-longest prefix.

The mechanism must be in place for all RPs to stay in sync on active sources.

Cisco calls this approach anycast RP with prefix-length arbitration. Anycast RP provides the ability to have multiple RPs, and the prefix length allows administrative control over primary and backup roles.

MP-BGP for the IPv6 Multicast Address Family

This topic describes MP-BGP for the IPv6 multicast address family.

Multiprotocol BGP

- Can carry routing information for IPv6 unicast and IPv6 multicast by using different address families.
 - **address-family ipv6 unicast**
 - **address-family ipv6 multicast**
- Handles different next hops for unicast and multicast prefixes.
- Contains route tables that can be divergent, allowing traffic to be managed explicitly.
- Prefixes need to be injected into BGP by using the **network** command or by redistribution:
 - The prefix must first exist in the unicast routing table.
- Multicast routing needs PIM within the autonomous system; BGP is used only to transport prefixes (not to handle join or leave requests, etc.)

© 2010 Cisco Systems, Inc. All rights reserved.

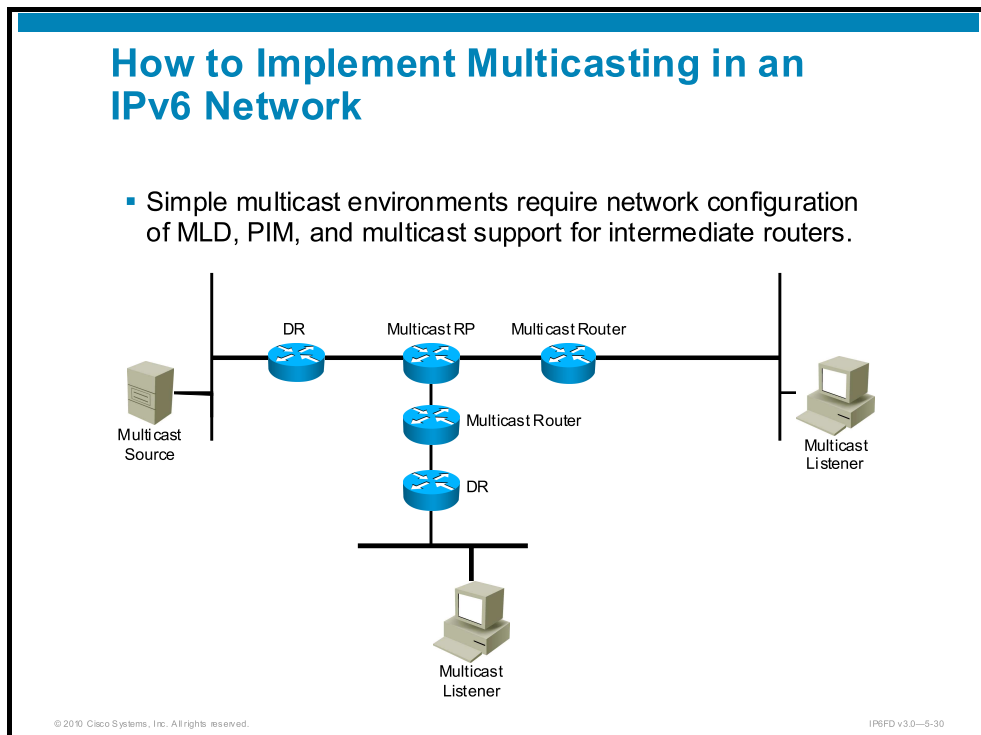
IPv6FD v3.0—5-28

MP-BGP is a routing protocol that can carry routes for different address families concurrently. MP-BGP can carry IPv6 unicast and IPv6 multicast routes (as well as other address families, such as IPv4).

If PIM is routing-protocol independent and can build a multicast routing table from the unicast routing information, why would you want to run MP-BGP and explicitly distribute multicast routes? In many network deployments, it is desirable for unicast and multicast addresses to peer in different places. By running MP-BGP independently for both IPv6 unicast and IPv6 multicast, you can use advanced BGP features to manage the traffic separately.

How to Implement Multicasting in an IPv6 Network

This topic describes the steps for implementing multicasting in an IPv6 network.



Follow these steps to implement multicasting for the intradomain network by using MLD and PIM:

Step 1 Configure MLD on all DR routers and multicast receivers.

MLD must be configured on all the DR routers and multicast hosts that will act as multicast receivers. When multicast routing is enabled globally, MLD is automatically enabled on the IPv6 interfaces. This configuration is required to allow the listeners to notify the DR of the addresses that they would like to listen to. MLD support on the hosts is operating-system dependent, but should be available on IPv6-enabled interfaces. MLD is usually configured on all DR routers because sources and listeners can be present anywhere in a multicast network.

Step 2 Configure PIM on all DR and RP routers.

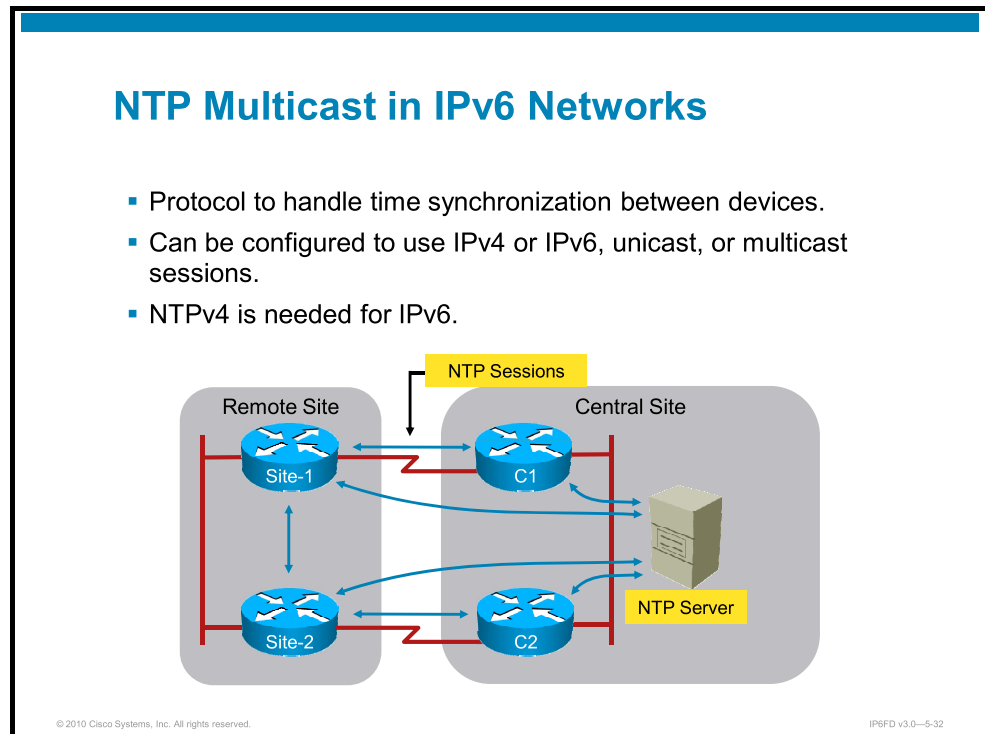
This step is required for sources and listeners to find each other via the RP and to build shared trees or SPTs between each other.

Step 3 Configure multicast support on all intermediate routers.

PIM routers form peer link-local relationships with their neighbors, and only multicast-enabled routers can build the shared and source-specific trees to deliver packets.

IPv6 Multicast Application Example

This topic presents examples of IPv6 multicast applications.



NTP

NTP is a simple protocol that uses UDP port 123. RFC 1305 describes the protocol as well as the architectural framework and in-depth implementation recommendations.

The NTP architectural framework specifies a hierarchy of time servers that use stratum values from 1 to 16 to indicate their relative accuracy. The most accurate servers, which use external clocks (Global Positioning System [GPS] receivers are commonly used because of their low cost), are stratum 1 servers. Any server that synchronizes to a stratum X server advertises itself as stratum X+1 server.

Two NTP servers communicate in client-server or peer-to-peer mode (the desired peering mode is configured manually on the servers and is indicated in the outgoing NTP packets). The fundamental difference is the synchronization behavior: An NTP server can synchronize to a peer with better stratum but never synchronizes to its client (regardless of the client stratum).

The upstream NTP servers must be defined manually in the client NTP server (or the NTP client); there is no autodiscovery mechanism in the client-server relationships. If you use NTP on a Cisco IOS device, the NTP peers must be defined only on one side. The Cisco IOS implementation of the NTP server automatically creates a peer-to-peer association when it receives an incoming NTP packet that indicates that the remote IP host wants to establish a peering relationship.

The communication between the NTP server and clients can be established via IPv6 multicast.

VideoLAN

VideoLAN is a very popular multicast application used to play audio and video streams over IPv4 or IPv6.

VideoLAN

- Multimedia application for playing multicast audio or video streams
- VLC Media Player is the core application



© VideoLAN, www.videolan.org

© 2010 Cisco Systems, Inc. All rights reserved. IP6FD v3.0--5-33

VideoLan Client (VLC) Media Player is a free and open-source cross-platform multimedia player and framework that plays most multimedia files, as well as DVD, Audio CD, VCD, and various streaming protocols, regardless of transport used (IPv4 or IPv6).

It is simple to use, yet very powerful and extendable.

Summary

This topic summarizes the key points that were discussed in this lesson.

Summary

- IP multicast is used to pass one copy of traffic to multiple receivers.
- The four major components in multicasting are DRs, multicast receivers and sources, and RPs.
- PIM allows DRs and RPs to discover the best forwarding paths for multicast control and data packets.
- Embedded RPs eliminate the need for static configuration of RPs on the first-hop and last-hop routers.
- RP redundancy can be achieved either by using BSR or by using an embedded anycast RP and the prefix-length/anycast RP concept.
- MP-BGP can carry multicast routing information.
- NTP and VLC are examples of multicast IPv6 applications.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6 v3.0—5-34

References

For additional information, refer to these resources:

- Cisco Systems, Inc. *IPv6 Multicast Configuration for Cisco IOS Software*.
http://www.cisco.com/en/US/tech/tk828/technologies_white_paper09186a0080203f7a.shtml.
- Cisco Systems, Inc. *Cisco IOS IPv6 Configuration Guide, Release 12.4*, Implementing IPv6 Multicast. San Jose, California, August 2008.
<http://www.cisco.com/en/US/partner/docs/ios/ipv6/configuration/guide/ip6-multicast.html#wp1059064>.
- IETF. *RFC 3956: Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*. November 2004. <http://tools.ietf.org/html/rfc3956>.
- IETF. *RFC 5059: Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*. January 2008. <http://tools.ietf.org/html/rfc5059>.
- IETF. *RFC 2858: Multiprotocol Extensions for BGP-4*. June 2000.
<http://tools.ietf.org/html/rfc2858>.

Using IPv6 MLD

Overview

This lesson will talk about advanced IP version 6 (IPv6) multicast features, comprising Multicast Listener Discovery (MLD) features such as MLD snooping and MLD group limits, as well as support for multicast authentication, profiles, and group ranges.

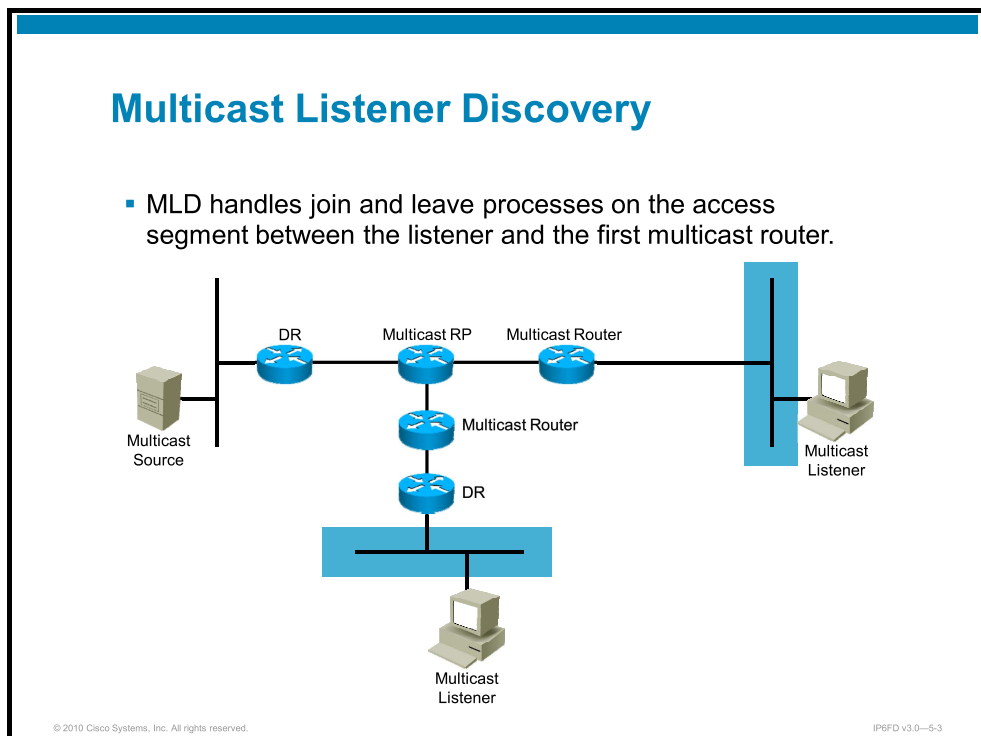
Objectives

Upon completing this lesson, you will be able to describe IPv6 multicast addressing options, MAC address mappings, and multicast address scoping. This ability includes being able to meet these objectives:

- Describe and use the MLD protocol in multicast networks
- Configure MLD snooping and MLD group limits
- Configure multicast authentication

Multicast Listener Discovery

This topic describes the MLD protocol and process in IPv6 networks.



The Multicast Listener Discovery (MLD) protocol is used on the LAN segment that interconnects the listeners and a multicast router.

By using MLD, the listeners and the router control the participation of the listeners in multicast groups.

If the router does not have the traffic for a multicast group that a listener requests by using MLD, then the router requests the group data by using Protocol Independent Multicast (PIM) from its upstream multicast router.

Multicast Listener Discovery: MLDv1

- MLD version number specifics:
 - MLDv1 (RFC 2710) is similar to IGMPv2 (RFC 2236).
 - MLDv2 (RFC 3810) is similar to IGMPv3 (RFC 3376).
- LAN switching technologies:
 - CGMP (IPv4)
 - IGMP snooping (IPv4)
 - MLD snooping (IPv6)
- MLDv1 is equivalent to IGMPv2 in IPv4:
 - MLDv1 messages are transported over ICMPv6.
 - MLDv1 uses link-local source addresses.
 - MLDv1 packets use the Router Alert option in IPv6 header (RFC 2711).

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--5-4

To start implementing multicasting in the campus network, users must first define who receives the multicast.

MLD specifies the protocol that an IPv6 multicast router uses to discover the presence of multicast receivers (nodes that want to receive multicast packets) on its directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. This information is then provided to whichever multicast routing protocol (for example, PIM) the router uses, to ensure that multicast packets are delivered to all links in which there are interested receivers. The MLD protocol provides a means to automatically control and limit the flow of multicast traffic throughout the network by using special multicast queriers and hosts.

The difference between multicast queriers and hosts is as follows:

- A querier is a network device, such as a router, that sends query messages to discover which network devices are members of a given multicast group.
- A host is a receiver, including routers, that sends report messages to inform the querier of a host membership.

A set of queriers and hosts that receive multicast data streams from the same source is called a multicast group. Queriers and hosts use MLD reports to join and leave multicast groups and to begin receiving group traffic.

MLD is an asymmetric protocol that specifies different behaviors for multicast receivers and routers. A node, for example, always performs the listener part of MLD. This means that the node wants to listen to specific multicast streams and needs to notify the upstream router about those specific addresses. The router can perform the listener part of MLD as well (it can also want to listen to specific multicast addresses), but the router always performs the router part of MLD.

MLD version 1 (MLDv1) is equivalent to IP version 4 (IPv4) multicast Internet Group Management Protocol (IGMP) version 2 (IGMPv2), in which MLD messages are always between multicast routers and link-local multicast nodes. As such, MLD messages always use link-local addressing. One important difference is that MLD uses Internet Control Message Protocol (ICMP) version 6 (ICMPv6) message types (IP Protocol 58), rather than IGMP (IP Protocol 2) message types.

MLD messaging uses the IPv6 router alert header option (within a hop-by-hop option, as described in RFC 2711, *IPv6 Router Alert Option*) to signal to the MLD router to look into these packets because some hop-by-hop service is needed. Use of the router alert option is needed to force the MLD router to look into packets that are not destined to the router.

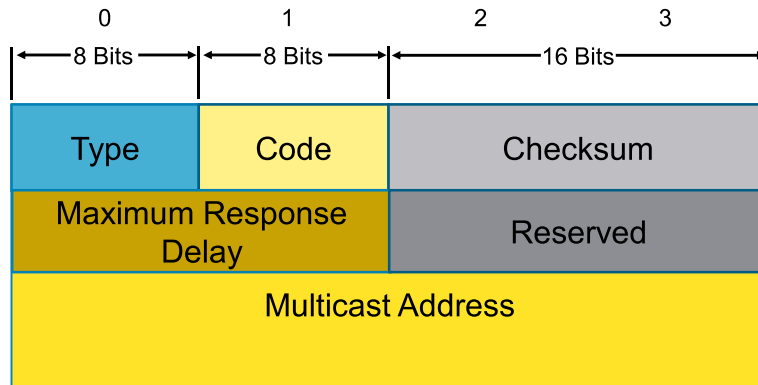
There are currently two versions of MLD: MLDv1 (which is based on IGMPv2) and MLD version 2 (MLDv2, which is based on IGMP version 3 [IGMPv3]). MLDv2 provides support for link-local receivers to specify sources for multicast groups—called source filtering—as well as to specify just the multicast group. This support, in turn, provides support for Source Specific Multicast (SSM, RFC 3569), in which the last-hop router can obtain the multicast stream directly from the multicast source first-hop router.

In an enterprise scenario, switching mechanisms cause multicast packets to be sent to ports that are not multicast listeners, because the switch has no way to determine which multicast listeners are on which ports. MLD snooping, described in RFC 4541, *Considerations for IGMP and MLD Snooping Switches*, describes how switches can listen to MLD packets to determine how to more efficiently forward multicast packets. This snooping is usually done in hardware by high-end switches. Cisco Group Management Protocol is a Cisco proprietary software protocol that runs between the multicast router and the switch. Cisco Group Management Protocol enables the Cisco multicast router to understand IGMP messages that hosts send and informs the switch about the information in the IGMP packet. This solution to the multicast flooding problem on switches is usually done for lower-end devices.

Note	Cisco Group Management Protocol is a proprietary Cisco protocol that has a very narrow user base and has been replaced by IGMP snooping, which is standardized functionality.
-------------	---

MLDv1 Message Format

- MLD is a subprotocol of ICMPv6 and has three message types:
 - Multicast listener query
 - Multicast listener report
 - Multicast listener done



MLD is a subprotocol of ICMPv6; that is, MLD message types are a subset of ICMPv6 messages, and MLD messages are identified in IPv6 packets by a preceding Next Header value of 58. All the MLD messages that are described here are sent with a link-local IPv6 source address, an IPv6 hop limit of 1, and an IPv6 Router Alert option [RTR-ALERT] in a Hop-by-Hop Options header.

There are three types of MLD messages:

- Multicast listener query (Type = decimal 130)
 - General query is used to learn which multicast addresses have listeners on an attached link.
 - Multicast address specific query is used to learn whether a particular multicast address has any listeners on an attached link.

Note The previous types are differentiated by the contents of the Multicast Address field.

- Multicast listener report (Type = decimal 131)
- Multicast listener done (Type = decimal 132)

The code is initiated to zero by the sender and ignored by receivers.

The checksum is the standard ICMPv6 checksum, which covers the entire MLD message plus a pseudo-header of IPv6 header fields.

The Maximum Response Delay field is meaningful only in multicast listener query messages and specifies the maximum allowed delay before sending a responding multicast listener report, in units of milliseconds. In all other messages, the field is set to zero by the sender and is ignored by receivers.

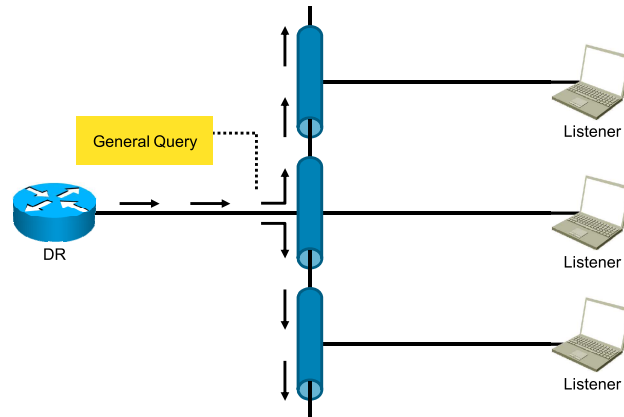
The Reserved field is initialized to zero by the sender and is ignored by receivers.

In a multicast listener query message, the Multicast Address field is set to zero when sending a general query and to a specific IPv6 multicast address when sending a multicast address specific query.

In a multicast listener report or multicast listener done message, the Multicast Address field holds a specific IPv6 multicast address to which the message sender is listening or ceasing to listen, respectively.

General MLDv1 Process: General Query

- General query message
- “Which multicast addresses have listeners on the link?”



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6 v3.0-5-6

Routers use MLD to learn which multicast addresses have listeners on each of their attached links. Each router keeps a list for each attached link; these lists indicate which multicast addresses have listeners on the link. Each router also keeps a timer that is associated with each address. Note that the router needs to learn only that listeners for a given multicast address are present on a link. The router does not need to learn the identity (for example, unicast address) of those listeners or even know how many listeners are present.

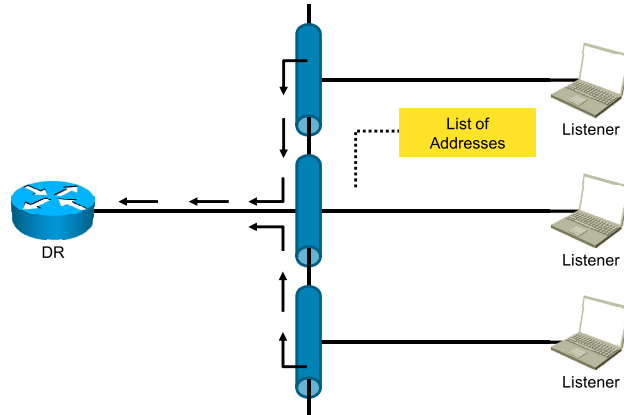
MLDv1 routers send general queries on each link, asking for reports from each multicast link-local receiver (node). This information tells the designated router (DR) for the link which multicast groups the link-local receivers are listening to.

For each attached link, a router selects one of its link-local unicast addresses on that link to be used as the IPv6 source address in all MLD packets that the router transmits on that link.

General queries are sent to the link-scope all-nodes multicast address (FF02::1), with a Multicast Address field of 0 and a Maximum Response Delay that matches the Query Response Interval.

General MLDv1 Process: Report

- Report message.
- Nodes reply with the list of multicast groups they are receiving.



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-7

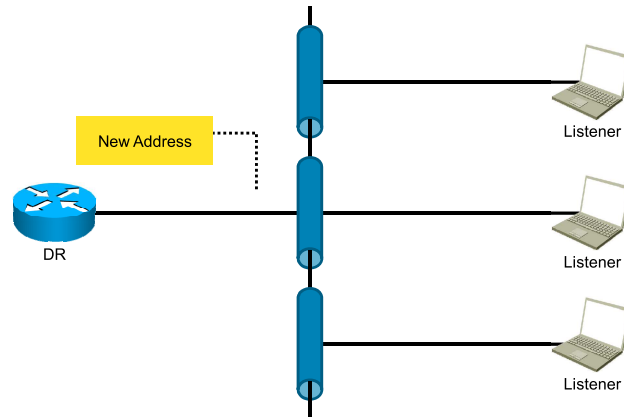
When a node receives a general query, it sets a delay timer for each multicast address to which it is listening on the interface from which it received the query, excluding the link-scope all-nodes address and any multicast addresses of scope 0 (reserved) or 1 (node-local). Each timer is set to a different random value, using the highest clock granularity that is available on the node; this granularity is selected from the range [0, Maximum Response Delay] with a Maximum Response Delay as specified in the query packet. If a timer for any address is already running, that timer is reset to the new random value only if the requested Maximum Response Delay is less than the remaining value of the running timer. If the query packet specifies a Maximum Response Delay of zero, each timer is effectively set to zero, and the action that is specified for timer expiration is performed immediately.

If a node timer for a particular multicast address on a particular interface expires, the node transmits a report to that address via that interface. The address that is reported is carried in both the IPv6 Destination Address field and the MLD Multicast Address field of the report packet. The IPv6 hop limit of 1 and the presence of a link-local IPv6 source address prevent the packet from traveling beyond the link to which the reporting interface is attached.

If a node receives another node report from an interface for a multicast address while the node has a timer running for that same address on that interface, then the node stops its timer and does not send a report for that address, thus suppressing duplicate reports on the link.

General MLDv1 Process: Report (Cont.)

- Node joining a group.
- Using an MLD report, a node requests a multicast group from the router.



© 2010 Cisco Systems, Inc. All rights reserved.

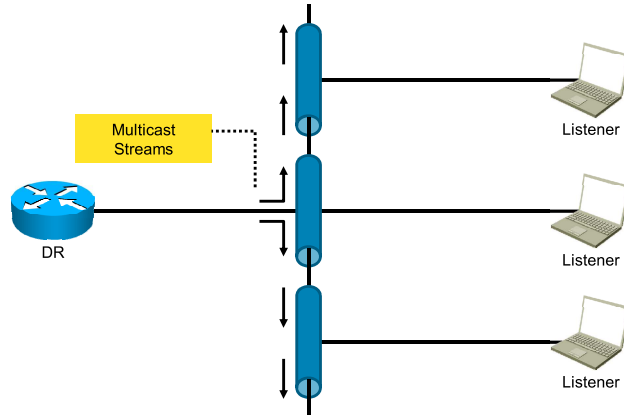
IPv6 v3.0-5-8

When a node starts listening to a multicast address on an interface, it should immediately transmit an unsolicited report for that address on that interface, in case it is the first listener on the link. In case the initial report is lost or damaged, it should be repeated once or twice after short delays (as determined by the Unsolicited Report Interval). A simple way to accomplish this repetition is to send the initial report and then act as if a multicast address specific query was received for that address, and set a timer appropriately.

These reports are unsolicited.

General MLDv1 Process: Report (Cont.)

- In response to an MLD report, the router procures the traffic for a multicast group (e.g., by using a PIM join message).
- Router starts transmitting the group data to the listener on the link.



© 2010 Cisco Systems, Inc. All rights reserved.

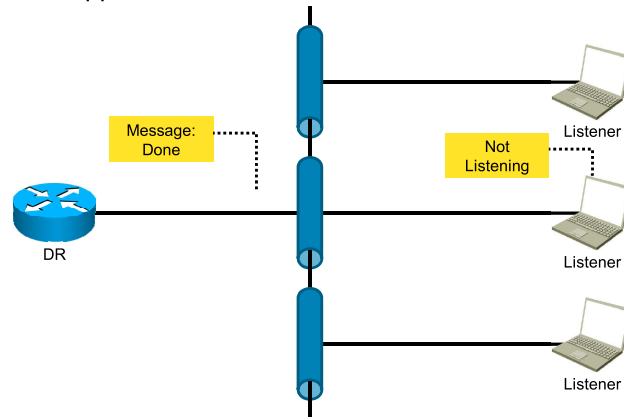
IPv6FD v3.0—5-9

When a router receives a report from a link, if the reported address is not already present in the router list of multicast addresses that have listeners on that link, then the reported address is added to the list, its timer is set to the Multicast Listener Interval, and its appearance is made known to the router multicast routing component. If a report is received for a multicast address that is already present in the router list, then the timer for that address is reset to the Multicast Listener Interval. If an address timer expires, it is assumed that no other listeners for that address are present on the link, so address is deleted from the list and its disappearance is made known to the multicast routing component.

Each multicast stream is obtained independently. When the stream arrives, the router floods the packets to the appropriate link.

General MLDv1 Process: Report (Cont.)

- Node leaving a group.
- If there are other known listeners for that group, this message might be suppressed.



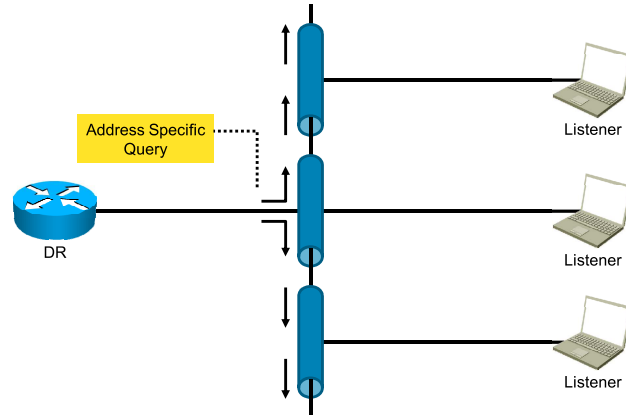
© 2010 Cisco Systems, Inc. All rights reserved.

IPv6 v3.0—5-10

When a node ceases to listen to a multicast address on an interface, it should send a single multicast listener done message to the link-scope all-routers multicast address (FF02::2), carrying in its Multicast Address field the address to which it is ceasing to listen. If the most recent report message for the node was suppressed by hearing another report message, the node might not send anything because another listener for that address is likely still present on the same link.

General MLDv1 Process: Multicast Address Specific Query

- Address specific query.
- In response to an MLD leave, router asks whether any others are still interested in the group.



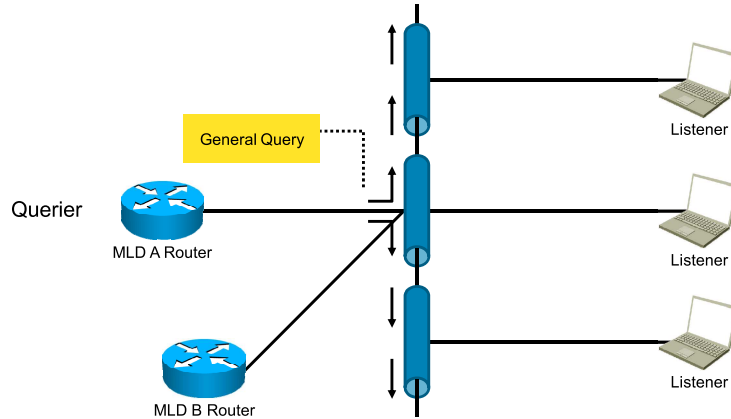
© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-11

When a router in Querier state receives a multicast listener done message from a link, and the multicast address that is identified in the message is present in the querier list of addresses that have listeners on that link, then the querier sends n multicast address specific queries to that multicast address (where n equals the Last Listener Query Count). One query is sent per interval, based on the Last Listener Query Interval, which also determines the Maximum Response Delay for the messages. If no multicast listener report messages for the address are received from the link after the response delay of the last query has passed, then the routers on the link assume that the address no longer has any listeners on that link. The address is therefore deleted from the list and its disappearance is made known to the multicast routing component. This process continues to its resolution (that is, until a report is received or the last multicast address specific query is sent with no response), despite any transition from querier to non-querier on the link.

General MLDv1 Process: Querier and Non-Querier Functions

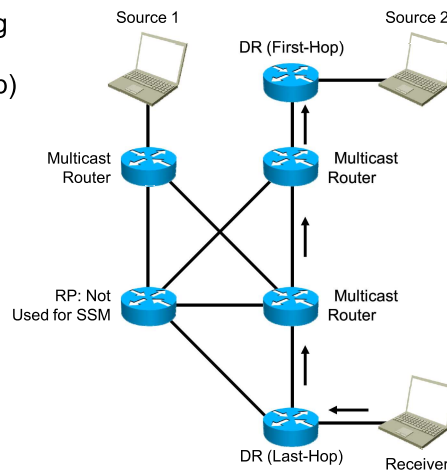
- When multiple multicast routers are present on a link, only one is elected a querier.
- There is only one querier per link.



With respect to each of its attached links, a router can assume one of two roles: querier or non-querier. There is normally only one querier per link. All routers start up as a querier on each of their attached links. If a router hears a query message whose IPv6 source address is numerically less than its own selected address for that link, then the router must become a non-querier on that link. If the Other Querier Present Interval passes without the router receiving, from a particular attached link, any queries from another router that has an address less than its own, then the first router resumes the role of querier on that link. A querier for a link periodically (according to the Query Interval) sends a general query on that link, to solicit reports of all multicast addresses of interest that are on the link. On startup, a router should send n general queries (where n is the Startup Query Count), spaced closely together (according to the Startup Query Interval), on all attached links, to quickly and reliably discover the presence of multicast listeners on those links.

Multicast Listener Discovery: MLDv2

- Support for source filtering (a node can select a source for a specific group)
- Interoperable with MLDv1
- Analogous functionality to IGMPv3 for IPv4



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—5-13

An IPv6 router uses MLD to discover the presence of multicast listeners on directly attached links and to discover which multicast addresses are of interest to those neighboring nodes. MLDv2 is designed to be interoperable with MLDv1. MLDv2 adds the ability for a node to report interest in listening to packets with a particular multicast address only when those packets are from specific source addresses or are from all sources except for specific source addresses.

MLDv2 is a translation of the IGMPv3 protocol (RFC 3376) for IPv6 semantics.

A node maintains or computes multicast listening state for each of its interfaces. Conceptually, that state consists of a set of records, each of which contains an IPv6 multicast address, a filter mode, and a source list. The filter mode may be either INCLUDE or EXCLUDE. In INCLUDE mode, the reception of packets that are sent to the specified multicast address is enabled only from the source addresses that is listed in the source list. In EXCLUDE mode, the reception of packets that are sent to the given multicast address is enabled from all source addresses except those that are listed in the source list.

For example, with MLDv1, a router must track multicast destinations for which there are link-local listeners. A multi-interface router needs to track these destinations for each interface and build a consolidated set of destination addresses that to send to the router. With MLDv2, a router must track, for each link, the list of multicast destination addresses and the list of associated source addresses for those multicast destinations and must keep a consolidated list of those streams. This additional task makes it possible to fully specify, as a multicast receiver, a very specific multicast stream to listen for. Even if two multicast senders use the same multicast destination address (not a reserved but a self-selected temporary address), a sender can be sure to listen to the desired stream.

In the previous example, the receiver indicates to the DR, via MLDv2, an interest in a given multicast group. The MLDv2 report includes, for SSM, the source addresses for the group. The router uses PIM to build forwarding state directly back to the source; no rendezvous point (RP) is involved. In this way, MLDv2 enables SSM. If no source is specified, then multicast state is built back to the RP, as in any source multicasting.

MLDv2 Messages

- General query
- Multicast address-specific query
- Multicast address and source-specific query
- Current state report
- State change report

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-14

There are three types of MLDv2 query messages:

- General queries
- Multicast address specific queries
- Multicast address and source specific queries

The querier periodically sends general queries to learn multicast address listener information from an attached link. These queries are used to build and refresh the Multicast Address Listener state inside all multicast routers on the link.

Nodes respond to these queries by reporting their per-interface Multicast Address Listening state, through current state report messages that are sent to a specific multicast address to which all MLDv2 routers on the link listen. On the other hand, if the listening state of a node changes, the node immediately reports these changes through a state change report message. The state change report contains either Filter Mode Change records, Source List Change records, or both types of record.

MLDv2 Additional Features and Changes

- Source filtering
- Filter mode and source list on the IP Service interface
- Per-socket and per-interface multicast listening states on MLDv2 nodes
- Filter mode, list of sources and source timers on MLDv2 routers
- Additional fields in queries

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—5-15

These are some of the additional features and changes with MLDv2:

- MLDv2 introduces source filtering, like IGMPv3 does.
- The IP Service interface of MLDv2 nodes is modified accordingly. This interface enables the specification of a filter mode and a source list.
- An MLDv2 node keeps per-socket and per-interface multicast listening states that include a filter mode and a source list for each multicast address. These states enable packet filtering that is based on the multicast reception state of a socket.
- The MLDv2 state that is kept on routers includes a filter mode and a list of sources and source timers for each multicast address that has listeners on the link. MLDv1 routers keep only the list of multicast addresses.
- Queries include additional fields:
 - Maximum Response Code, S flag (Suppress Router-Side Processing), Querier's Robustness Variable, Querier's Query Interval Code, and Number of Sources

Note For detailed information, refer to RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*.

MLDv2 Additional Features and Changes (Cont.)

- S flag included in queries.
- Robustness variable and query interval code included in queries.
- Multicast address and source-specific query as a new query type.
- Maximum response delay not included in the query.
- Reports include multicast-address records.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6 v3.0—5-16

- The S flag (Suppress Router-Side Processing) is included in queries to fix robustness issues.
- Querier's Robustness Variable and Querier's Query Interval Code are included in queries to synchronize all MLDv2 routers that are connected to the same link.
- The multicast address and source specific query is a new query type.
- The Maximum Response Delay is no longer directly included in the query. Instead, an exponential algorithm is used to calculate its value, based on the Maximum Response Code that is included in the query.
- Reports include multicast address records. Information on the listening state for several multicast addresses can be included in the same report message. MLDv2 also distinguishes between types of reports. MLDv1 has a single report message that listening nodes send to MLD routers. MLDv2 breaks these reports into two types:
 - **Current state reports:** These reports are sent in response to queries from MLD routers and often refresh the state tables that are kept on the routers. These reports do not cause any state change; they only refresh it.
 - **State change reports:** The listening node sends these reports whenever it wants to modify the list of multicast destination addresses to which it listens, desired sources for those streams, or the INCLUDE or EXCLUDE policy for those streams. These reports replace MLDv1 done reports and the start-listen reports.

MLDv2 Additional Features and Changes (Cont.)

- Sending reports to the “all MLDv2-capable multicast routers” address
- No host suppression
- Unsolicited reports sending changes
- No multicast listener done messages
- Interoperability with MLDv1
- Interoperability assurance

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0--5-17

- Reports are sent to the “all MLDv2-capable multicast routers” address, instead of to the multicast address to which the host listens, as in MLDv1. This process facilitates the operation of Layer 2 snooping switches. The new reserved “all MLDv2-capable multicast routers” multicast address is FF02::16.
- There is no host suppression as in MLDv1. All nodes send report messages for these reasons:
 - Routers might want to track (as a proprietary feature), which specific hosts are listening to which specific addresses. This process could be undertaken for several reasons: The Internet Engineering Task Force (IETF) suggests possible advantages for congestion management by the routers or to aid accounting practices. Cisco refers to this process as explicit tracking.
 - Host suppression simplifies host implementations because hosts no longer need to listen for peer reports and decide whether to respond to MLD queries.
- Unsolicited reports, which announce changes in receiver listening state, are sent according to the specified Robustness Variable.
- There are no done messages.
- Interoperability with MLDv1 systems is achieved by MLDv2 state operations.
- To ensure interoperability, hosts maintain a Host Compatibility Mode variable and an Older Version Querier Present timer, per interface. Routers maintain a Multicast Address Compatibility Mode variable and an Older Version Host Present timer, per multicast address.

MLD Snooping and MLD Group Limits

This topic describes the MLD snooping feature that switches use.

MLD Snooping

- Default behavior for multicast switched traffic is to flood it to all Layer 2 interfaces (in the same VLAN).
- High volume of multicast traffic loads the switch data plane and control plane.
- Snooping of MLD packets is used to determine which switch ports receive data for multicast groups.
- Layer 2 ports that do not have any listeners do not transmit multicast traffic.
- If a multicast group has only sources and no receivers in a VLAN, MLD snooping constrains the multicast traffic to only the multicast router ports.
- MLD snooping monitors Layer 3 MLD traffic.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6 v3.0—5-19

In IPv4, Layer 2 switches can use IGMP snooping to limit the flooding of multicast traffic, by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces that are associated with IP multicast devices. In IPv6, MLD snooping performs a similar function. With MLD snooping, IPv6 multicast data is forwarded selectively to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN. This list is constructed by snooping IPv6 multicast control packets.

MLD Normal Behavior

MLD, which runs at Layer 3 on a multicast router, generates Layer 3 MLD queries in subnets on which the multicast traffic needs to be routed.

You can configure the MLD snooping querier on the switch to support MLD snooping in subnets that do not have any multicast router interfaces.

MLD (on a multicast router) or the MLD snooping querier (locally) sends out periodic general MLD queries, which the switch forwards through all ports in the VLAN and to which hosts respond. MLD snooping monitors the Layer 3 MLD traffic.

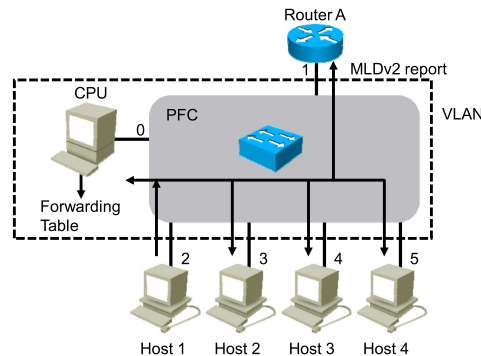
MLD Snooping

You can configure the switch to use MLD snooping in subnets that receive MLD queries from either MLD or the MLD snooping querier. MLD snooping constrains IPv6 multicast traffic at Layer 2 by dynamically configuring Layer 2 LAN ports to forward IPv6 multicast traffic to only those ports that want to receive it.

Note MLD is derived from IGMP. MLDv1 is equivalent to IGMPv2, and MLDv2 is equivalent to IGMPv3. MLD is a subprotocol of ICMPv6, and MLD messages are a subset of ICMPv6 messages and are identified in IPv6 packets by a preceding Next Header value of 58.

MLD Snooping on a Switch

- MLD snooping is available on switches only.
- MLD snooping is available for MLDv1 and MLDv2, depending on switch.
- MLD Snooping Proxy Reporting: First report to the router interface is sent, all other reports for the same group are suppressed.



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—5-20

The switch supports two versions of MLD snooping:

- MLDv1 snooping detects MLDv1 control packets and sets up traffic bridging, based on IPv6 destination multicast addresses.
- MLDv2 basic snooping uses MLDv2 control packets to set up traffic forwarding, based on IPv6 destination multicast addresses.

The switch can snoop on both MLDv1 and MLDv2 protocol packets and bridge IPv6 multicast data, based on destination IPv6 multicast addresses.

Relevant MLD Messages

These MLD messages are relevant for MLD snooping operations:

- Multicast listener queries
 - **General query:** Sent by a multicast router, to learn which multicast addresses have listeners
 - **Multicast address specific query:** Sent by a multicast router, to learn whether a particular multicast address has any listeners
 - **Multicast address and source specific query:** Sent by a multicast router, to learn whether any of the sources from the specified list for the particular multicast address has any listeners
- Multicast listener reports
 - **Current state record (solicited):** Sent by a host, in response to a query to specify the INCLUDE or EXCLUDE mode for every multicast group in which the host is interested
 - **Filter mode change record (unsolicited):** Sent by a host, to change the INCLUDE or EXCLUDE mode of one or more multicast groups

- **Source list change record (unsolicited):** Sent by a host, to change information about multicast sources

Note	Because the Layer 2 table is based on the MAC group and the VLAN, it is preferable to have only one multicast source per MAC group for MLD hosts.
-------------	---

MLD Snooping Proxy Reporting

Because MLD does not have report suppression, all the hosts send their complete multicast group membership information to the multicast router, in response to queries. The switch snoops these responses, updates the database, and forwards the reports to the multicast router. To prevent the multicast router from becoming overloaded with reports, MLD snooping does proxy reporting.

Proxy reporting forwards to the router only the first report for a multicast group and suppresses all other reports for the same multicast group.

Proxy reporting processes solicited and unsolicited reports. Proxy reporting is enabled and cannot be disabled.

MLD Group Join

- Hosts join a multicast group:
 - By using an unsolicited report
 - By replying to a route MLD general query
- Switch creates a Layer 2 FIB entry for the multicast group in that VLAN.
- When additional hosts want to join the group, the switch adds their port to the Layer 2 FIB for the group.
- Only one entry in Layer 2 FIB per multicast group and per VLAN where the report was seen.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—5-21

Hosts join IPv6 multicast groups either by sending an unsolicited MLD report or by sending an MLD report in response to a general query from an IPv6 multicast router. (The switch forwards general queries from IPv6 multicast routers to all ports in a VLAN.) The switch snoops these reports.

In response to a snooped MLD report, the switch creates an entry in its Layer 2 forwarding information base (FIB) table for the VLAN on which the report was received. When other hosts that are interested in this multicast traffic send MLD reports, the switch snoops their reports and adds them to the existing Layer 2 FIB table entry. The switch creates only one entry per VLAN in the Layer 2 FIB table for each multicast group for which the switch snoops an MLD report.

MLD snooping suppresses all but one of the host reports per multicast group and forwards this one report to the IPv6 multicast router. The multicast router begins sending the multicast group data to the switch.

The switch hardware can distinguish MLD information packets from other packets for the multicast group. Only MLD packets should be sent to the CPU, which prevents the switch from becoming overloaded with multicast frames. Frames that are addressed to the multicast MAC address (which is derived from the group address) and that are not MLD packets should be sent to the multicast router and to the host that has joined the group.

If another host sends an unsolicited MLD report for the same group, the switch snoops that message and adds the port number of this host to the FIB table.

Note Because the FIB table directs MLD messages only to the switch, the message is not flooded to other ports. Any known multicast traffic is forwarded to the group and not to the switch.

Configuring MLD Snooping

MLD snooping configuration is per VLAN interface.

```
Switch(config)#
```

```
interface vlan vlan id
```

- Enters the VLAN interface configuration mode

```
Switch(config-if)#
```

```
ipv6 mld snooping
```

- Enables MLD snooping

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-22

MLD snooping is activated by using the **ipv6 mld snooping** interface configuration command.

Verifying MLD Snooping

```
Router#show ipv6 mld interface vlan 25 | include snooping
MLD snooping is globally enabled
MLD snooping is enabled on this interface
MLD snooping fast-leave is enabled and querier is enabled
MLD snooping explicit-tracking is enabled
MLD snooping last member query response interval is 1000 ms
MLD snooping report-suppression is disabled
Router#
```

```
Router#show mac-address-table multicast vlan 1
vlan  mac address      type    qos      ports
-----+-----+-----+-----+-----
1  0100.5e02.0203  static --  Gi1/1,Gi2/1,Fa3/48,Router
1  0100.5e00.0127  static --  Gi1/1,Gi2/1,Fa3/48,Router
1  0100.5e00.0128  static --  Gi1/1,Gi2/1,Fa3/48,Router
1  0100.5e00.0001  static --  Gi1/1,Gi2/1,Fa3/48,Router,Switch
Router#
```

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-23

The **show ipv6 mld interface** and **show mac-address-table** commands can be used to monitor MLD snooping.

MLD Group Limits

- Access control for multicast group receivers.
- Limits the list of groups that a receiver can join.
- Group limits can be implemented:
 - Globally
 - Per interface
- Membership reports that exceed the limit are ignored.

Switch (config)# or Switch (config-if)#

```
ipv6 mld state-limit number
```

- Limits the number of MLD states globally or per interface.

Switch (config-if)#

```
ipv6 mld access-group acl name
```

- Allows the user to perform IPv6 multicast receiver access control.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-24

Implementing MLD Group Limits

Per-interface and global MLD limits operate independently of each other. Both per-interface and global MLD limits can be configured on the same router. The number of MLD limits, globally or per interface, is not configured by default; the user must configure the limits. A membership report that exceeds either the per-interface or the global state limit is ignored.

MLD Access Group

The MLD access group provides receiver access control in Cisco IOS IPv6 multicast routers. This feature limits the list of groups that a receiver can join, and it allows or denies sources that are used to join SSM channels.

Note To learn more about customizing MLD on a router interface, refer to the section “Customizing and Verifying MLD on an Interface” in *Implementing IPv6 Multicast* at <http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-multicast.html#wp1056822>.

Multicast User Authentication and Group Range Support

This topic describes Multiprotocol Border Gateway Protocol (MP-BGP) for the IPv6 multicast address family.

Multicast User Authentication Support

- Available on multicast routers.
- IPv6 multicast, by design, allows any node to become a receiver or a source for a multicast group.
- Access control can be implemented for:
 - Resource access control and accounting
 - Receiver access control and accounting
- Standard AAA mechanism is used at the last-hop router.
- User profiles are created on the AAA server and are downloaded to the router.
- MLD join request on the access router triggers the AAA process and profile download.
- Accounting can be used to track resource consumption.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5.26

By design, IPv6 multicast allows any host in the network to become a receiver or a source for a multicast group. Therefore, multicast access control is needed to control multicast traffic in the network. Access control functionality consists mainly of source access control and accounting, receiver access control and accounting, and provisioning of this access control mechanism.

Multicast access control provides an interface between multicast and authentication, authorization, and accounting (AAA) for provisioning, authorizing, and accounting at the last-hop router, receiver access control functions in multicast, and group or channel disabling capability in multicast.

When you deploy a new multicast service environment, you must add user authentication and provide a user profile download on a per-interface basis. The use of AAA and IPv6 multicast supports user authentication and downloading of the user profile in a multicast environment.

The arrival of an MLD join on the access router is the event that triggers the download of a multicast cast access-control profile from the RADIUS server to the access router. When this event occurs, a user can cause the authorization cache to time out and request download periodically or can use an appropriate multicast **clear** command to trigger a new download if profile changes have occurred.

Accounting occurs via RADIUS accounting. Start and stop accounting records are sent to the RADIUS server from the access router. To help you track resource consumption on a per-stream basis, these accounting records provide information about the multicast source and group. The start record is sent when the last-hop router receives a new MLD report, and the stop record is sent at MLD leave or if the group or channel is deleted.

Multicast Group Range

- Specifies the multicast groups that the router receives, unless:

- From authorized channels
- From authenticated groups

Router(config)#

```
ipv6 multicast group-range acl name
```

- Disables multicast protocol actions and traffic forwarding for unauthorized groups or channels on all the interfaces in a router

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-27

Disabling the Router from Receiving Unauthenticated Multicast Traffic

In some situations, access control might be needed to prevent multicast traffic from being received unless the subscriber is authenticated and the channels are authorized, per access control profiles. That is, there should be no traffic at all unless specified otherwise by access control profiles.

The following task shows how to disable the router from receiving multicast traffic from unauthenticated groups or unauthorized channels.

Configuring Multicast Authentication

- Enable AAA access control for IPv6 multicast.
- Specify AAA method list and enable multicast accounting.
- Disable the router from receiving unauthenticated multicast traffic.
- Reset authorization status on an MLD interface.

```
aaa new-model
aaa authorization multicast default
aaa accounting multicast default

interface fastethernet 0/1
ipv6 multicast aaa account receive list1

ipv6 multicast group-range allowed-mcast-acl
clear ipv6 multicast aaa authorization fastethernet 0/1
```

- Authentication is supported only per port, interface, VC, or VLAN, not per user or hostname or by using password.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—5-28

Multicast authentication is configured by using the standard AAA framework. On the high level, the process consists of the following steps:

- Step 1** Enable AAA access control for IPv6 multicast.
- Step 2** Specify the AAA method list and enable multicast accounting.
- Step 3** Disable the router from receiving unauthenticated multicast traffic.
- Step 4** Reset the authorization status on an MLD interface.

The **aaa accounting multicast default** command enables AAA authorization and sets parameters that restrict user access to an IPv6 multicast network, using the default method.

The **ipv6 multicast aaa account receive** command enables AAA accounting on groups or channels that are specified in **list1**.

The **ipv6 multicast group-range** command disables multicast protocol actions and traffic forwarding for unauthorized groups or channels on all the interfaces in a router.

Note The port, interface, virtual circuit (VC), or VLAN ID is the user or subscriber identity. User identity by hostname, user ID, or password is not supported.

Summary

This topic summarizes the key points that were discussed in this lesson.

Summary

- MLD handles the join and leave process on the LAN segment between the listener and the router.
- MLDv1 and MLDv2 have analogous functionality to IGMPv2 and IGMPv3 in IPv4.
- MLD snooping allows you to optimize the use of resources (both control plane and data plane) on a switch so that only the ports with multicast receivers receive multicast traffic.
- MLD group limits limit the number of groups that a receiver can join.
- Multicast user authentication implements access control for the groups that a user can join.
- Multicast group ranges specify a list of allowed multicast groups and sources for which the router will forward traffic.
- Multicast authentication uses AAA as the underlying mechanism.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--6-29

References

To learn more, browse through the following resources:

- Cisco Systems, Inc. *Configuring MLD Snooping for IPv6 Multicast Traffic*.
<http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/snoopmld.html#wp1074543>.
- Cisco Systems, Inc. *Implementing IPv6 Multicast*.
<http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-multicast.html#wp1057003>.
- Cisco Systems, Inc. *Implementing IPv6 Multicast*.
http://www.cisco.com/en/US/docs/ios/ios_xe/ipv6/configuration/guide/ip6-multicast_xe.html.
- IETF. *RFC 2710: Multicast Listener Discovery (MLD) for IPv6*. October 1999.
<http://tools.ietf.org/html/rfc2710>.
- IETF. *RFC 3810: Multicast Listener Discovery Version 2 (MLDv2) for IPv6*. June 2004.
<http://tools.ietf.org/html/rfc3810>.

Module Summary

This topic summarizes the key points that were discussed in this module.

Module Summary

- The principles of IPv6 multicast remain the same as with IPv4. With the shift to IPv6 addressing formats, the change in IPv6 multicast is in the core of IPv6 multicast addressing.
- MLD is an asymmetric protocol that specifies different behaviors for multicast receivers and routers.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0--5-1

Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

- Q1) Which of these options is the most accurate representation of multicast scope assignments for IPv6? (Source: “Implementing Multicast in an IPv6 Network”)
- A) interface-local, organization-local, regional
 - B) interface-local, site-local, site-local
 - C) site-local, organization-local, campus-local
 - D) global, interface-local, enterprise-local
- Q2) MLD is used between which network components in a multicast deployment? (Source: “Implementing Multicast in an IPv6 Network”)
- A) link-local listeners and the last-hop DR
 - B) link-local routers and their upstream multicast router
 - C) last-hop DR and the link-local multicast source
 - D) core multicast routers
- Q3) IPv6 supports embedded RP, in which the unicast address of the RP is specified within the multicast group address. In the example FF7E:540:2001:DB8:2:3::105, what does the “40” in “540” signify? (Source: “Implementing Multicast in an IPv6 Network”)
- A) the RP is at host number “::40”
 - B) the multicast is organization-scope
 - C) the embedded prefix is a /40
 - D) the embedded prefix is a /64

Module Self-Check Answer Key

Q1) B
Q2) A
Q3) D

IPv6 Transition Mechanisms

Overview

Moving from IP version 4 (IPv4) to IP version 6 (IPv6) is challenging because of the infrastructure that has been developed, both hardware and software, which supports only IPv4. To make it cost effective and technically feasible for organizations to upgrade their networks to IPv6, the Internet Engineering Task Force (IETF) developed a series of transition mechanisms for IPv6 adoption. This module describes IPv6 transition mechanisms and which methods will be most effective in your network.

Module Objectives

Upon completing this module, you will be able to evaluate the scenario and desired outcome and identify the best transition mechanism for the situation. This ability includes being able to meet these objectives:

- Explain the dual-stack transition mechanism
- Identify the different encapsulation mechanisms that are used for IPv6 transition
- Describe the NAT-PT transition mechanism and why it is a method of last resort

Implementing Dual-Stack

Overview

When integrating IP version 6 (IPv6) into an existing network, you are required to make choices about the method or methods that are used to support the integration. Numerous approaches are available. This lesson describes the dual-stacking approach to integrating IPv6 functionality into an existing IP version 4 (IPv4)-only environment. Dual-stack is the most common IPv6 transition approach.

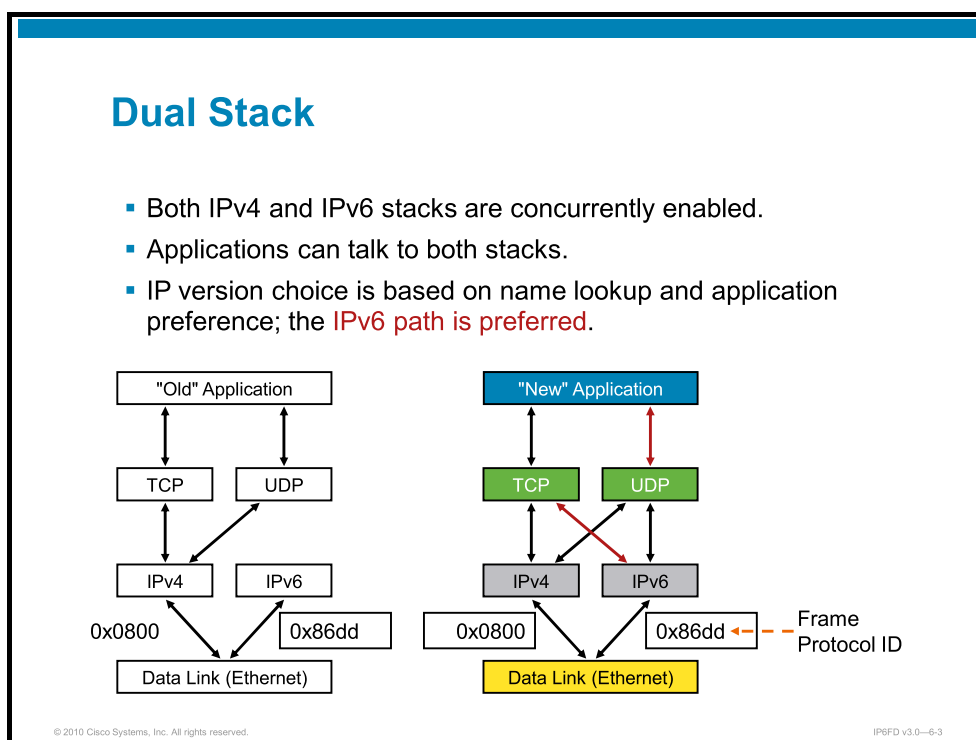
Objectives

Upon completing this lesson, you will be able to describe the dual-stack approach to integration. This ability includes being able to meet these objectives:

- Describe the dual-stack applications
- Describe the dual-stack approach and Cisco configuration
- Identify some of the benefits and issues that are associated with dual-stacking

Dual-Stack Applications

This topic describes the dual-stack applications.



The preferred strategy for transitioning to IPv6 is the dual-stack approach, in which nodes have both IPv4- and IPv6-enabled stacks; some applications must be modified to use IPv6. “Old” IPv4-only applications continue to work as before. New and modified applications take advantage of both IP layers, usually preferring to use the IPv6 protocol for communication.

A new application programming interface (API) is defined to support both IPv4 and IPv6 addresses and Domain Name System (DNS) requests. This new API replaces the **gethostbyname** and **gethostbyaddr** calls. A converted application can use both IPv4 and IPv6. An application can be converted to the new API while still using IPv4. (Bear in mind that the application developers are responsible for updating the application.)

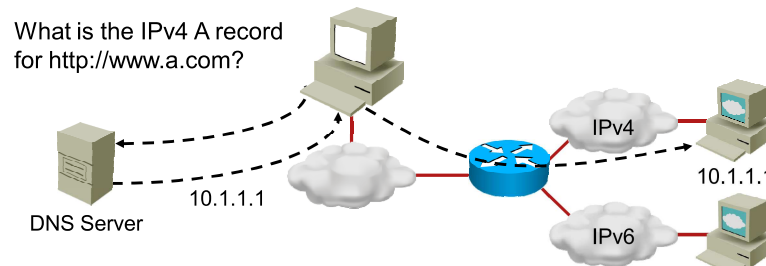
Past experience in porting IPv4 applications to IPv6 shows that, for most applications, the transition makes a minimal change in some localized places inside the source code.

The dual-stack technique is well known and has been applied in the past for other protocol transitions. It enables one-by-one application upgrades to IPv6.

Dual-Stack Operation

An application that is not aware of IPv6 or that is forcing the use of IPv4:

- It asks the DNS for IPv4 address.
- It connects to the IPv4 address.
- Legacy applications; new applications should be designed as protocol-independent.



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0-6-4

The figure shows the typical situation without IPv6.

An application that is not aware of IPv6, or is forcing the use of IPv4, requests only the IPv4 address of the destination hostname (http://www.a.com) from the DNS. The DNS request for an IPv4 address is a request for an A record. The DNS server sends back only the IPv4 address of the host. The application then requests the source host to connect to the destination host using IPv4.

Tip

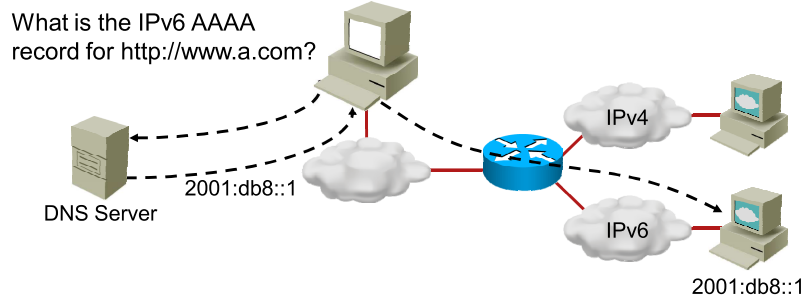
In the figure, the DNS queries from the host to the DNS caching server can be carried over IPv4 or IPv6 transport. The content of the queries (A, quad-A record [AAAA], pointer record [PTR], and others) are not related to the IP transit protocol used to carry the DNS (User Datagram Protocol [UDP] or TCP-based) queries.

Dual-Stack Operation (Cont.)

IPv6-Only

An application that is only IPv6-enabled has to use IPv6 because it is the only stack or is forcing the use of IPv6:

- It asks the DNS for IPv6 address.
- It connects to the IPv6 address.



The figure shows the typical situation for IPv6-only or IPv6-preferred services.

The application requests only the IPv6 address of the destination hostname (http://www.a.com) from the DNS when an application is one of the following:

- The application is only IPv6-enabled.
- IPv6 is the only stack available on the host.
- The application is forcing the use of IPv6.

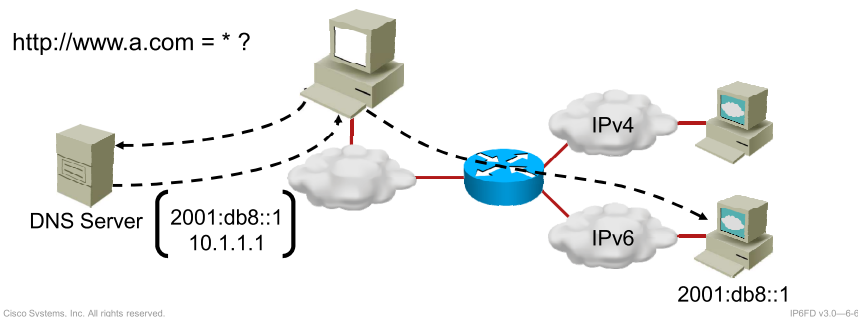
The DNS request for an IPv6 address is a request for an AAAA record. The DNS server sends back only the IPv6 address of the host. The application then requests the source host to connect to the destination host using IPv6.

Dual-Stack Operation (Cont.)

Dual Stack

In a dual-stack case, an IPv4-enabled and IPv6-enabled application asks the DNS for all types of addresses:

- It chooses one address.
- It connects, for example, to the IPv6 address.
- Some content providers require you to register to receive AAAA DNS responses.



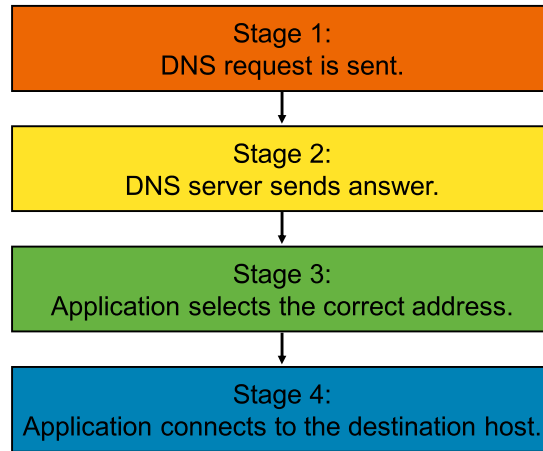
The figure shows the typical situation for dual-stack hosts with applications supporting both protocols.

An IPv6-enabled and IPv4-enabled application sends requests to the DNS for all types of addresses for the destination hostname (`http://www.a.com`). The DNS server sends all available host addresses (IPv4 and IPv6). The application chooses one or the other (the typical default is IPv6) and then requests the source host to connect to the destination host using IPv6.

Tip RFC 3484, *Default Address Selection for Internet Protocol version 6 (IPv6)*, describes how nodes choose a destination address from a set of IPv4, and possibly IPv6, destination addresses and how nodes choose the source address to use for a connection. The rules in this RFC prefer IPv6 over IPv4, when both are available, and suggest that nodes should be able to change this default behavior.

Since it is possible that the IPv6 stack can be misconfigured, some content providers (for example, Google) currently require that you register your domain to receive DNS AAAA records in answers to queries. This requirement serves to prevent scenarios in which applications acquire IPv6 addresses, but an inoperative stack could result in no connectivity. In this case, it is better to use the IPv4 stack. In environments with working IPv6 implementations, the endpoints should receive AAAA answers and prefer the IPv6 stack for communication as intended.

Dual-Stack Address Host Selection Process



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—6-7

Dual-stacking manages most address selection and DNS resolution issues. The process is as follows:

- **Stage 1: DNS request is sent.** An IPv6-enabled and IPv4-enabled application sends requests to the DNS for all types of addresses for the destination hostname (for example, <http://www.example.com>).
- **Stage 2: DNS server sends an answer.** The DNS server sends the answer back with all available host addresses (IPv6 and IPv4).
- **Stage 3: Application selects the correct address.** The application chooses one or the other (the typical default is IPv6).
- **Stage 4: Application connects to the destination host.** The application requests that the source host connect to the destination host using IPv6.

Dual-Stack Node

This topic describes the dual-stack approach and Cisco configuration.

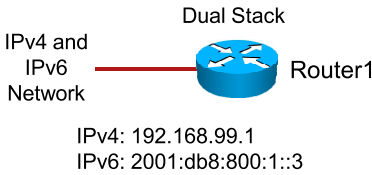
Cisco IOS Dual Stack

Cisco IOS Software supports IPv6:

- When IPv4 and IPv6 addresses are configured on an interface, the interface is dual-stacked.

IPv4 and IPv6 Network

Dual Stack



Router1

IPv4: 192.168.99.1
IPv6: 2001:db8:800:1::3

```
ipv6 unicast-routing
!
interface Ethernet0
 ip address 192.168.99.1 255.255.255.0
 ipv6 address 2001:db8:c18:1::3/64
```

or

```
ipv6 unicast-routing
!
interface Ethernet0
 ip address 192.168.99.1 255.255.255.0
 ipv6 address 2001:db8:c18:1::/64 eui-64
```

Note: IPv6 addresses can be acquired by autoconfiguration as well, using the EUI-64 suffix.

© 2010 Cisco Systems, Inc. All rights reserved.IPv6FD v3.0-6-9

Cisco IOS Software is IPv6-ready. As soon as both IPv4 and IPv6 basic configurations are completed on an interface, the interface is dual-stacked and forwards both IPv4 and IPv6 traffic.

Additionally, an interface IPv6 address can be configured using a fixed assignment or using autoconfiguration, in which the last 64 bits of the IPv6 address are derived from the MAC address (the **eui-64** keyword; see the lower example in the figure).

Steps to Implement Dual Stack in an IP Network

There are basic steps required to implement a dual-stack approach:

1. Review the network, application, and security aspects of your environment.
2. Upgrade, if needed, the hosts, routers, and infrastructure services to support IPv6.
3. Enable IPv6 support.
4. Upgrade all services where possible to provide IPv6 functionality.
5. Ensure that dual-stack operation is functioning correctly.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--6-10

Dual stack is the process of introducing both IPv4 and IPv6 in the same environment on the same interface.

Follow this procedure to implement a dual stack in an IP network:

- Step 1** Review the network, application, and security aspects of your environment. Proper planning is absolutely critical to the successful deployment of IPv6-enabled services and must be as inclusive as possible. Network engineers, application designers, and security staff are essential to this process. Management support is important as well.
- Step 2** Upgrade hosts, routers, and infrastructure services to support IPv6. You may need to upgrade network routers and end nodes to dual-stack-capable operating system levels. The IPv4 support does not go away; rather, it continues in parallel with IPv6. IPv4 communication uses the IPv4 protocol stack with forwarding of IPv4 packets, which are based on routes that are learned through running IPv4-specific routing protocols. IPv6 communication uses the IPv6 stack with routes that are learned through IPv6-specific routing protocols. Some routers may require upgrades to support the additional overhead of dual-stack operation. Special attention should be paid to key infrastructure services such as DNS, HTTP, and Simple Mail Transfer Protocol (SMTP).
- Step 3** Enable IPv6 support. Most router and host vendors ship an IPv6 stack along with the IPv4 stack. Once all nodes support dual-stack operation, activating it should simply require enabling the IPv6 stack.
- Step 4** Upgrade all services, where possible, to provide IPv6 functionality. All network services should also be made available via IPv6. Certain network services, such as DNS and HTTP, are already IPv6-enabled but may require some additional attention to use IPv6 as the preferred transport.
- Step 5** Ensure that dual-stack operation is functioning correctly. Once you have enabled IPv6 on the network, it is important to verify that all systems are still functioning correctly. Review network policies for items such as security to address IPv6 enablement on the networks. Additionally, review network capacity estimates, because a dual-stack configuration will use more system resources than single protocol operations.

The Dual-Stack Approach

This topic describes the benefits and issues that are associated with using dual-stack as the primary approach to IPv6 transition.

Using the Dual-Stack Approach

- Advantages of using dual-stacking as the primary approach to IPv6 integration include the following:
 - Relatively simple approach
 - Can be less costly in the short term
 - Continues to leverage existing IPv4 infrastructure
 - Allows indefinite coexistence between IPv4 and IPv6
- Issues when using dual-stacking as the primary approach to IPv6 integration include the following:
 - Can be more costly over the long term
 - Increases network complexity

© 2010 Cisco Systems, Inc. All rights reserved.IPv6FD v3.0—6-12

In most cases, IPv6 integration requires dual-stacking a node. The likelihood that all the applications and services in a network will be IPv6-only is very small (at least for the foreseeable future). As a result, access to IPv4 connectivity will remain a requirement. However, every IPv6 integration scenario is different, requiring careful evaluation of the transition mechanisms available, including a review of the benefits and challenges of using any particular mechanism.

Dual-stacking is a relatively easy approach to incorporating IPv6 into a network. Most modern operating systems support IPv6, so the majority of nodes (hosts, servers, or routers) probably support basic IPv6 enablement. This support results in a lower transition cost. Also, by dual-stacking, an organization can continue to leverage their existing IPv4-only applications until those applications have been made IPv6-capable or have been phased out in favor of applications that support IPv6.

However, as with any environment in which new technologies are introduced, the complexity of the environment will increase. It is not enough to simply IPv6-enable a desktop computer. Leveraging IPv6 requires additional infrastructure changes. These changes include modifications to services like DNS, DHCP, security policies, and more. Supporting two protocols over time will presumably be more costly than only one protocol.

Tip There is no standard way to transition a network to IPv6. The strategy depends on too many initial conditions, business plans, budget concerns, and so on. Each transition, or IPv6 integration into an existing network, is unique.

Dual-Stack Considerations

- Indefinite coexistence of IPv6 and IPv4 stacks in the network can be a drawback:
 - Performance and security issues
 - Higher complexity and costs
- Transitions are problematic:
 - Old DNS client can misinterpret the AAAA response.
 - Progress versus maintaining compatibility with legacy systems.
- Eventually, even if you use dual stack, you need end-to-end IPv6 connectivity, so you need to implement IPv6 end to end anyway.
- Network security should be on the same level for IPv4 and IPv6.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—6-13

Indefinite coexistence of a dual-stack implementation should not be a long-term goal and is not practical to implement. Connectivity itself is not problematic, but other aspects are:

- **Performance:** Network devices, such as switches, will still achieve higher performance when forwarding IPv4 packets in hardware. (This situation applies to hardware platforms, such as Cisco 7600 Series Routers, Cisco Catalyst 6500 Series Switches, and so on.)
- **Security:** Not all inspection engines are available for IPv6 protocols; security devices with early support had problems with failover, and so on.
- **Application services:** Devices, such as server load balancers that support IPv6, are difficult to find. This challenge is especially problematic because these devices make IPv6 content widely available to many clients. Without support on the content side, IPv6 availability cannot grow.

There is a turning point, until which it is feasible to maintain connectivity for legacy systems (that is, IPv4-only), and after that to care only for “new” systems (IPv6-only). This point will come in ten or more years, but until then it does make sense to consider legacy devices. For example, sending A and AAAA records in a DNS answer is correct, but an old endpoint might misinterpret the DNS answer and act unpredictably. This misinterpretation will result in no connectivity, in spite of it receiving a correct A response. To solve this issue in the interim, some DNS providers require that you register to receive both A and AAAA records.

Note Further information about DNS and IPv6 is covered in a separate section in this course.

Lastly, end-to-end IPv6 connectivity will be required—at least for IPv6-only content—so an IPv6 path will be needed anyway. There is little sense in maintaining an IPv4 stack when the majority of the content will be available on IPv6.

Summary

This topic summarizes the key points that were discussed in this lesson.

Summary

- Dual-stack networks allow applications the choice of running over IPv4 or IPv6 transport, because the nodes are configured to do either.
- Dual-stack is a relatively easy method for integrating IPv6 into an existing IPv4 environment, because there is wide support for IPv6 in most modern operating systems. It may, however, create a more complex network environment.
- Over the long-term, dual stack will cease to be feasible since the majority of the content will be available on IPv6.
- Security services, performance, and application services will become more evolved to discontinue dual-stack integration.

© 2010 Cisco Systems, Inc. All rights reserved. IPv6FD v3.0--6-14

Resources

For additional information, refer to these resources:

- *Implementing IPv6 Addressing and Basic Connectivity*, section *Dual IPv4 and IPv6 Protocol Stacks* at http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-addrg_bsc_con_ps6441_TSD_Products_Configuration_Guide_Chapter.html#wp1292298
- *Implementing IPv6 Addressing and Basic Connectivity*, section *Configuring an Interface to Support the IPv4 and IPv6 Protocol Stacks* at http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-addrg_bsc_con_ps6441_TSD_Products_Configuration_Guide_Chapter.html#wp1060846
- *My customers are not interested in IPv6 ... what can I do?* at <http://blog.ioshints.info/2010/02/my-customers-are-not-interested-in-ipv6.html>

Describing IPv6 Tunneling Mechanisms

Overview

When deploying IPv6, there will no doubt be a scenario where a part of the transit network does not support native IPv6. In such case the easiest solution is to somehow encapsulate IPv6 and send it over existing IPv4-enabled network. This lesson describes tunneling mechanisms for IP version 4 (IPv4)-to-IP version 6 (IPv6) transition, or for supporting IPv4 and IPv6 coexistence.

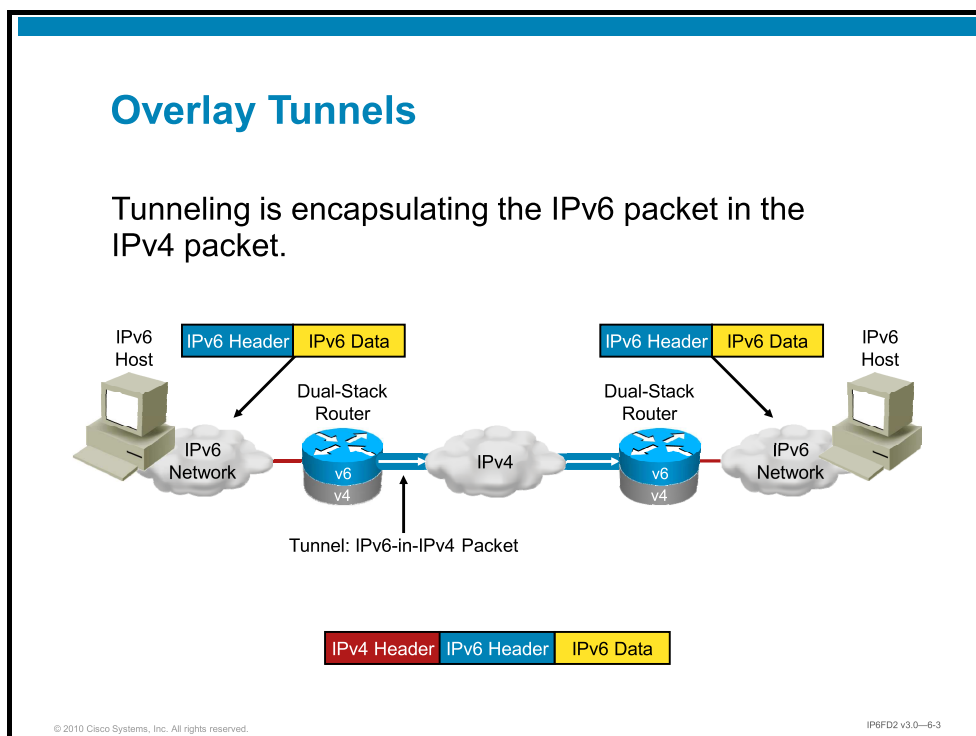
Objectives

Upon completing this lesson, you will be able to describe the dual-stack and automatic tunneling mechanisms. This ability includes being able to meet these objectives:

- Show the IPv6-in-IPv4 tunnel methodology and implementation
- Discuss the types of manually configured tunnels
- Describe how automatic tunnels work

Overlay Tunnels

This topic describes IPv6-in-IPv4 tunneling.



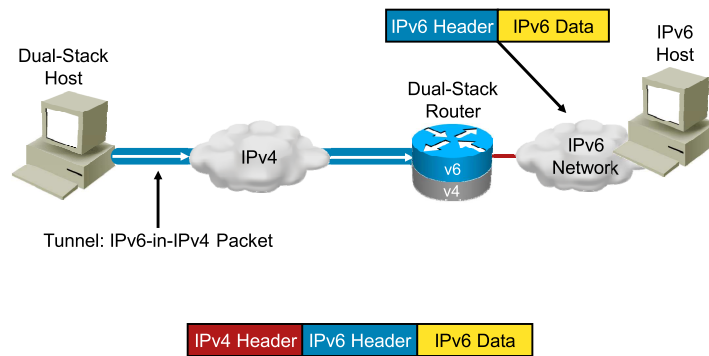
Tunnels are often used in networking to build a virtual network over an existing network. Typically, when IPv6 traffic is tunneled over an IPv4 network, one edge router encapsulates the IPv6 packet inside an IPv4 packet and then the other router decapsulates it, and vice versa. This enables the connection of IPv6 islands without the need to also convert the backbone to IPv6.

These are two of the primary issues of tunneling:

- The maximum transmission unit (MTU) is effectively decreased by 20 octets (if the IPv4 header does not contain any options data, which it normally would not).
- A tunneled network is often difficult to troubleshoot. Tunneling is a transition technique that should be used when it is appropriate and should not be considered as a final architecture. A native IPv6 architecture should be the target architecture.

Overlay Tunnels (Cont.)

Tunneling can be used by routers and hosts.



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD2 v3.0-6-4

Tunneling can be done between routers, between hosts or between a host and a router. This example shows an isolated dual-stack host using tunneling to connect to the edge router of the IPv6 network.

Note that if IP protocol 41 (IPv6 tunneling) is filtered anywhere along the path, any IPv6-in-IPv4 tunnel mechanism will not work. Furthermore, Network Address Translation (NAT) will break many of the tunneling mechanisms.

Note IP protocol 41 is the marker placed in the protocol field in an IPv4 packet header which notifies handling nodes that the packet payload is actually an IPv6 packet.

Tunnel Techniques Available for IPv6 Deployment

Many techniques are available for establishing a tunnel:

- Manually configured:
 - IPv6-in-IPv4
 - GRE
 - VPN
- Semiautomatic:
 - Tunnel broker
- Automatic:
 - 6to4
 - 6rd
 - ISATAP
 - Teredo

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD2 v3.0—6-5

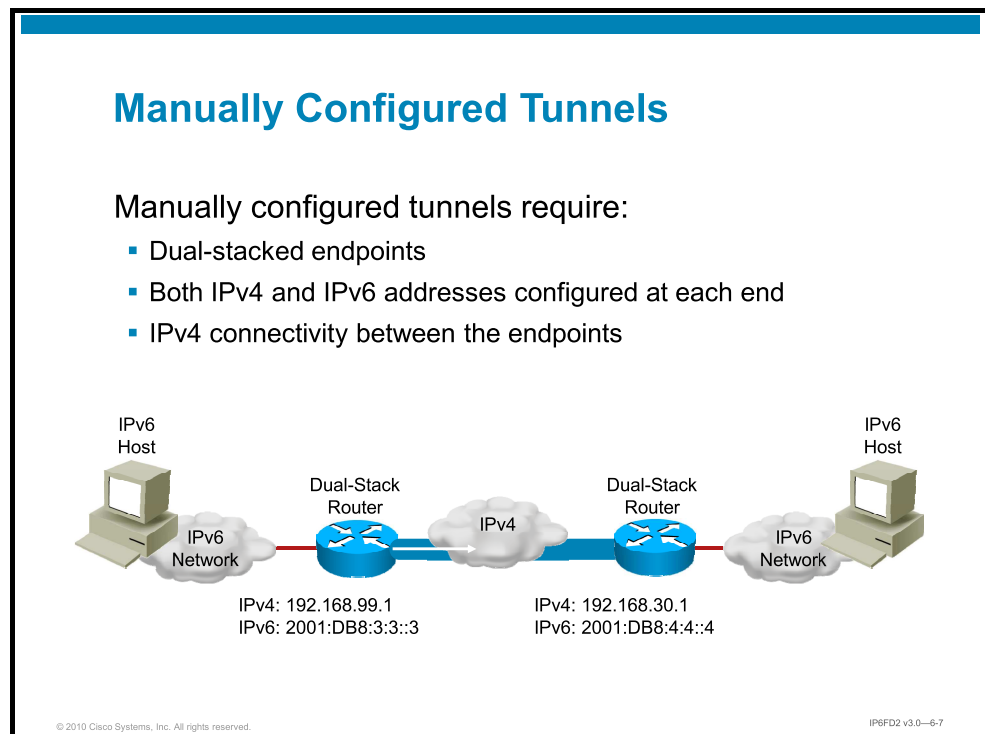
Many techniques have been designed to make tunnel establishment easier than manually configured tunnels. Some semiautomation can be achieved by the tunnel broker approach. Several automatic tunnel establishment techniques are also available, including 6to4, ISATAP and Teredo, and 6rd for service provider environments.

Tip Many tunnels do not include any type of security for inner or outer packets. For example, a protocol-41 tunnel does not imply any privacy for the tunneled IPv6 packets. The IPv6 packets inside the tunnel may be IPsec-protected, or the outer IPv4 packets could be protected with IPsec, or both (although "both" is probably not the best choice). For these tunnels to be considered Virtual Private Network (VPN) tunnels, normally some IPsec protection must be provided explicitly by the tunnel implementer.

Note **6rd** (Rapid Deployment) is another tunneling mechanism for transition to IPv6. It is used in service provider networks to convey IPv6 traffic across the SP access and core networks to the IPv6 Internet gateway. Encapsulation is done by the CPE equipment, and de-encapsulation by the gateway. 6rd is described in **Module 9: IPv6 and Service Providers**.

Manually Configured Tunnels

This topic describes a manually configured tunnel.



In this manually configured tunnel example, both the IPv4 and IPv6 addresses are configured statically on the routers at each end of the tunnel. These end routers must be dual-stacked and have IPv4 connectivity to each other. This configuration cannot change dynamically according to network needs and routing changes. Routing must also be set up properly to enable the forwarding of packets between the two IPv6 networks.

In a manually configured IPv6-in-IPv4 tunnel, IPv4 nodes are configured with a pseudo-interface that acts as a virtual tunnel interface. This tunnel interface sits “on top of” the IPv4 interface and must have an IPv6 link-local address (FE80::/64) that is created from the associated underlying IPv4 address as the lower 32 bits. This provides a locally unique interface identifier. In most scenarios, the tunnel pseudo-interface will also have a statically assigned and globally unique IPv6 unicast address.

IPv6 neighbor discovery mechanisms occur across the tunnel as normal. The tunnel acts as one hop for the IPv6 packet, whereas the IPv4 encapsulation packet may take many hops. Note that, for Cisco IOS software, router advertisements are not sent on tunnel interfaces by default—you must use the **no ipv6 nd suppress-ra** command to enable them.

Note that if IPv4 protocol 41 is filtered anywhere along the path, the tunnel will not work.

Manually Configured Tunnels (Cont.)

Configured tunnels connect IPv4/IPv6 dual-stack hosts or networks to larger IPv6 networks:

- Local network administrators arrange for a tunnel between IPv6 networks across IPv4-only networks.
- Configured tunnels are simple to deploy.
- Configured tunnels allow transport of IPv6 packets over an IPv4 network.
- Configured tunnels are available on most platforms.
- Configured tunnels do not scale well.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD2 v3.0—6-8

Manually configured tunnels have a few advantages over automatic tunnels. Most significantly, from security point of view, static tunnels are not activated without administrator's knowledge. A local administrator must explicitly configure a tunnel between two points.

Such tunnels are also simple to deploy, because one needs to set up just two endpoints, configure IP addresses to which the tunnel is attached and take care of routing.

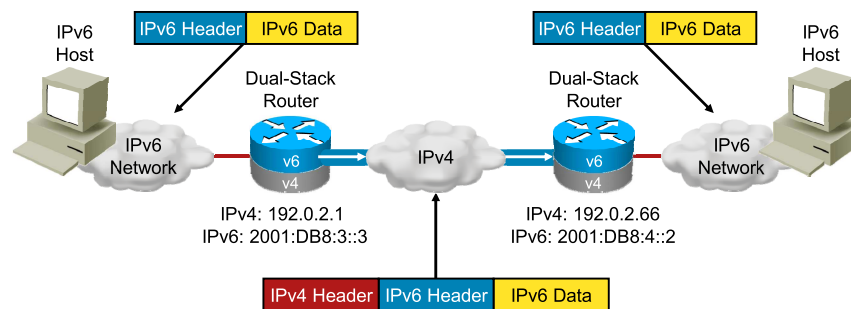
Simple static tunnel support is also available on most platforms, because extra processing is not required.

Configured tunnels also support multicast, making it easy to run IPv6 routing protocols over them.

The disadvantage of static tunnels is that they do not scale well. If a full mesh of tunnels is required, the number of tunnels will grow with the second power of number of nodes.

Manually Configured Tunnels: IPv6-in-IPv4 Tunnels

- Used between two points
- Require configuration of both tunnel source and destination addresses



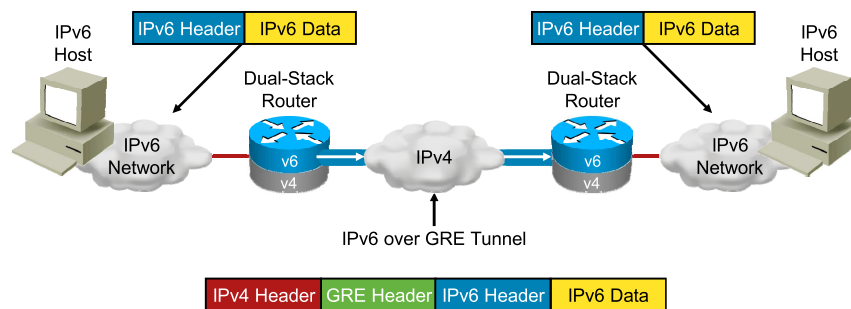
© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD2 v3.0-6-9

Regular IPv6-in-IPv4 tunnels tunnel IPv6 packets directly in IPv4 header as payload. The advantage of this is lower overhead. However, this also means that the configured tunnel can only carry IPv6 as payload. Nevertheless if you need to transport only IPv6, this is the preferred method.

Manually Configured Tunnels: GRE Tunnels

- Configure dual-stack IPv4 and IPv6 addresses on GRE tunnel interface
- Identify tunnel entry and exit with IPv4 addresses
- Required for using IS-IS protocol



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD2 v3.0-6-10

IPv6 packets can be carried over a Generic Routing Encapsulation (GRE) tunnel by encapsulating the IPv6 packet as the payload of the GRE packet. The resulting GRE packet can then be encapsulated in some other protocol (IPv4 in the example) and forwarded.

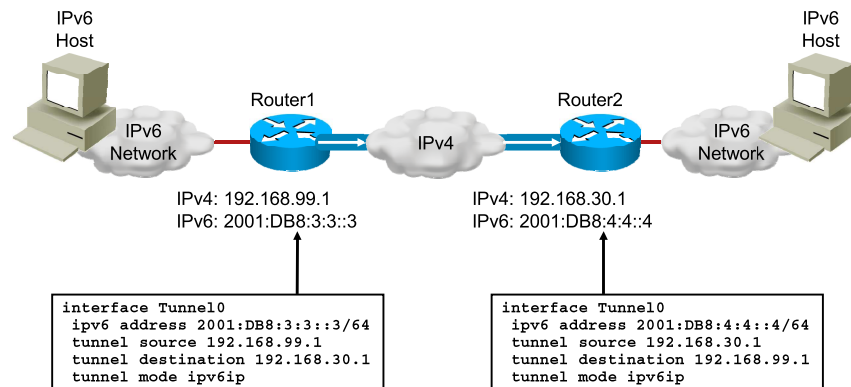
The IPv6 over IPv4 GRE tunnel uses the standard GRE tunneling technique, which provides services to implement any standard point-to-point encapsulation scheme. Similar to manually configured IPv6-in-IPv4 tunnels, GRE tunnels link two points, with a separate tunnel for each link. GRE tunnels are not tied to a specific passenger or transport protocol, but, in this case, they carry IPv6 traffic as the passenger protocol over GRE as the carrier protocol.

If the domain uses Intermediate System-to-Intermediate System (IS-IS) protocol for internal routing, only GRE tunnels can be used because IS-IS protocol runs over a Layer 2 data link. IS-IS protocol traffic can coexist on the same GRE tunnel as IPv6 traffic.

If IPv4 protocol 47 (“GRE”) is filtered anywhere along the path, the tunnel will not work.

Tip IPv6 over GRE over IPv4 is really only needed to support networks using the I/IS-IS routing protocol. For most deployments, straight protocol-41 tunneling is simpler and more efficient.

IPv6-in-IPv4 Tunnels Configuration Example



The example shows the configuration of two Cisco routers connected through an IPv4 tunnel. **Interface Tunnel0** is used, with the tunnel source and destination specified as the router IPv4 physical interface addresses. A static IPv6 address is configured on the Tunnel0 interface.

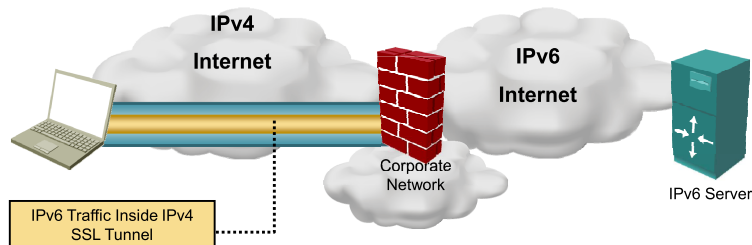
The command that enables the IPv6-in-IPv4 tunneling is **tunnel mode ipv6ip**.

You can also use the loopback interface as the source, which could have a more stable IPv4 address.

On Cisco routers, if the tunnel interface is configured properly, these interfaces will show as “up/up”. This does not imply that the tunnel is functional, however, because, in the absence of any keepalive mechanism, the virtual router interface cannot tell, for example, if there is an access control list (ACL) in the path breaking protocol-41 tunneling.

Cisco AnyConnect: IPv6 for Remote Devices Using VPN

- Remote nodes using Cisco AnyConnect can connect over a public IPv4 connection and access the IPv6 Internet



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD2 v3.0-6-12

Cisco AnyConnect is the new VPN client for SSL VPN connections. Currently it supports IPv6 tunneling inside SSL VPN over IPv4. It does not support IPv6 SSL VPN over IPv6.

IPv6 requires the following configuration on the Cisco ASA:

1. IPv6 on the outside interface
2. IPv6 on the inside interface
3. Local pool of IPv6 addresses for VPN clients
4. Local pool of IPv4 addresses for VPN clients
5. IPv6 default gateway

Note It is necessary to configure IPv4 address pool for the clients; however those IPv4 addresses do not need connectivity to the Internet.

Automatic Tunnels

This topic describes how automatic tunnels work.

Automatic Tunnels

A tunnel across an IPv4-only network is automatically created by a dual-stack host or network:

- Manual configuration of tunnel endpoints is not required.
- Automatic tunnel types include:
 - **6to4**: Used for interconnecting islands in IPv6
 - **ISATAP**: “Intranet” format, not designed for public networks
 - **Teredo**: Host terminated tunnels, which require helper servers on the Internet and are able to work through IPv4 NAT devices

© 2010 Cisco Systems, Inc. All rights reserved.IPv6FD2 v3.0—6-14

Automatic tunnels differ from static tunnels in that they are automatically connected to a remote point and brought up. At the time of configuration of one endpoint, the other endpoint is not known. Some of such tunnels scale better, because you do not have to explicitly configure all endpoints.

Automatic tunnels require relays, which often results in pervasive traffic paths. There is a danger of potentially broken connectivity if there is a firewall in the way which denies tunneled traffic. Another downpoint of such tunnels is that they depend on 3rd party helper servers/relays somewhere in the Internet. These tunnels often do not support multicast as well.

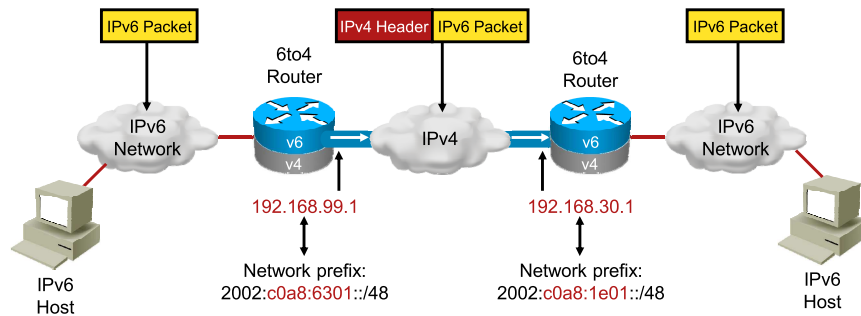
In this topic we will cover three types of automatic tunnels:

- 6to4 tunnels, which are best for connecting IPv6 network islands across the IPv4 Internet. The disadvantage of these tunnels is that the network prefix is defined by the IPv4 address. Therefore it is impossible to use a LIR-assigned global IPv6 address with 6to4.
- ISATAP tunnels, which are meant for enterprise networks, where the network infrastructure does not yet support IPv6, while the end nodes do or require IPv6 connectivity. In contrast with 6to4, ISATAP tunneling uses IPv4 information to build the host part of the IPv6 address.
- Teredo tunnels, which can be used by end nodes that support IPv6, while their entire network only supports IPv4. With the help of Internet Teredo servers, bidirectional connectivity can be established.

Note As described later in this lesson, Teredo tunnels have significant security implications, and are not the preferred way to deploy IPv6 on end nodes.

6to4 Tunnel

- An automatic tunnel method
- Gives a prefix to the attached IPv6 network, based on IPv4 address
- 6to4 specific prefix is 2002::/16



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD2 v3.0-6-15

The 6to4 tunnel method is an automatic tunnel establishment method that enables the connection of IPv6 islands through an IPv4 network. It also gives a valid IPv6 prefix to each IPv6 island, which enables the fast deployment of IPv6 without the need to get addresses from Internet service providers (ISPs) or registries, because the 2002::/16 address block is specifically reserved for 6to4 tunneling.

6to4 requires a special code on the edge routers, but the IPv6 hosts and routers inside the 6to4 site do not require any new features to support 6to4. Each 6to4 site receives a /48 prefix that is the concatenation of 2002 and the IPv4 address of the edge router. For example, if the IPv4 address of the edge router is 192.0.2.1, the prefix of its IPv6 network will be 2002:<ipv4 address of the edge router>::/48, which is 2002:c000:201::/48, because c000:201 is the hexadecimal representation of 192.0.2.1. The IPv6 network can then subnet this space as with any /48 prefix.

Note The IPv4 address 192.0.2.1 is given here just as an example. 6to4 relays use the 192.88.99.1 anycast address in the Internet.

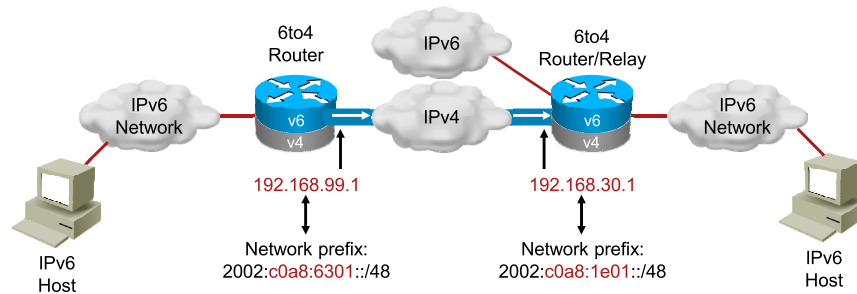
When an IPv6 packet with a destination address in the 2002::/16 range reaches the 6to4 edge router, the 6to4 router extracts the IPv4 address embedded in the 2002:: destination address. The 6to4 router will then encapsulate the IPv6 packet in an IPv4 packet, with the IPv4 destination address extracted from inside the IPv6 destination address. This IPv4 address represents the address of the other 6to4 edge router of the destination 6to4 site. The destination edge router decapsulates the IPv6 packet in the IPv4 packet and then forwards the native packet toward its final destination.

Note that if IPv4 protocol 41 (“IPv6-in-IPv4 encapsulation”) is filtered anywhere along the path, the tunnel will not work.

Tip Notice that if the IPv4 address of the border router changes the entire site /48, self-allocation also changes, and the 6to4 environment must be renumbered.

6to4 Relay

- Acts as a gateway between IPv6 Internet and 6to4 network
- Public 6to4 relays are available at 192.88.99.1



The 6to4 tunnel method enables the edge router to forward packets to any destination with a 2002::/16 prefix. However, all other destinations are unreachable unless one of the 6to4 edge routers, called a 6to4 relay, offers traffic forwarding to the IPv6 Internet. In that case, all other 6to4 routers include a default route entry to this 6to4 relay.

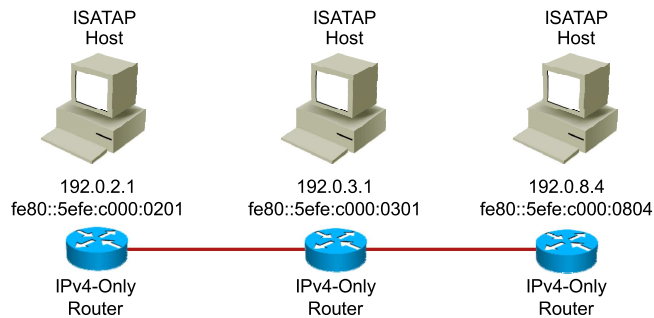
In Cisco IOS software, the following command would be used on Router1 to forward all non-6to4 traffic to the 6to4 relay: **ipv6 route ::/0 <ipv6 address of the 6to4 relay>**.

Note that the 192.168.0.0/16 addresses used in the example are for demonstration only, and that they are RFC 1918 (*Address Allocation for Private Internets*) addresses and will not work as or through a 6to4 relay.

Tip In many ways, the 6to4 relay routers provide transit services from the IPv4 Internet to the IPv6 Internet in the same way that 6to4 site routers provide transit from the IPv4 Internet to their internal /48 IPv6 site prefix.

ISATAP

- ISATAP uses IPv4 as an NBMA link-layer for IPv6 over IPv4
- Interface identifiers are built from IPv4 address:
192.0.2.1 → 0000:5EFE:C000:0201
- Router discovery is done via PRL (potential router list)



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD2 v3.0-6-17

Using ISATAP, host computers configure addresses of the form “0:5EFE:w.x.y.z”, in which “w.x.y.z” is the IPv4 address of a host interface. Using these addresses, ISATAP-capable hosts throughout a larger IPv4-routed network can tunnel IPv6 packets to each other, treating the IPv4 infrastructure as a single nonbroadcast multiaccess (NBMA) network. ISATAP hosts can deliver the IPv6-in-IPv4 packet to the right ISATAP peer host by using the embedded destination IPv4 address. In this case, an ISATAP host need only configure the link-local IPv6 address. Remember that ISATAP treats the entire enterprise IPv4-routed network as a single NBMA link.

In order for this mechanism to work, each host must have an IPv4 address routable within an enterprise and must implement ISATAP. Part of the ISATAP process is building a static default route that places all IPv6 traffic into an ISATAP interface.

Note that, within an enterprise, ISATAP works with RFC 1918 or other nonglobally routable addresses.

An ISATAP host (even with a nonglobally routable IPv4 address) can attain global IPv6 connectivity via the services of an ISATAP router, which is a router that implements ISATAP and also has an IPv6-facing interface through which to forward traffic bound to the greater IPv6 Internet.

Tip Notice that if the IPv4 address of the ISATAP node changes, the IPv6 ISATAP address changes, because the IPv4 address is embedded directly in the IPv6 address.

The ISATAP router must have a valid prefix for the site (such as 2001:db8:1::/48) from which it can allocate addresses internally, and the ISATAP router must be a fully functioning member of the IPv6 Internet.

To obtain a unique 128-bit IPv6 address, the ISATAP host must obtain a valid /64 prefix. At a high level, this is what the host does:

1. Queries the Domain Name Service (DNS) for the address of ISATAP, which should return an IPv4 address of the previously configured ISATAP edge router

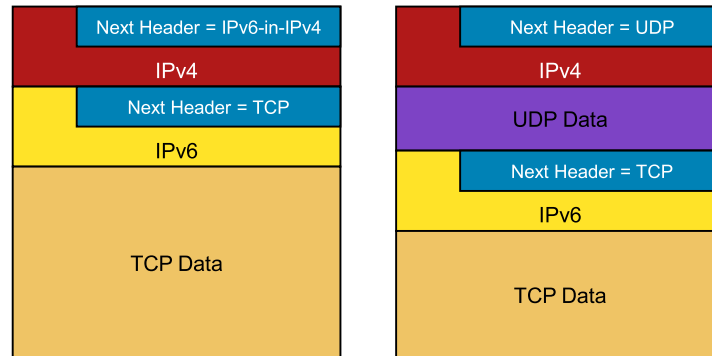
2. Sends a router solicitation request to the router, tunneled in IPv4
3. Configures a 128-bit address by combining the /64-bit prefix with the ISATAP address (the lower 64 bits of the 5EFE:w.x:y.z address)
4. Configures a default route for IPv6, destined for the ISATAP router

Now, traffic bound from an ISATAP host to any fe80::/10 address will be delivered directly, via an ISATAP host to an ISATAP-host IPv4 automatic tunnel. Traffic bound for the greater IPv6 Internet would not match the IPv6 link-local route but would match the IPv6 default route, so the packet would be tunneled to the ISATAP router. The ISATAP router, in turn, removes the IPv4 headers and routes the IPv6 packet as it would any IPv6 packet.

Traffic inbound from the IPv6 Internet to the internal ISATAP host would work similarly, with the packet being delivered first to the ISATAP border router, which, checking its ISATAP configuration and routes, delivers the packet tunneled in IPv4.

Teredo

Encapsulates IPv6 packet in IPv4 UDP rather than IPv4 protocol 41 (passes NAT)



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD2 v3.0-6-18

When tunneling from hosts, IPv4-encapsulated IPv6 packets are sent with the Protocol field in the IPv4 header set to 41. This presents an issue for NATs. Most NATs only translate TCP or User Datagram Protocol (UDP) traffic and must either be manually configured to translate other protocols or have installed NAT editors that handle the translation. Because IPv4 protocol 41 translation is not a common feature of NATs, IPv4-encapsulated IPv6 traffic will not flow through typical NATs. Therefore, to allow IPv6 traffic to flow through one or multiple NATs, the IPv6 packet is encapsulated as an IPv4 UDP message, containing both an IPv4 and UDP header. UDP messages can be universally translated by NATs and can traverse multiple layers of NATs.

In the figure, the left packet uses IP protocol 41 (IPv6-in-IPv4), which most NATs will not handle. By encapsulating the entire IPv6 packet in an IPv4 packet with UDP, the IPv4 packet looks normal to NAT and will pass. At the far end of the tunnel, the IPv4 UDP packet layers will be stripped off.

Teredo Characteristics

Teredo is an IPv6–IPv4 transition technology, which allows automatic IPv6 tunneling between hosts that are located across IPv4 NAT translations:

- IPv6 traffic sent as IPv4 UDP messages
- Last resort transition technology for IPv6 connectivity
- Has significant security concerns

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD2 v3.0—6-19

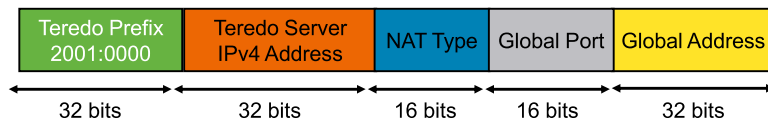
These are some characteristics of Teredo:

- IPv6 traffic from Teredo hosts can flow across NATs because the traffic is sent as an IPv4 UDP message. If the NAT supports UDP port translation, the NAT supports Teredo. The exception is a symmetric NAT.
- Teredo is designed as a last resort transition technology for IPv6 connectivity. If native IPv6, 6to4, or ISATAP connectivity is present, the host does not act as a Teredo client. As more IPv4 NATs are upgraded to support 6to4 and IPv6 connectivity, Teredo will be used less frequently until it is finally not used at all. Teredo also has several significant security issues.
- Teredo servers are stateless and only manage the initial tunnel establishment. After a tunnel is established, Teredo relays actually pass the traffic.

Teredo Address Structure

IPv6 address encoding embeds:

- Global address and port
- Teredo server address
- Type of NAT box



© 2010 Cisco Systems, Inc. All rights reserved.

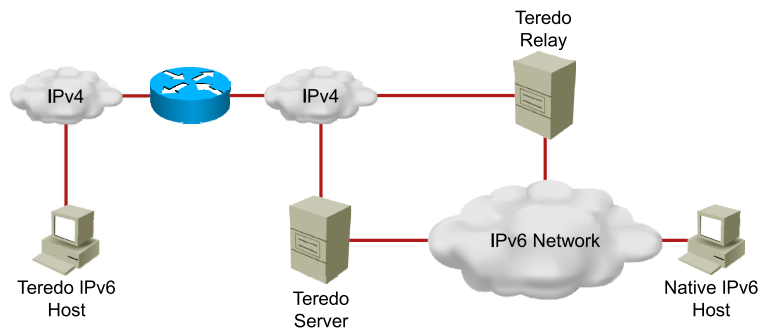
IP6FD2 v3.0-6-20

The address structure of Teredo includes these elements:

- **Teredo prefix (prefix):** The first 32 bits are for the Teredo prefix, which is the same for all Teredo addresses. The Internet Assigned Numbers Authority (IANA) has assigned 2001::/32.
- **Teredo server IPv4 address:** The next 32 bits contain the IPv4 public address of the Teredo server that helped configure this Teredo address.
- **Flags (NAT type):** The next 16 bits are reserved for Teredo flags. The only defined flag is the high-order bit known as the Cone flag. The Cone flag is set when the NAT connected to the Internet is a Cone NAT. The determination of whether the NAT connected to the Internet is a Cone NAT occurs during the initial configuration of the Teredo client.
- **Obscured external port (global port):** The next 16 bits store an obscured version of the external UDP port corresponding to all Teredo traffic for this Teredo client. When the Teredo client sends its initial packet to a Teredo server, the source UDP port of the packet is mapped by the NAT to a different, external UDP port. The Teredo client maintains this port mapping so that it remains in the translation table of the NAT. Therefore, all Teredo traffic for the host uses the same external, mapped UDP port. The external UDP port is determined by the Teredo server from the source UDP port of the incoming initial packet sent by the Teredo client and sent back to the Teredo client.
- **Obscured external address (global address):** The last 32 bits store an obscured version of the external IPv4 address corresponding to all Teredo traffic for this Teredo client. Just like the external port, when the Teredo client sends its initial packet to a Teredo server, the source IP address of the packet is mapped by the NAT to a different, external (public) address. The Teredo client maintains this address mapping so that it remains in the translation table of the NAT. Therefore, all Teredo traffic for the host uses the same external, mapped, public IPv4 address. The external IPv4 address is determined by the Teredo server from the source IPv4 address of the incoming initial packet sent by the Teredo client and sent back to the Teredo client.

Components of Teredo Infrastructure

- Teredo host
- Teredo server
- Teredo relay



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD2 v3.0—6-21

A Teredo infrastructure consists of these devices:

- **Teredo client:** A Teredo client is an IPv6/IPv4 node that supports a Teredo tunneling interface through which packets are tunneled to either other Teredo clients or nodes on the IPv6 Internet (via a Teredo relay). A Teredo client communicates with a Teredo server to obtain an address prefix from which a Teredo-based IPv6 address is configured or to help initiate communication with other Teredo clients or hosts on the IPv6 Internet.
- **Teredo server:** A Teredo server is an IPv6/IPv4 node that is connected to both the IPv4 Internet and the IPv6 Internet. It supports a Teredo tunneling interface over which packets are received. The general role of the Teredo server is to assist in the address configuration of the Teredo client and to facilitate the initial communication between Teredo clients or between Teredo clients and IPv6-only hosts. The Teredo server listens on UDP port 3544 for Teredo traffic.
- **Teredo relay:** A Teredo relay is an IPv6/IPv4 router that can forward packets between Teredo clients on the IPv4 Internet (using a Teredo tunneling interface) and IPv6-only hosts. In some cases, the Teredo relay interacts with a Teredo server to help it facilitate initial communication between Teredo clients and IPv6-only hosts. The Teredo relay listens on UDP port 3544 for Teredo traffic. Individual ISPs could deploy their own Teredo relays. Teredo relays are not needed to communicate with Teredo host-specific relays.
- **Teredo host-specific relay:** Communication between Teredo clients and IPv6 hosts that are configured with a global address must go through a Teredo relay. This is required for IPv6-only hosts connected to the IPv6 Internet. However, when the IPv6 host is IPv6-capable and IPv4-capable and connected to both the IPv4 Internet and IPv6 Internet, communication should occur between the Teredo client and the IPv6 host over the IPv4 Internet, rather than having to traverse the IPv6 Internet and go through a Teredo relay. A Teredo host-specific relay is an IPv6/IPv4 node that has an interface and connectivity to both the IPv4 Internet and the IPv6 Internet and can communicate directly with Teredo clients over the IPv4 Internet without the need for an intermediate Teredo relay. The connectivity to the IPv4 Internet can be through a public IPv4 address or through a private IPv4 address and a neighboring NAT. The connectivity to the IPv6 Internet can be through

a direct connection to the IPv6 Internet or through an IPv6 transition technology such as 6to4, where IPv6 packets are tunneled across the IPv4 Internet. The Teredo host-specific relay listens on UDP port 3544 for Teredo traffic.

Following stages show how traffic can flow from an outside IPv6-only host to a teredo host, even if IPv6-only host is the one who initiates the connection.

Stage 1—IPv6 host sends initial communication packet: The IPv6-only host sends an initial communication packet to the Teredo client. Because of the routing infrastructure of the IPv6 Internet, the Teredo-addressed packet is forwarded to the nearest Teredo relay.

Stage 2—Teredo relay sends a bubble packet to the Teredo server: The Teredo relay determines that the Teredo client is behind a restricted NAT. If the Teredo relay were to send the packet to the Teredo client, the NAT would silently discard it because there is no source-specific mapping for Teredo traffic from the Teredo relay. Therefore, the Teredo relay sends a bubble packet to the Teredo client via the Teredo server over the IPv4 Internet.

Stage 3—Teredo server forwards bubble packet to the Teredo host: The Teredo server receives the bubble packet from the Teredo relay. The Teredo server forwards the bubble to the Teredo client, with an origin indicator that contains the IPv4 address and UDP port number of the Teredo relay. Because a source-specific mapping for Teredo traffic from the Teredo server exists in the NAT, the bubble packet is forwarded to the Teredo client.

Stage 4—Teredo host sends bubble packet back to Teredo relay: The Teredo client determines the IPv4 address of the Teredo relay closest to the IPv6-only host from the origin indicator of the received bubble packet. To establish a source-specific mapping for Teredo traffic from the Teredo relay, the Teredo client sends a bubble packet to the Teredo relay.

Stage 5—Teredo relay transmits original packet to Teredo host: After receiving the receipt of the bubble packet, which corresponds to an IPv6-only host packet that is queued for forwarding, the Teredo relay determines that a source-specific NAT mapping now exists in the restricted NAT of the Teredo client. The Teredo relay forwards the packet to the Teredo client.

Stage 6—Subsequent packets flow directly: The Teredo client sends the response to the initial communication packet to the IPv6-only host.

Tunnel Review

Tunnel Type	Features
Manually Configured	Stable "always on" link Used to interconnect sites IPv6-in-IPv4, GRE, and IPsec/VPN
Tunnel Broker and Tunnel Server	Semiautomatic tunnel Allocates IPv6 addresses or prefixes to end users or sites
6to4	Automatic tunnel IPv6 prefix formed using IPv4 address (2002:<IPv4>::/48) Used by end sites
ISATAP	Automatic tunnel IPv6 address formation = <Prefix/64>:0:5efe:<IPv4/32> Intrasite mechanism for sparse enterprise deployments
Teredo	Automatic tunnel IPv6 address formation = 2001:0000:<Teredo Server's IPv4/32>:<NAT type/16>:<Global Port/16>:<Global Address/32> NAT aware, used by SOHO users

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD2 v3.0—6-22

The table gives a short summary of the tunneling techniques discussed in this module.

Manually configured static links have the benefit of being easy to administer and troubleshoot. They allow usage of any IPv6 addresses and run over any transit network. Scalability is low, but it may be improved with the use of GRE/NHRP.

Tunnel broker provided tunnels are used when there is no IPv6 service available at directly connected ISP. Technology used to encapsulate IPv6 traffic is dictated by the tunnel broker. Usually these tunnels are semiautomatic; the tunnel provider supports dynamic tunnel destination addresses, but otherwise they still function like regular tunnels.

6to4 tunnels have the advantage of greater scalability – only one interface for all tunnel endpoints. The disadvantages include reliance on IPv4 for transport (6to4 can only function with an underlying IPv4 network) and addresses based on IPv4 tunnel endpoint.

ISATAP treats IPv4 network as an NBMA medium for overlaying IPv6. Multiple network segments in IPv4 can appear as one subnet in IPv6 built by ISATAP has basically the same advantages and disadvantages as 6to4 with the added complexity of relying on DNS for getting the PRL and ability to use any prefix for numbering of subnets.

All tunneling technologies so far rely either on IPv6 in IPv4 (protocol 41) or GRE (protocol 47) tunneling. These tunnels may have difficulties traversing firewalls and NAT devices. Teredo is a way of tunneling IPv6 over UDP/IPv4 and can traverse such devices with ease, if the UDP traffic is permitted in the outbound direction. Because it allows subsequent inbound connections over IPv6 to bypass the firewall it may cause a security breach and most administrators will want to block Teredo.

Summary

This topic summarizes the key points discussed in this lesson.

Summary

- Configured tunnels (IPv6-in-IPv4, GRE, VPN) are stable and reliable but are difficult to manage as the number of tunnels required increases; each tunnel must have administrator-configured IPv4 endpoints.
- Automatic tunneling mechanisms scale better than configured tunnels but are less versatile and more difficult to troubleshoot when issues arise. Also, automatic tunneling mechanisms solve specific problems:
 - 6to4 connects an IPv6-enabled island to another island or native IPv6 Internet across an IPv4-only network.
 - ISATAP, by contrast, provides IPv6 capability to sparsely scattered dual-stack nodes to an enterprise with an IPv4-only backbone.
 - Teredo allows an IPv6 host on a private IPv4 network to talk to other IPv6 hosts.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD2 v3.0—6-23

Summary (Cont.)

- Tunnel brokers aid with configuring manual tunnels to make them much simpler for nontechnical users to deploy. These brokers make IPv6-enabling a node appear "automatic" to the end user.
- When deploying or permitting any sort of tunneling mechanism, great care must be taken to not introduce security issues or weaken existing controls.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD2 v3.0—6-24

Module Summary

This topic summarizes the key points that were discussed in this module.

Module Summary

- Dual stack is the most common IPv6 transition approach.
- Satisfying the technical and economic requirements for your IPv6 transition effort requires that you assess and utilize a number of choices.
- NAT-PT is a translation mechanism that allows IPv4-only and IPv6-only nodes to exchange traffic.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0--6-1

Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

- Q1) A dual-stacked node can be described as having which two of these characteristics? (Choose two.) (Source: Implementing Dual-Stack)
- A) both an IPv4 and IPv6 networking stack
 - B) can only make inquiries to a dual-stacked DNS server
 - C) can make DNS inquiries and understand both A and AAAA records
 - D) can only use IPv4-enabled applications
 - E) will always choose IPv6 DNS records over IPv4
- Q2) Which of these stage sequences accurately describes the address selection process for a dual-stacked node? (Source: Implementing Dual-Stack)
- A) DNS request is sent from host, DNS queries application for address status, DNS server sends response, application connects to destination host
 - B) DNS request is sent to host, DNS queries application for address version support, DNS server sends response, application connects to destination host
 - C) DNS request is sent from host, application sends record preference, DNS sends correct record, application connects to host
 - D) DNS request is sent from host, DNS server sends all available records, application selects correct address, application connects to destination host
- Q3) What is a significant issue for IPv6-in-IPv4 manual tunnels? (Source: Describing IPv6 Tunneling Mechanisms)
- A) Manual tunnels only run well over the Ethernet.
 - B) Manual tunnels only support the 6to4 transition mechanism.
 - C) Manual tunneling does not scale well.
 - D) Manual tunnels require that Layer 4 protocols be modified.
- Q4) For which purpose is a 6to4 transition mechanism used? (Source: Describing IPv6 Tunneling Mechanisms)
- A) primarily as an intranet solution
 - B) primarily as a public Internet solution for connecting isolated IPv6 networks
 - C) primarily as a transition mechanism for individual nodes with nonroutable IPv4 addresses
 - D) primarily for IPv4-in-IPv6 tunneling
- Q5) What are the ISATAP components called? (Source: Describing IPv6 Tunneling Mechanisms)
- A) client, router
 - B) client, server
 - C) client, relay
 - D) client, relay, and server

- Q6) Which limitations are there to using IPsec to secure IPv4-in-IPv6 tunnels? (Source: Describing IPv6 Tunneling Mechanisms)
- A) The outer tunnel must be secured with IPv6 IPsec protection.
 - B) The inner packets must be secured with IPv4 IPsec.
 - C) Only IPsec AH can be used or ESP with null encryption.
 - D) There are no limitations—IPv6 IPsec, IPv4 IPsec, AH, or ESP can be used.
- Q7) What does DSTM require? (Source: Describing IPv6 Tunneling Mechanisms)
- A) The client must be dual stack and running the DSTM client software.
 - B) The DSTM address server must be running IPv4.
 - C) The DSTM tunnel endpoint must be running IPv6 Integrated IS-IS.
 - D) The DSTM client must be IPv4-only with translation.
- Q8) Which of these statements is true about the NAT-PT transition mechanism? (Source: Describing NAT-PT)
- A) It provides IPv6-to-IPv4 address translation for outbound IPv6 traffic, enabling IPv6-only islands to reach the IPv4-only Internet.
 - B) It provides IPv6-to-IPv4 and IPv4-to-IPv6 address and protocol translation, enabling IPv4-only and IPv6-only nodes to communicate.
 - C) It provides NAT-PT tunneling for carrying IPv6 traffic over an intervening IPv4-only backbone network.
 - D) It provides IPv4-to-IPv6 protocol translation in cooperation with the ISATAP mechanism to allow intranet-based nodes access to internal resources across an IPv4-only enterprise backbone.
- Q9) How does NAT-PT benefit from the use of a DNS-ALG? (Source: Describing NAT-PT)
- A) Nodes initiating new sessions can use the destination hostname for the session target, and the DNS-ALG will return a protocol-appropriate address to enable the session to begin.
 - B) Nodes that are initiating new sessions can open connections toward nodes running the other protocol—using the native protocol—and the NAT-PT device will contact the DNS-ALG for address resolution.
 - C) DNS-ALG makes it possible for any-to-any connectivity through the NAT-PT device, using automatic source address translation at the initiating node.
 - D) DNS-ALG provides more efficient translation between IPv4 and IPv6 for data packets using the SIIT (stateless translator) technology.

Module Self-Check Answer Key

Q1)	A, C
Q2)	D
Q3)	C
Q4)	B
Q5)	B
Q6)	D
Q7)	A
Q8)	B
Q9)	A

IPv6 Security

Overview

Moving transactions and communications in an IP-based network requires support services that were not originally called for in the IP version 4 (IPv4) specifications. One of these services, security, is required to help ensure that access to sensitive data and restricted content is kept safe, data transmissions are encrypted, and unauthorized access is prohibited. IP version 6 (IPv6) fundamentally changes some well-known security paradigms and introduces some security issues. This module describes security issues, how security for IPv6 is different from IPv4, and emerging practices for IPv6-enabled networks.

Module Objectives

Upon completing this module, you will be able to describe security issues, how security for IPv6 is different than for IPv4, and emerging practices for IPv6-enabled networks. This ability includes being able to meet these objectives:

- Configure ACLs in an IPv6 environment
- Describe how security is implemented in IPv6 and associated topics
- Identify security and transition issues in an IPv6 environment
- Describe security practices for IPv6 deployment
- Describe and configure Cisco IOS Firewall for IPv6 traffic

Configuring IPv6 ACLs

Overview

Successful implementation of IP version 6 (IPv6) services is essential to the creation of a secure and robust network design. Implementing IPv6 security features in Cisco IOS Software involves creating access control lists (ACLs). This lesson describes the features of ACLs in an IPv6 environment.

Objectives

Upon completing this topic, you will be able to configure ACLs in an IPv6 environment. This includes being able to meet these objectives:

- Describe standard and extended ACLs in IPv6 networks
- Describe configuration of an ACL in IPv6 networks
- Describe reflexive and time-based ACLs in IPv6 networks
- Identify Cisco IOS Software IPv6 header filtering keywords
- Identify new Cisco IOS Software ICMPv6 types
- Edit IPv6 ACLs
- Configure ACLs in an IPv6 environment

IPv6 ACLs

This topic describes ACLs in IPv6 networks.

IPv6 ACLs

- All IPv6 ACLs are:
 - Named ACLs
 - Extended ACLs
- Configuration commands are slightly different from IPv4 commands.
- Implicit deny at the end of ACL is different.
- There are no more wildcards (masks only).
- IPv6 access lists are sequenced:
 - Individual entries can be added and removed.
 - ACLs cannot be resequenced.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-3

ACLs are used for packet matching. Often, but not always, ACLs are used for security purposes. The packets are permitted or denied through an interface. You can also use ACLs to match packets to be used for other reasons, such as selecting packets for quality of service (QoS) processing. In this lesson, security-related ACLs are addressed.

In contrast with IP version 4 (IPv4) ACLs, all IPv6 ACLs are named and extended ACLs. Some commands have changed slightly, but all the concepts remain the same.

The implicit deny at the end of the ACLs has changed.

Instead of wildcards, netmask length is used.

Although IPv6 ACLs are sequenced like named IPv4 ACLs and adding or removing individual entries is possible, resequencing an ACL is not possible.

IPv6 ACLs (Cont.)

Checked Parameters

Version	Traffic Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address (128)			
Destination Address (128)			
Next Header	Extension Header Length(8)=?		
Hop-by-Hop Header Options			
Source Port (16)		Destination Port (16)	
TCP Header and Data 5 x 21 Bits of Other Parameters			

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7.4

An IPv6 ACL can match packets, based on the following:

- Source address
- Destination address
- Layer 4 protocol
- Higher-layer fields such as port numbers; for example, port numbers in User Datagram Protocol (UDP) and TCP
- Internet Control Message Protocol (ICMP) type fields and codes; for example, ICMP type 4, code 1—parameter problem, unrecognized next header field
- Traffic class
- Differentiated services code point (DSCP) value—used for differentiated services (DiffServ) in QoS
- Flow label
- Presence of extension headers; for example, fragments and routing

IPv6 ACL Configuration

This topic describes configuration of an ACL in IPv6 networks.

IPv6 ACL Configuration

- Example ACL configuration

```
ipv6 access-list Named ACL
deny 2001:db8:4::/64 any
permit ipv6 2001:db8:3:1::/64 any
permit icmp any any packet-too-big
permit tcp 2001:db8:3::/64 host 2001:db8:4::80 eq 80
```

- Attaches ACL to an interface

```
interface FastEthernet 0/0
ipv6 traffic-filter Named ACL in
```

© 2010 Cisco Systems, Inc. All rights reserved. IPv6FD v3.0—7-6

The figure shows an example of an ACL configuration.

```
ipv6 access-list Named ACL
deny 2001:db8:4::/64 any
permit ipv6 2001:db8:3:1::/64 any
permit icmp any any packet-too-big
permit tcp 2001:db8:3::/64 host 2001:db8:4::80 eq 80
```

The ACL is configured like an IPv4 named ACL. Differences are in address format and lack of wildcard masks, which are replaced by prefix lengths.

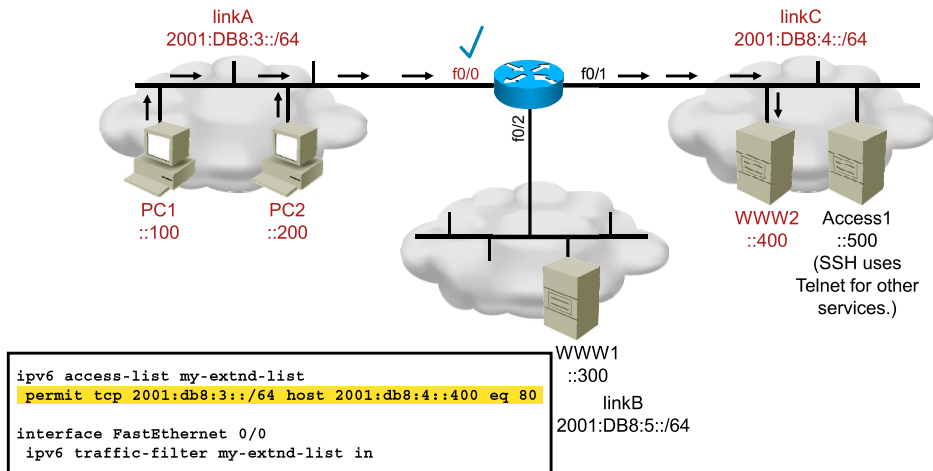
The command to attach an ACL to an interface has changed from **access-group** to **traffic-filter**. The concept and parameters are the same.

```
interface FastEthernet 0/0
ipv6 traffic-filter Named ACL in
```

Note IPv6 has capabilities that might complicate the rules of writing for IPv6. An autoconfigured and privacy IPv6 address reduces the likelihood that internal users will have long-lived IPv6 addresses. IPv6 interfaces may also have multiple addresses, so a node can choose different source addresses for different outbound connections. As a rule, an ACL should specify prefixes (subnets), as opposed to host addresses. Then, interface identifier selection mechanisms should not matter.

ACL: Practical Example

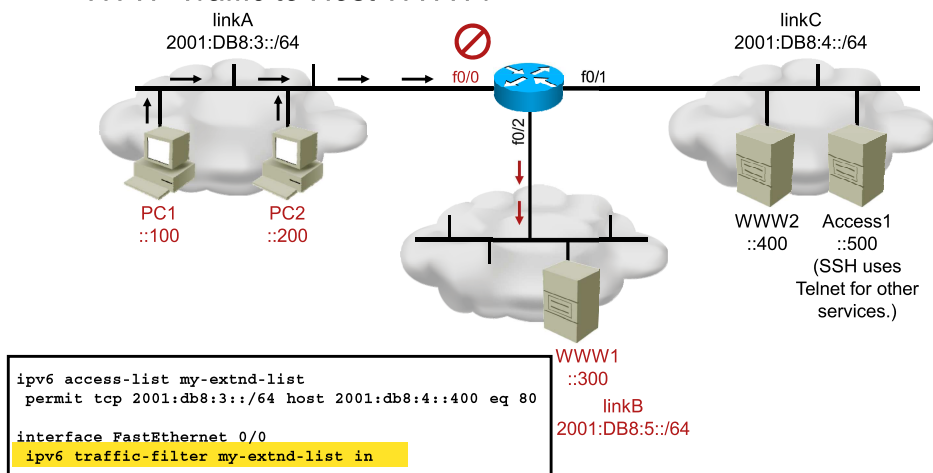
HTTP Traffic to Host WWW2



This configuration allows nodes on linkA to reach WWW2, but only on port 80. No other traffic is allowed in from linkA through Fast Ethernet 0/0, toward the router and networks beyond. No restrictions are placed on traffic that enters the router on Fast Ethernet 0/1 or Fast Ethernet 0/2, or on traffic that comes from linkB or linkC and is bound for linkA.

ACL: Practical Example (Cont.)

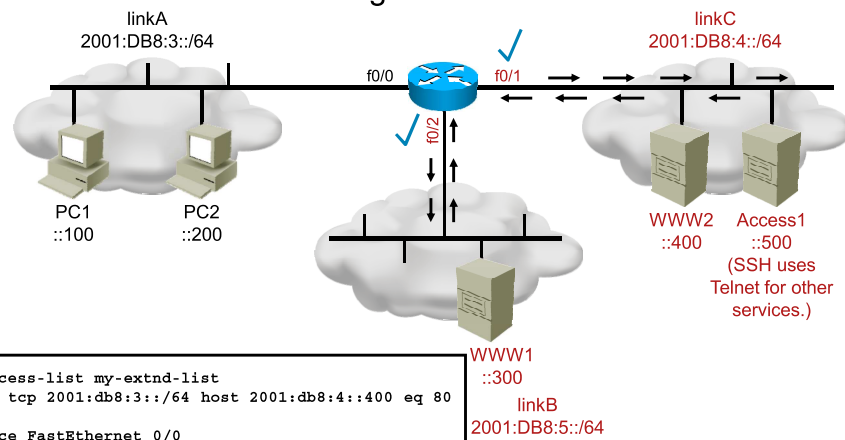
HTTP Traffic to Host WWW1



Access to node WWW1 is restricted.

ACL: Practical Example (Cont.)

Traffic between Server Segments



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7.9

Traffic between linkC and linkB can flow freely. The restrictive ACL is attached to the Fast Ethernet 0/0 interface.

```
ipv6 access-list my-extnd-list
permit tcp 2001:db8:3::/64 host 2001:db8:4::400 eq 80

interface FastEthernet 0/0
ipv6 traffic-filter my-extnd-list in
```

Reflexive and Time-Based ACLs

This topic describes reflexive and time-based ACLs in IPv6 networks.

Reflexive and Time-Based ACLs

- Reflexive ACLs provide the means to control traffic flow, based on session initiation:
 - Router tracks state.
 - Permitted outbound session automatically creates temporary converse rule for return packet flow.
- Time-based ACLs permit or deny traffic, based on a configurable time range.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-11

Reflexive ACLs

A reflexive ACL allows a router to control traffic flows, based on session initiation.

Remember that a normal ACL has no sense of state. Packets are evaluated individually, so there is no concept of a collection of packets or sessions. In an HTTP connection, for example, packets flow both ways. The HTTP client initiates the session by sending packets to the HTTP server, and then the server sends packets to the HTTP client. With a simple ACL, the router must have an ACL to explicitly permit each flow.

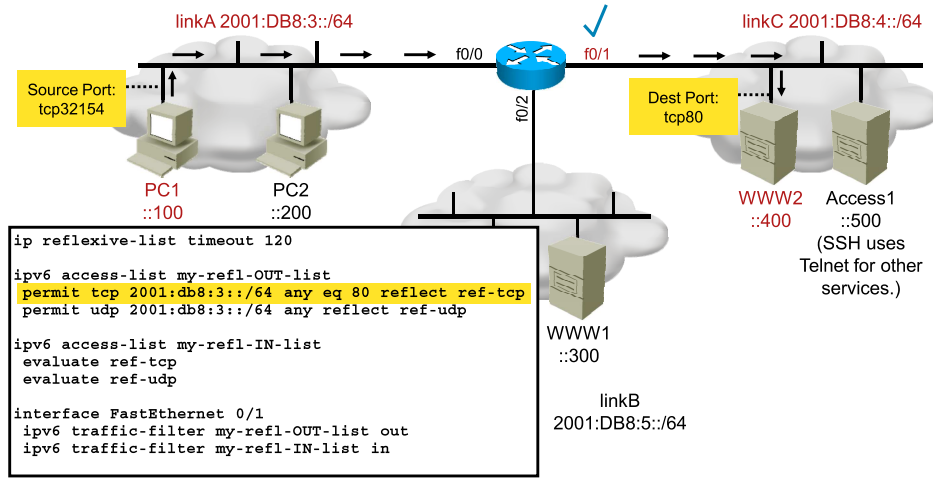
With a reflexive ACL, you will see that the return traffic ACL can be written in real time and closed after the session has concluded. For example, suppose that you have traffic entering the router from Fast Ethernet 0/0, headed out Fast Ethernet 0/1, and destined for an HTTP server on port 80 on the Internet. By applying a reflexive ACL to Fast Ethernet 0/1, you can allow that traffic out to the Internet, and you can dynamically create a rule that allows the return traffic from the HTTP server back to the requesting node. The inbound Fast Ethernet 0/1 rule that is created would be temporary and as specific as the outbound Fast Ethernet 0/1 rule. To make the same traffic flow without using a reflexive ACL, you would need two static ACLs: one on Fast Ethernet 0/1 for the outbound traffic (to the Internet-based HTTP server) and one on Fast Ethernet 0/1 for the inbound traffic. Dynamically and temporarily creating the Fast Ethernet 0/1 inbound rule is a far better option.

Time-Based ACLs

A time-based ACL permits or denies traffic, based on a configurable time range. Therefore, access can be restricted selectively at different times, without any systems administrator action. Unlike most ACLs, which are always active, time-based ACL allows the specification of periodic time ranges to enable or disable specific packet flows.

Reflexive ACL Example

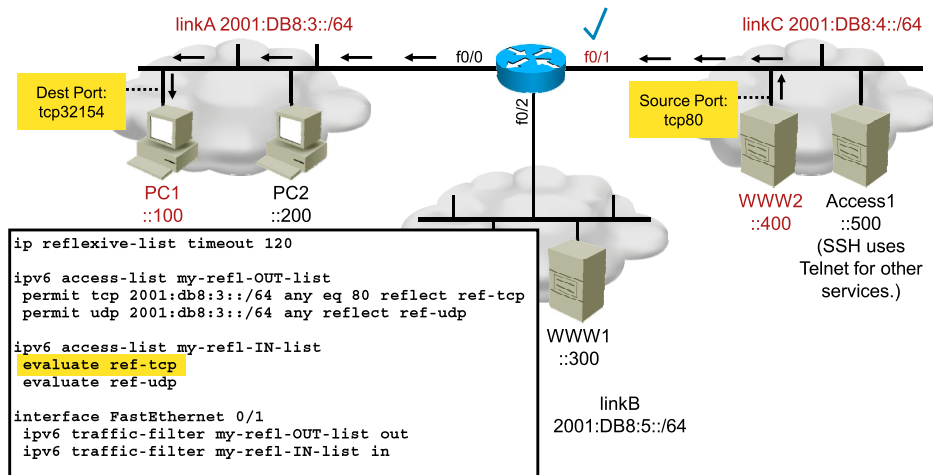
Initial Traffic from Inside



This example illustrates how a reflexive ACL works. The only TCP traffic that is allowed out the f0/1 interface is from linkA, to anywhere, but only for HTTP traffic (port 80).

Reflexive ACL Example (Cont.)

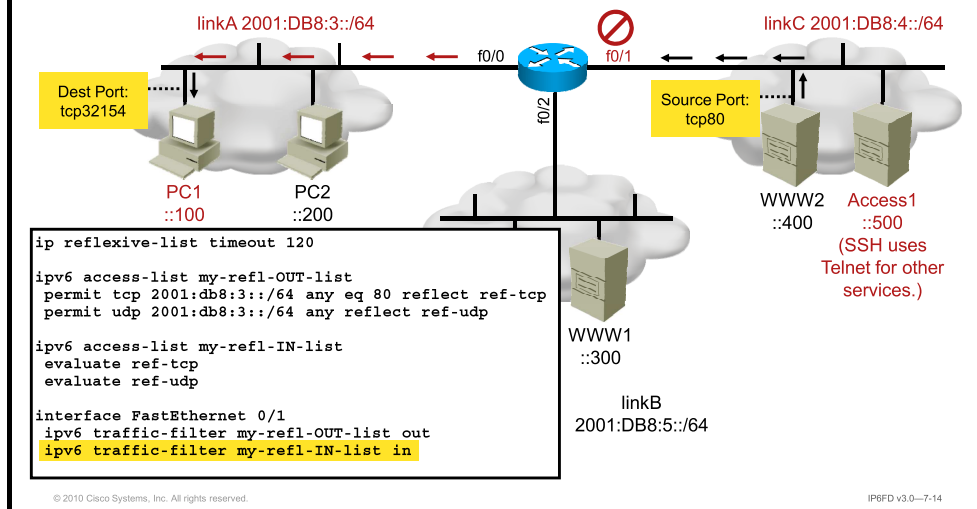
Return Traffic



However, because you are using a reflexive ACL and are applying the “my-refl-IN-list” to protect the router from inbound traffic on Fast Ethernet 0/1, only HTTP return traffic is allowed to linkA. The router and links that are behind it are protected from other traffic, and even the HTTP return traffic is denied after a session timeout value expires.

Reflexive ACL Example (Cont.)

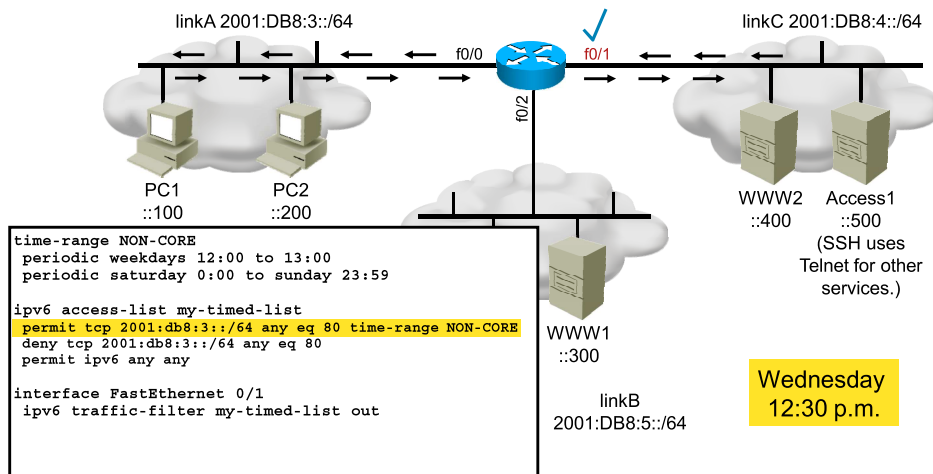
Initiating Traffic from Outside



Note that UDP traffic is allowed outbound from linkA and linkB, but no UDP traffic is allowed in from linkC to any destination. The traffic is neither permitted by rule nor by part of an evaluate expression.

Time-Based ACL Example

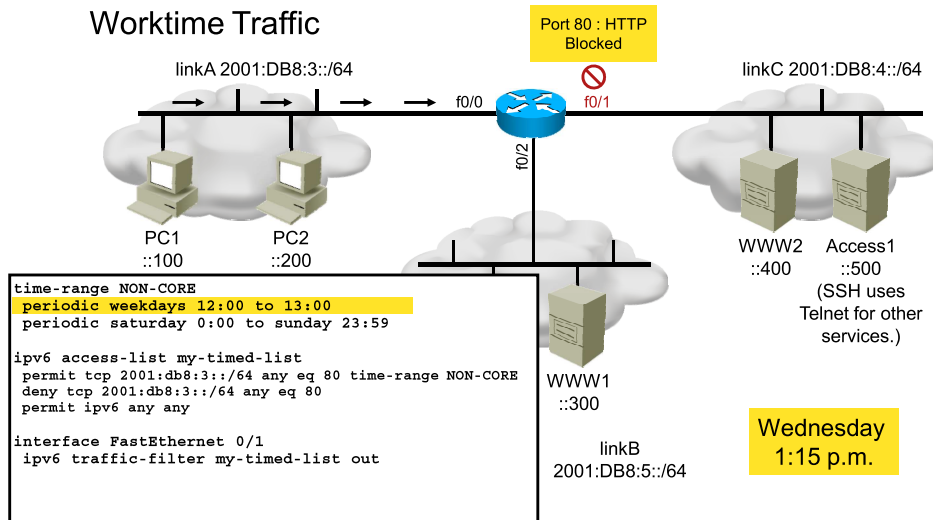
Lunchtime Traffic



Time-based ACLs allow nodes on linkA and linkB to browse linkC-based HTTP servers at lunch and on weekends. All other IPv6 traffic is permitted.

Time-Based ACL Example (Cont.)

Worktime Traffic



On the other hand, the time-based ACL does not allow nodes on linkA and linkB to browse linkC-based HTTP servers during core business hours. All other IPv6 traffic is permitted.

Cisco IOS IPv6 Header Filtering

This topic describes Cisco IOS IPv6 header filtering keywords.

Cisco IOS IPv6 Header Filtering

Extended ACLs also provide means to inspect packet headers:

- **DSCP:** Value
- **Flow label:** Value
- **Fragmentation header:** Presence
- **Routing header:** Presence, type
- **Mobility header:** Presence, type
- **Destination option header:** Presence, type
- **Authentication header:** Presence

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-18

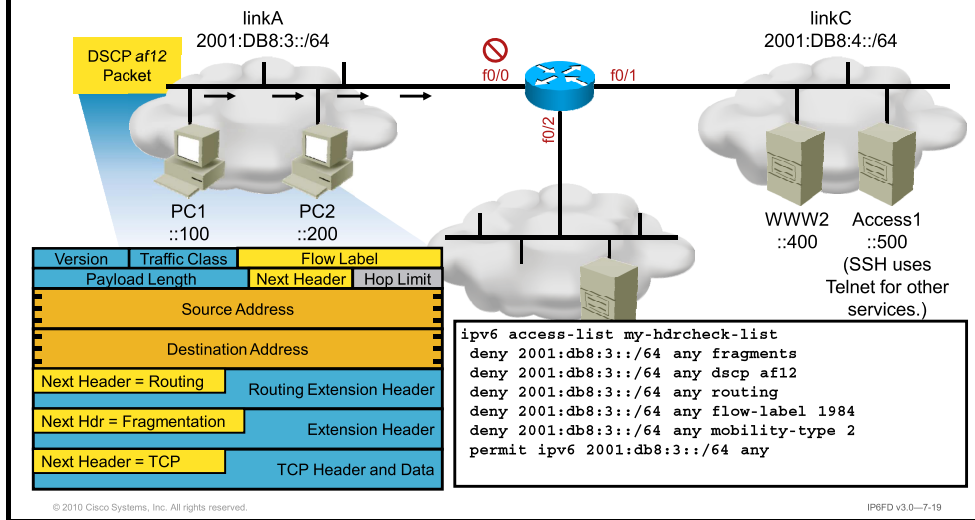
In IPv6, new fields in the base header can be examined by ACLs and selected for permit, deny, log, or other reasons. IPv6 also uses extension headers for various functions, and these can be matched as well.

IPv6 ACLs allow you to filter packets, based on the following header fields:

- **DSCP:** This parameter includes the 6 left-most bits of the IPv6 Traffic Class field in DSCP. The Traffic Class field is 8-bits wide. The DSCP field is used in providing DiffServ or a unique per-hop behavior (PHB) for traffic on a router.
- **Flow Label:** This parameter is a 20-bit field that is carried in the base IPv6 header. The use of this field is not well defined, but it is intended to provide a way to classify streams of packets into flows, which may then be treated as a group.
- **Fragments:** The Fragments field allows the selection of IPv6 packets, based on whether they are packet fragments and contain the fragmentation header (Next Header value 44).
- **Routing:** The Routing field allows the selection of packets that use the routing header (Next Header value 43).
- **Undetermined-Transport (unknown Next Header):** This field allows the selection of packets that do not contain the Next Header field that describes the Layer 4 protocol, as could happen in a long chain of IPv6 extension headers.

Cisco IOS IPv6 Header Filtering (Cont.)

Example (extended ACL filtering in IPv6 header values)



The figure shows a sample configuration of an ACL. Several parameters are given as an example of how an administrator can match on header information.

```

ipv6 access-list my-hdrcheck-list
deny 2001:db8:3::/64 any fragments
deny 2001:db8:3::/64 any dscp af12
deny 2001:db8:3::/64 any routing
deny 2001:db8:3::/64 any flow-label 1984
deny 2001:db8:3::/64 any mobility-type 2
permit ipv6 2001:db8:3::/64 any
  
```

Note

The IPv6 Flow Label field, 20 bits, is still under development. The present specification states that the source node chooses a random flow label to assign to unique flows, and that no priority or other characteristics can be assumed from the Flow Label value. Cisco allows an ACL to match a specific flow label, but it is unclear at this time whether that will support a real-world flow-label processing scheme.

ACLs and AH

ACLs can check the presence of an AH in two ways:

- Protocol matching

```
permit ahp 2001:db8:100::/48 2001:db8:200::/48
deny ahp any any
permit ipv6 any any
```

- Extension matching

```
permit tcp 2001:db8:100::/48 2001:db8:200::/48 eq www
permit tcp any 2001:db8:200::/48 eq 80 auth
deny any any auth
permit ipv6 any any
```

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-20

ACLs enable matching on the upper-layer protocol (ULP)—for example, TCP, User Datagram Protocol (UDP), ICMP, Stream Control Transmission Protocol (SCTP)—regardless of whether an Authentication Header (AH) is present or absent. Matching of an AH can be done in two ways.

Extension Matching of an AH

TCP or UDP traffic can be matched to the ULP if an AH is present or absent. Before this feature was introduced, this function was available only if an AH was absent.

This feature introduces the **auth** keyword to the **permit** and **deny** commands. The **auth** keyword allows the matching of traffic against the presence of the AH in combination with the specified protocol; that is, TCP or UDP.

```
ipv6 access-list EXAMPLE
permit tcp any host 2001:db8:200::51 eq www
permit tcp 2001:db8:100::/48 host 2001:db8:200::51 auth
deny tcp any host 2001:db8:200::51
permit ipv6 any any
```

Protocol Matching of an AH

IPv6 traffic can be matched to the ULP when an AH header is present. To perform this function, enter the **ahp** option for the protocol argument when using the **permit** or **deny** command.

```
ipv6 access-list EXAMPLE
permit ahp 2001:db8:100::/48 host 2001:db8:200::51
deny ahp any host 2001:db8:200::51
permit ahp 2001:db8::/32 any
permit ipv6 any any
```

Cisco IOS New ICMPv6 Types

This topic describes new Cisco IOS Software ICMPv6 types.

New Cisco IOS Software ICMPv6 Types

New ICMP types for IPv6 (ICMPv6) include:

- Error messages
- Information messages
- Multicast messages
- Route advertisement, neighbor solicitation, or neighbor advertisement messages
- Mobility (Mobile IPv6) messages

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--7-22

New ICMP version 6 (ICMPv6) types for IPv6 include the following:

- Error messages
- Information messages
- Multicast messages
- Route advertisement, neighbor solicitation, or neighbor advertisement messages
- Mobility (Mobile IPv6) messages

ICMPv6 has several ICMP messages that are used for error reporting and informational purposes. Extended ACLs can match on these values.

These ICMP messages are assigned by the Internet Assigned Numbers Authority (IANA):

Type	Name	Reference
-----	-----	-----
1	Destination Unreachable	[RFC2463]
2	Packet Too Big	[RFC2463]
3	Time Exceeded	[RFC2463]
4	Parameter Problem	[RFC2463]
100	Private experimentation	[RFC4443]
101	Private experimentation	[RFC4443]
127	Reserved for expansion of ICMPv6 error messages	[RFC4443]
128	Echo Request	[RFC2463]
129	Echo Reply	[RFC2463]
130	Multicast Listener Query	[RFC2710]
131	Multicast Listener Report	[RFC2710]
132	Multicast Listener Done	[RFC2710]
133	Router Solicitation	[RFC2461]
134	Router Advertisement	[RFC2461]
135	Neighbor Solicitation	[RFC2461]

136	Neighbor Advertisement	[RFC2461]
137	Redirect Message	[RFC2461]
138	Router Renumbering	[Crawford]
139	ICMP Node Information Query	[RFC-ietf-ipngwg-icmp-
name-lookups-15.txt]		
140	ICMP Node Information Response	[RFC-ietf-ipngwg-icmp-
name-lookups-15.txt]		
141	Inverse Neighbor Discovery Solicitation Message	[RFC3122]
142	Inverse Neighbor Discovery Advertisement Message	[RFC3122]
143	Version 2 Multicast Listener Report	[RFC3810]
144	Home Agent Address Discovery Request Message	[RFC3775]
145	Home Agent Address Discovery Reply Message	[RFC3775]
146	Mobile Prefix Solicitation	[RFC3775]
147	Mobile Prefix Advertisement	[RFC3775]
148	Certification Path Solicitation Message	[RFC3971]
149	Certification Path Advertisement Message	[RFC3971]
150	ICMP messages utilized by experimental mobility protocols such as Seamoby	[RFC4065]
151	Multicast Router Advertisement	[RFC4286]
152	Multicast Router Solicitation	[RFC4286]
153	Multicast Router Termination	[RFC4286]

New Cisco IOS Software ICMPv6 Types (Cont.)

- Neighbor discovery and ICMP
 - ND messages provide mapping between Layer 3 and Layer 2 addresses (in IPv4, ARP does this).
 - In contrast with ARP, ICMP can be blocked by ACLs.
 - Never block ND messages on local segments.
- Implicit deny
 - Implicit deny has been changed to the following sequence:

```
permit icmp any any nd-na
permit icmp any any nd-ns
deny ipv6 any any
```

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-23

In IPv6, the neighbor discovery protocol has taken from Address Resolution Protocol (ARP) the job of mapping between Layer 3 and Layer 2 addresses. The protocol uses ICMP to transmit messages and can be blocked by ACLs.

As you learned, if IPv4 has an implicit default deny all at the end of every ACL, then IPv6 has these three implicit statements:

```
permit icmp any any nd-na
permit icmp any any nd-ns
deny ipv6 any any
```

The two statements preceding the deny statement are unique to IPv6. These statements allow link-local neighbor advertisements and neighbor solicitation to work between the router and any link-local nodes; they are needed for proper functioning of on-link messages. The ACL must allow on-link functions that relate to neighbor discovery.

Note that if you put an explicit deny at the end, you will block all neighbor discovery packets, unless they are explicitly permitted. Doing so will result in your router being unable to receive ND solicitations and ND advertisements.

Editing of ACLs

This topic describes how to edit IPv6 ACLs.

Editing of ACLs

- Inserting new entries
- Removing existing entries

```
deny ipv6 2001:db8:10::/48 any log sequence 10
permit icmp any 2001:db8:10::/48 echo-reply sequence 20
permit icmp any 2001:db8:10::/48 echo-request sequence 30
permit icmp any 2001:db8:10::/48 hop-limit sequence 40
permit icmp any 2001:db8:10::/48 no-admin sequence 50
permit icmp any 2001:db8:10::/48 no-route sequence 60
permit icmp any 2001:db8:10::/48 time-exceeded sequence 70
permit icmp any 2001:db8:10::/48 unreachable sequence 80
```

no sequence 30

```
permit icmp any 2001:db8:10::/48 packet-too-big sequence 65
```

- No resequence command

© 2010 Cisco Systems, Inc. All rights reserved. IPv6FD v3.0—7-25

Like IPv4 named ACLs, IPv6 ACLs can be edited in place by adding ACL entries at the end or by inserting them in the middle; in this case, a sequence number must be specified. Any entries with sequence numbers that differ from the default sequencing (spaced by 10) are stored in the configuration, with the sequence number appended at the end.

Although insertion and removal of individual entries presents no problem, lack of a **resequence** command might prevent the administrator from making an arbitrary number of changes to an ACL before being forced to rewrite it completely.

IPv6 ACL Tips

- Do not block all ICMP.
 - Packet-too-big (PMTUD)
 - Neighbor discovery
- Careful sequencing of ACL entries.
- Be careful with zeroes in address specifications.
$$\text{FF}::/8 \neq \text{FF00}::/8$$
- General (protocol-independent) guidelines:
 - Put more-specific entries at the top.
 - Remember the implicit deny at the end.
 - Reference to an empty or non-existing ACL on an interface permits all traffic.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-26

There are a couple of tips for writing IPv6 ACLs, because of the additions to the protocol.

Fragmentation is not used in IPv6 on transit devices, so the senders must know the correct path maximum transmission unit (MTU). The best way to do so is by using the path MTU discovery (PMTUD) technique that is used for dynamically discovering the MTU. Do not block packet-too-big messages, unless you use the minimum guaranteed MTU, which is 1280 bytes.

Do not block neighbor discovery packets. Doing so would have the same effect as blocking ARP in IPv4.

Be careful when writing IP addresses. IPv6 addresses can be presented in many forms. $\text{FF}::/8$ is not the same as $\text{FF00}::/8$.

Other guidelines from IPv4 still apply:

- Put more-specific entries at the top.
- There is an implicit deny at the end of the list.
- Referencing a nonexistent IPv6 ACL permits all traffic.

How to Configure ACLs in an IPv6 Environment

This topic describes how to configure ACLs in an IPv6 environment.

How to Configure ACLs in an IPv6 Environment

ACL configuration procedure:

1. Design the traffic flows.
2. Examine the interfaces.
3. Create the ACLs.
4. Build and apply ACLs.
5. Test the ACLs.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-28

There are many ways to build an ACL. For example, the ACL can be put on different interfaces. If an ACL is put in different directions, outgoing traffic can be blocked (for example, ICMPv6 echo requests) or the return traffic can be blocked (for example, ICMPv6 echo replies). Either of these restrictions disables user ping.

The following procedure can be used when configuring ACLs.

Step 1 Design the traffic flows.

Decide which traffic you want to control, to permit, and to deny. Be specific: Consider all the parameters (for example, source IP, destination IP, protocols, ports, headers, and so on) that you can match packets against with an extended ACL. Consider using a reflexive ACL and limiting access by time.

Step 2 Examine the interfaces.

Decide to which interfaces you will apply the ACL, and in which direction.

Step 3 Create the ACLs.

Create the ACLs so that you have, at most, two ACLs for each interface: one inbound and one outbound. Try to design the ACLs so that you need only one per interface. Doing so will make the ACLs simpler and easier to troubleshoot and maintain.

Step 4 Build and apply the ACLs.

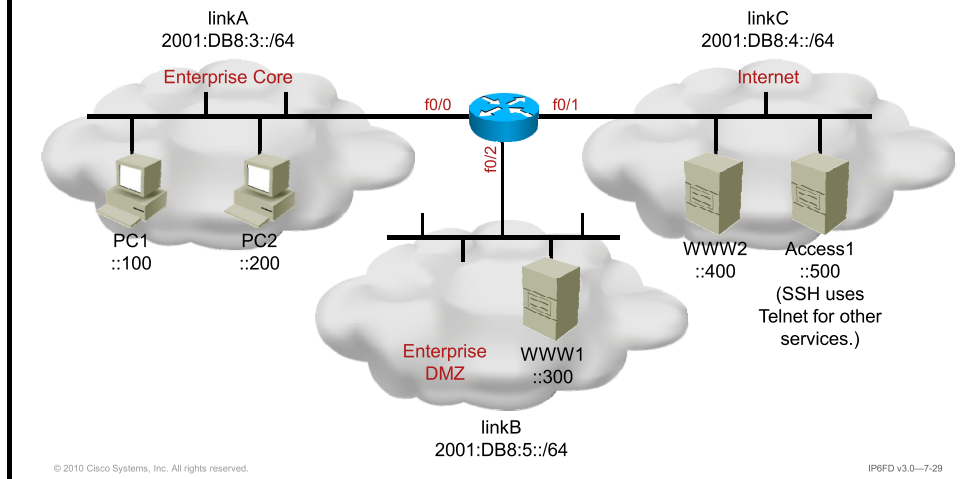
Apply the ACLs appropriately to the interfaces for standard, extended, reflexive, and time-based ACLs.

Step 5 Test the ACLs.

Verify that the ACLs are correct by reviewing the configuration, by using the **show ipv6 access-list** command.

How to Configure ACLs in an IPv6 Environment (Cont.)

Example



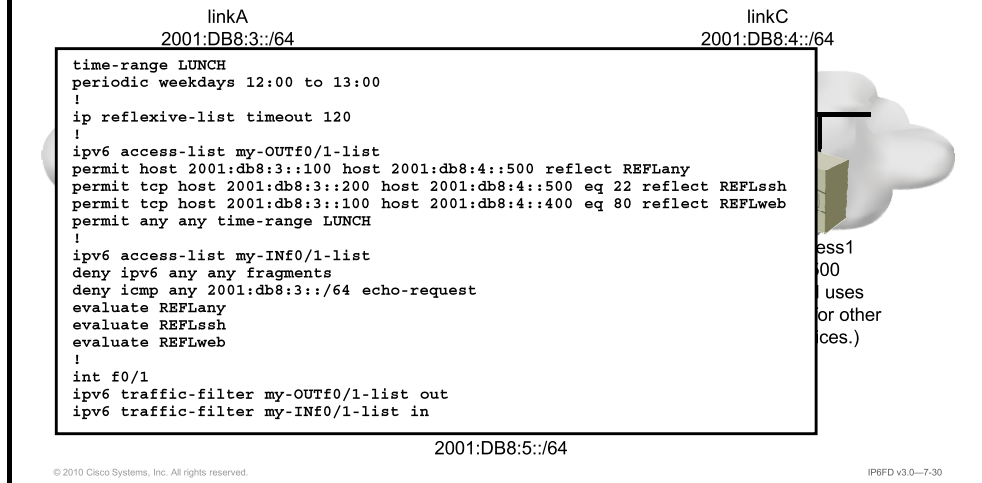
The example topology illustrates an ACL configuration and assumes that you want to apply the following restrictions on traffic:

- Permit PC1 connection to Access1 (apply outbound on Fast Ethernet 0/1).
- Permit PC2 connection to Access1, for Secure Shell (SSH; apply outbound on Fast Ethernet 0/1).
- Permit PC1 connection to WWW2, for HTTP, reflexive (apply outbound on Fast Ethernet 0/1).
- Permit any connection to any, only at lunchtime (apply outbound on Fast Ethernet 0/1).
- Deny any connection to linkA, for ICMPv6 echo request (apply inbound on Fast Ethernet 0/1).
- Deny any connection to linkA or linkB, for fragmented packets (apply inbound on Fast Ethernet 0/1).

Note The steps that are shown do not match one-to-one with the bullet points. ACLs are not configured in that manner.

How to Configure ACLs in an IPv6 Environment (Cont.)

Example



This figure shows an output of the sample configuration.

```
time-range LUNCH
periodic weekdays 12:00 to 13:00
!
ip reflexive-list timeout 120
!
ipv6 access-list my-OUTf01-list
permit host 2001:db8:3::100 host 2001:db8:4::500 reflect REFLany
permit tcp host 2001:db8:3::200 host 2001:db8:4::500 eq 22 reflect REFLssh
permit tcp host 2001:db8:3::100 host 2001:db8:4::400 eq 80 reflect REFLweb
permit any any time-range LUNCH
!
ipv6 access-list my-INF01-list
deny ipv6 any any fragments
deny icmp any 2001:db8:3::/64 echo-request
evaluate REFLany
evaluate REFLssh
evaluate REFLweb
!
int f0/1
ipv6 traffic-filter my-OUTf01-list out
ipv6 traffic-filter my-INF01-list in
```

Summary

This topic summarizes the key points that were discussed in this lesson.

Summary

- Standard ACLs examine only source and destination addresses; extended ACLs allow packet matching on more IPv6 header fields.
- Reflexive ACLs allow traffic flow control, based on session initiation; time-based ACLs allow control, based on time ranges.
- Extended ACLs can examine new IPv6 header fields.
- Configuring ACLs involves designing traffic flow; examining the interfaces; creating, building, and applying the ACLs; and testing the ACLs.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-31

Using IPsec, IKE, and VPNs

Overview

Developed to meet unanticipated IP version 4 (IPv4) security requirements, IP Security (IPsec) is the baseline protocol that is established for enabling packet security during transmission. IP version 6 (IPv6) implementations mandate support for IPsec. This change will ultimately redefine how security policies are developed in the future.

This lesson describes IPsec and related topics in IPv6.

Objectives

Upon completing this lesson, you will be able to describe how security is implemented in IPv6. This ability includes being able to meet these objectives:

- Describe how security is implemented in IPv6
- Configure IPsec and IKE in Cisco IOS routers
- Describe and configure various VPN solutions, including IPsec VPN, SSL VPN, and DMVPN

IPsec, IKE, and VPNs Basics

This topic describes basic concepts of using IPsec, Internet Key Exchange (IKE), and virtual private networks (VPNs).

Using IPsec, IKE, and VPNs

- IPsec headers
 - AH
 - ESP
- IPsec and IKE
 - IKE is used for key exchange
 - Keys are used for IPsec encryption
- IPv6 and VPNs
 - Site-to-site VPN
 - SSL VPN
 - DMVPN (multipoint GRE with IPsec)

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-3

IPsec was developed as a vital part of IPv6, to provide data integrity and confidentiality during transport. Two extension headers can be used: Authentication Header (AH), which provides data integrity, and Encapsulating Security Payload (ESP), which also provides confidentiality.

IPsec is mandatory in IPv6, so any IPv6 network stack must know how to manage IPsec. IPsec uses symmetric keys to perform encryption and verify integrity. However, key exchange is not part of the IPsec specification, and end systems must rely on other mechanisms to securely exchange keys. The simplest way is to configure a static key on all participants in secure data exchange, but this option might not be possible to do in a secure fashion.

Internet Key Exchange (IKE) uses the Diffie-Hellman (DH) algorithm to compute keys on both sides, over an insecure channel. However, IKE must be implemented separately from the IPv6 network stack and might not be available on all end systems.

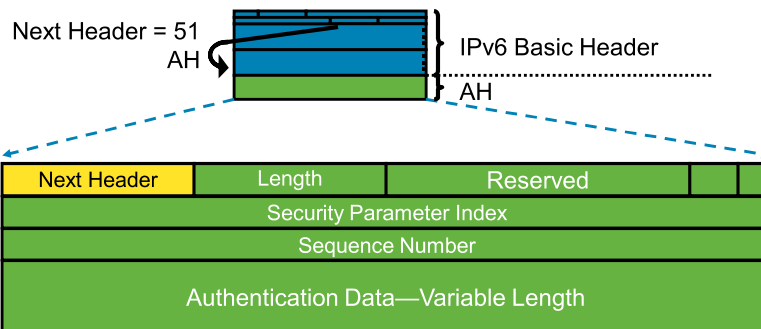
Because DH is vulnerable to man-in-the-middle attacks, additional mechanisms must be used to authenticate peers. Cisco IOS Software supports certificates and pre-shared keys (PSKs).

IPv6 on Cisco IOS Software is supported over site-to-site IPsec VPNs and Dynamic Multipoint Virtual Private Network (DMVPNs), and also on Secure Sockets Layer (SSL)-based VPNs, which do not use IPsec.

Security: IPsec AH

IPsec AH:

- Integrity
- Authentication of the source
- Optional replay protection



Security inside IPv6 is achieved by IPsec protocols. IPsec is available in both IPv4 and IPv6 but is mandatory in IPv6. IPsec has two headers that can be chained in the same IPv6 packet by using extension headers. There are many related protocols for IPsec, such as the key management protocols.

AH provides integrity and authentication of the source. AH also provides optional protection against replays. AH protects the integrity of most but not all IP header fields; some fields change over the path. AH authenticates the source through a signature-based algorithm.

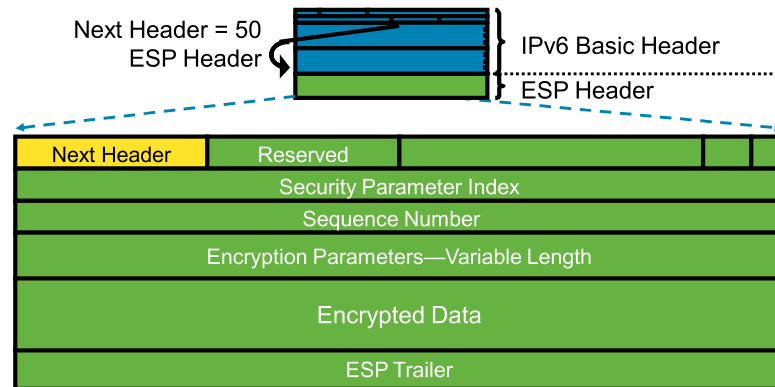
The primary security-related difference between IPv4 and IPv6 is that IPsec is mandatory for IPv6. Therefore, all IP communications can be secured, if there is enough keying infrastructure to do so on a large scale.

IPsec is documented by a series of RFCs. The base specification is RFC 4301, *Security Architecture for the Internet Protocol*. IPsec AH is documented in RFC 4302, *IP Authentication Header*.

Tip These RFCs (dated December 2005) replace the previous generation of IPsec specifications.

Security: IPsec ESP

IPsec ESP adds confidentiality.



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7.5

The other IPsec security header is the ESP header. ESP provides confidentiality, authentication of the source, connectionless integrity of the inner packet, antireplay, and limited traffic flow confidentiality.

IPsec ESP is documented in RFC 4303, *IP Encapsulating Security Payload (ESP)*.

IPsec and IKE

This topic describes IPsec and IKE with IPv6.

IKE

- IKE is the mechanism that IPsec nodes use to change keys periodically, to defeat eavesdropping or sampling attacks.
- Keys must be securely changed before sequence number cycles, to use anti-replay service.
- Small key lengths and fast networks make frequent key rollover important.
 - IPsec now supports extended sequence numbers, which are 64-bits wide.
- Most IPsec IPv6 stacks do not include IKE support, so manual keys must be used and managed.
- Key infrastructure still lacks wide deployment, so any IPsec implementation will be difficult to scale.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-7

The Internet Security Association and Key Management Protocol (ISAKMP) combines authentication, key management, and security association (SA) management to build secure communication channels between hosts.

IKE can be used to negotiate SAs.

Regardless of the setup of the initial security association, even if symmetric PSKs are used, a shared key must be changed before sequence numbers (also called the initialization vector) are reused.

For IPsec ESP, the sequence number is a 32-bit value, so a shared key can be used securely for no more than 4 billion packets before the key should be changed. IKE provides a mechanism for securely changing the shared key.

Most IPsec implementations for IPv4 support IKE. Most implementations for IPv6 do not support IKE. Static keys must be used and changed manually, which is rarely done. Thus, for IPv6, shared keys often remain constant for too long, resulting in a security risk.

A new version of IKE, IKE version 2 (IKEv2) was recently standardized. IKEv2 is specified in RFC 4306, *Internet Key Exchange (IKEv2) Protocol*.

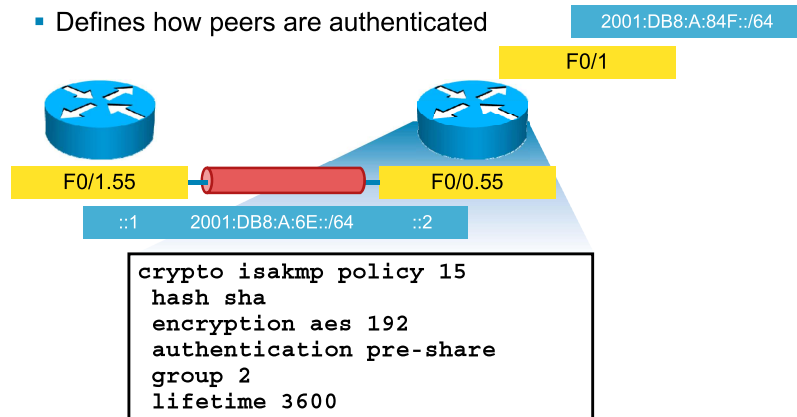
Tip

IKE is not mandated by the IPv6 specification RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*. IKE is a supporting protocol that makes IPsec more secure and more scalable but is optional for IPv6 vendors.

IPsec and IKE Configuration

Step 1: Configure the IKE policy.

- Defines how IKE traffic is encrypted
- Defines how peers are authenticated



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-8

Follow these steps to secure IPv6 traffic between two Cisco routers:

Step 1 Configure the IKE policy.

Implementing IKE and IPsec on Cisco IOS routers is a multistep process. First, create an ISAKMP policy that specifies the ISAKMP and IKE parameters to protect the IKE exchanges, which are where IPsec keys and other IPsec-related parameters are negotiated.

The figure shows an example of configuring the IKE policy, using these commands:

■ **crypto isakmp policy 15**

This command creates an ISAKMP policy with a priority number. Peers will negotiate to find a common ISAKMP policy, preferring lower-priority combinations to higher values. This example uses the same value on both routers.

■ **hash sha**

This command specifies the Secure Hash Algorithm 1 (SHA-1) for integrity checking.

■ **encryption aes 192**

This command specifies Advanced Encryption Standard (AES)-192 as the encryption algorithm to protect the ISAKMP negotiations.

■ **authentication pre-share**

This command specifies that a PSK will be used to establish the initial trust relationship between the two IKE peers. Certificates are sometimes used but are not supported by Cisco for IPv6.

■ **group 2**

This command specifies DH Group 2 (DH2).

■ **lifetime 3600**

This command specifies an IKE SA lifetime of 60 minutes. IKE will negotiate a new SA for its own use after this much time has elapsed.

Note IKE has several important features for enhancing IPv6 security. One primary advantage of IKE is scalability: It is easier to set up one PSK between two IKE peers than to manually configure numerous IPsec SAs. Perhaps more importantly, IKE provides key rollover services, periodically changing the IPsec keys. There are known eavesdropping attacks on IPsec networks that are entirely passive and they are able to derive secret keys if enough packets can be intercepted. Periodic rollover of the keys blunts this type of passive attack.

IPsec and IKE Configuration (Cont.)

Step 2: Configure the IKE key.

- Authenticates devices
- Must be identical on both sides
- Possible but not recommended to use the same key for more than one pair of nodes

```
crypto isakmp key GreenCableRE address ipv6 2001:DB8:A:6E::1/128
```

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-9

Step 2 Configure the IKE key.

Each IKE or ISAKMP (IKE uses the ISAKMP framework) peer must be defined with a shared secret to establish initial trust. In the previous step, the authentication mechanism “shared-secret” was chosen; now the key itself is specified.

```
crypto isakmp key GreenCableRE address ipv6 2001:DB8:A:6E::1/128
```

This command establishes a shared secret string, GreenCableRE, which should be passed to the IKE peer 2001:DB8:A:6E::1/128 when the initial IKE bootstrapping process is attempted. On the peer router, the address that is used must be the address of this router, the peer for the other router.

IPsec and IKE Configuration (Cont.)

Step 3: Configure the IPsec transform set and profile.

- Transform set defines how IPsec traffic is encrypted.
- Profile is used to attach transform set to interfaces.

```
crypto ipsec transform-set MY-TRANSFORM-SET esp-aes esp-sha-hmac
!
crypto ipsec profile MY-PROFILE-NAME
set transform-set MY-TRANSFORM-SET
```

© 2010 Cisco Systems, Inc. All rights reserved.

IPsec v3.0—7-10

Step 3 Configure the IPsec transform set and profile.

Now that the IKE configuration is complete, IPsec must also be configured. The first step is to specify the acceptable IPsec parameters. Multiple transform sets may be created, and the IKE negotiation will seek a common set of parameters between the two IKE peers.

```
crypto IPsec transform-set MY-TRANSFORM-SET esp-aes esp-sha-hmac
crypto IPsec profile MY-PROFILE-NAME
set transform-set MY-TRANSFORM-SET
```

Here, a single transform set has been configured, one that uses AES for encryption and SHA-1 for authentication and integrity protections. Alternatively, Data Encryption Standard (DES), Triple DES (3DES), Null, and Message Digest 5 (MD5) algorithms might be available.

IPsec and IKE Configuration (Cont.)

Step 4: Configure the VTI.

- Functions as a regular interface
- Applies IPsec profile settings to all traffic forwarded through it

```
interface Tunnel355
no ip address
ipv6 address FE80::2 link-local
ipv6 address 2001:DB8:A:FF4::2/64
tunnel source FastEthernet0/0.55
tunnel destination 2001:DB8:A:6E::1
tunnel mode ipsec ipv6
tunnel protection ipsec profile MY-PROFILE-NAME
```

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--7.11

Step 4 Configure the virtual tunnel interfaces (VTIs).

Cisco uses a VTI for IPsec tunnels, and this IPsec configuration uses tunnel mode—not transport mode, as host-to-host IPsec sometimes specifies. Many of these commands should look familiar to someone who has built IPv6 transition mechanism tunnels.

```
interface Tunnel355
no ip address
ipv6 address FE80::2 link-local
ipv6 address 2001:DB8:A:FF4::2/64
```

These commands define the tunnel interface number, which is arbitrary. They also assign the IPv6 link-local and global-scope addresses for the interface. Even though these interfaces are tunnel interfaces, they must follow the IPv6 specification. If the link-local address is not manually set, it will be constructed by the router.

```
tunnel source FastEthernet0/0.55
```

Set the tunnel source here by specifying the IPv6 interface on which the tunnel interface is built. The IPv6 address can also be specified here. In this case, the Fast Ethernet interface has the IPv6 address 2001:DB8:A:6E::2.

```
tunnel destination 2001:DB8:A:6E::1
```

The tunnel destination must also be set. In this example, the two IPsec peer routers are on the same subnet, but that does not need to be the case. Enter the IPv6 address of the peer IKE or IPsec router.

```
routerA(config-if)# tunnel mode IPsec ipv6
```

This specifies the encapsulation mode for the tunnel. This tunnel is IPsec inside IPv6.

```
tunnel protection IPsec profile MY-PROFILE-NAME
```

The command that is shown here specifies the IPsec profile to be used on this IPsec session. The profile was specified earlier and uses AES and SHA-1 as acceptable transforms.

IPsec and IKE Configuration (Cont.)

Step 5: Route traffic into the IPsec tunnel.

- Redirects traffic through the tunnel
- Dynamic routing can be used instead of static

```
ipv6 route ::/0 tunnel 355
```

Route verification:

```
routerA# show ipv6 route static
IPv6 Routing Table - 11 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGPZ
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S      ::/0 [1/0]
       via ::, Tunnel355
```

© 2010 Cisco Systems, Inc. All rights reserved.

IPsec v3.0-7.12

Step 5 Route traffic into the IPsec tunnel.

No traffic will traverse the IPsec tunnel until it is routed there, either via static routes or by enabling a routing protocol on the IPsec tunnel interface. In this example, a static route is used.

```
ipv6 route ::/0 tunnel 355
```

```
show ipv6 route static
```

```
IPv6 Routing Table - 11 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, IS - ISIS summary
```

```
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
```

```
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
```

```
S      ::/0 [1/0]
```

```
       via ::, Tunnel355
```

This command sends all IPv6 traffic that leaves the router across the IPsec-protected tunnel. Traffic between IPv6-capable VLANs that are attached to the local router will not use IPsec, but any traffic that is headed towards the default route will use IPsec.

IPsec and IKE Configuration (Cont.)

Step 6: Test and troubleshoot (1).

```
routerA# show crypto engine connections active
```

Crypto Engine Connections						
ID	Interface	Type	Algorithm	Encrypt	Decrypt	IP-Address
11	Fa0/0.55	IPsec	AES+SHA	0	38	2001:DB8:A:6E::2
12	Fa0/0.55	IPsec	AES+SHA	114	0	2001:DB8:A:6E::2
1124	Fa0/0.55	IKE	SHA+AES192	0	0	2001:DB8:A:6E::2

```
routerA# show crypto isakmp peer
```

```
Peer: 2001:DB8:A:6E::1 Port: 500 Local: 2001:DB8:A:6E::2  
Phase1 id: 2001:DB8:A:6E::1
```

© 2010 Cisco Systems, Inc. All rights reserved.IPv6FD v3.0--7-13

Step 6 Test and troubleshoot.

Several crypto-related **show** commands for Cisco IOS routers can help with troubleshooting and monitoring. The **show crypto engine connections active** command shows active crypto sessions, including IKE and IPsec.

```
show crypto engine connections active
```

Crypto Engine Connections						
ID	Interface	Type	Algorithm	Encrypt	Decrypt	IP-Address
11	Fa0/0.55	IPsec	AES+SHA	0	38	2001:DB8:A:6E::2
12	Fa0/0.55	IPsec	AES+SHA	114	0	2001:DB8:A:6E::2
1124	Fa0/0.55	IKE	SHA+AES192	0	0	2001:DB8:A:6E::2

The **show crypto isakmp peer** command lists IKE peers. This router has a single IKE relationship with the peer at 2001:DB8:A:6E::1, which is the peer that is configured in the example.

```
show crypto isakmp peer
```

```
Peer: 2001:DB8:A:6E::1 Port: 500 Local: 2001:DB8:A:6E::2  
Phase1 id: 2001:DB8:A:6E::1
```

IPsec and IKE Configuration (Cont.)

Step 6: Test and troubleshoot (2).

```
routerA# show crypto isakmp sa

IPv4 Crypto ISAKMP SA
dst          src          state          conn-id slot status
IPv6 Crypto ISAKMP SA
dst: 2001:DB8:A:6E::2
src: 2001:DB8:A:6E::1
state: QM IDLE      conn-id: 1124 slot: 0 status: ACTIVE
```

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-14

The **show crypto isakmp sa** command shows the current ISAKMP and IKE SAs. IKE SAs are bidirectional, and the one in this sample is IDLE, indicating that IKE is quiet at the moment. If the command is run during the IKE renegotiation, that will show in the status field.

```
show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst          src          state          conn-id slot status
IPv6 Crypto ISAKMP SA
dst: 2001:DB8:A:6E::2
src: 2001:DB8:A:6E::1
state: QM IDLE      conn-id: 1124 slot: 0 status: ACTIVE
```

IPsec and IKE Configuration (Cont.)

Step 6: Test and troubleshoot (3).

```
routerA# show crypto ipsec sa

interface: Tunnel355
  Crypto map tag: Tunnel355-head-0, local addr 2001:DB8:A:6E::2

  protected vrf: (none)
  local ident (addr/mask/prot/port): (::/0/0/0)
  remote ident (addr/mask/prot/port): (::/0/0/0)
  current peer 2001:DB8:A:6E::1 port 500
    PERMIT, flags={origin is acl,}
    #pkts encaps: 162, #pkts encrypt: 162, #pkts digest: 162
    #pkts decaps: 53, #pkts decrypt: 53, #pkts verify: 53
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 18, #recv errors 0
```

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--7.15

This **show** command provides information about the current IPsec SAs. Note that these SAs are unidirectional.

show crypto IPsec sa

```
interface: Tunnel355
  Crypto map tag: Tunnel355-head-0, local addr 2001:DB8:A:6E::2

  protected vrf: (none)
  local ident (addr/mask/prot/port): (::/0/0/0)
  remote ident (addr/mask/prot/port): (::/0/0/0)
  current peer 2001:DB8:A:6E::1 port 500
    PERMIT, flags={origin is acl,}
    #pkts encaps: 162, #pkts encrypt: 162, #pkts digest: 162
    #pkts decaps: 53, #pkts decrypt: 53, #pkts verify: 53
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 18, #recv errors 0

  local crypto endpt.: 2001:DB8:A:6E::2,
  remote crypto endpt.: 2001:DB8:A:6E::1
  path mtu 1514, ip mtu 1514
  current outbound spi: 0xFF23541D(4280505373)

inbound esp sas:
  spi: 0x7300512E(1929400622)
    transform: esp-aes esp-sha-hmac ,
    in use settings = {Tunnel, }
    conn id: 11, flow id: SW:11, crypto map: Tunnel355-head-0
    sa timing: remaining key lifetime (k/sec): (4381693/1726)
    IV size: 16 bytes
    replay detection support: Y
    Status: ACTIVE

inbound ah sas:

inbound pcg sas:
```

```
outbound esp sas:
spi: 0xFF23541D(4280505373)
(output trimmed)
```

IPsec and IKE Configuration (Cont.)

Step 6: Test and troubleshoot (4).

```
routerA# show debug
Cryptographic Subsystem:
  Crypto ISAKMP debugging is on
Jul 12 21:56: %CRYPTO-6-ISAKMP ON OFF: ISAKMP is ON
Jul 12 21:56: ISAKMP:(0): SA request profile is (NULL)
Jul 12 21:56: ISAKMP: Created a peer struct for 2001:DB8:A:6E::1, peer port 500
Jul 12 21:56: ISAKMP: New peer created peer = 0x865020E8 peer handle = 0x8000087B
```

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0-7-16

This **debug** command shows that much traffic is exchanged during IKE negotiation, including a comparison of available transforms. In the example, both routers are configured with a single transform; because they match, negotiation at this stage is successful.

show debug

Cryptographic Subsystem:

Crypto ISAKMP debugging is on

```
Jul 12 21:56:34.689: %CRYPTO-6-ISAKMP ON OFF: ISAKMP is ON
Jul 12 21:56:34.697: ISAKMP:(0): SA request profile is (NULL)
Jul 12 21:56:34.697: ISAKMP: Created a peer struct for 2001:DB8:A:6E::1, peer
port 500
Jul 12 21:56:34.697: ISAKMP: New peer created peer = 0x865020E8 peer handle =
0x8000087B
Jul 12 21:56:34.697: ISAKMP: Locking peer struct 0x865020E8, refcount 1 for
isakmp initiator
Jul 12 21:56:34.697: ISAKMP: local port 500, remote port 500
Jul 12 21:56:34.697: ISAKMP: set new node 0 to QM IDLE
Jul 12 21:56:34.701: insert sa successfully sa = 866CB870
Jul 12 21:56:34.701: ISAKMP:(0):Can not start Aggressive mode, trying Main
mode.
Jul 12 21:56:34.701: ISAKMP:(0):found peer pre-shared key matching
2001:DB8:A:6E::1
Jul 12 21:56:34.701: ISAKMP:(0):Input = IKE MESG FROM IPSEC, IKE SA REQ MM
Jul 12 21:56:34.705: ISAKMP:(0):Old State = IKE READY New State = IKE I MM1

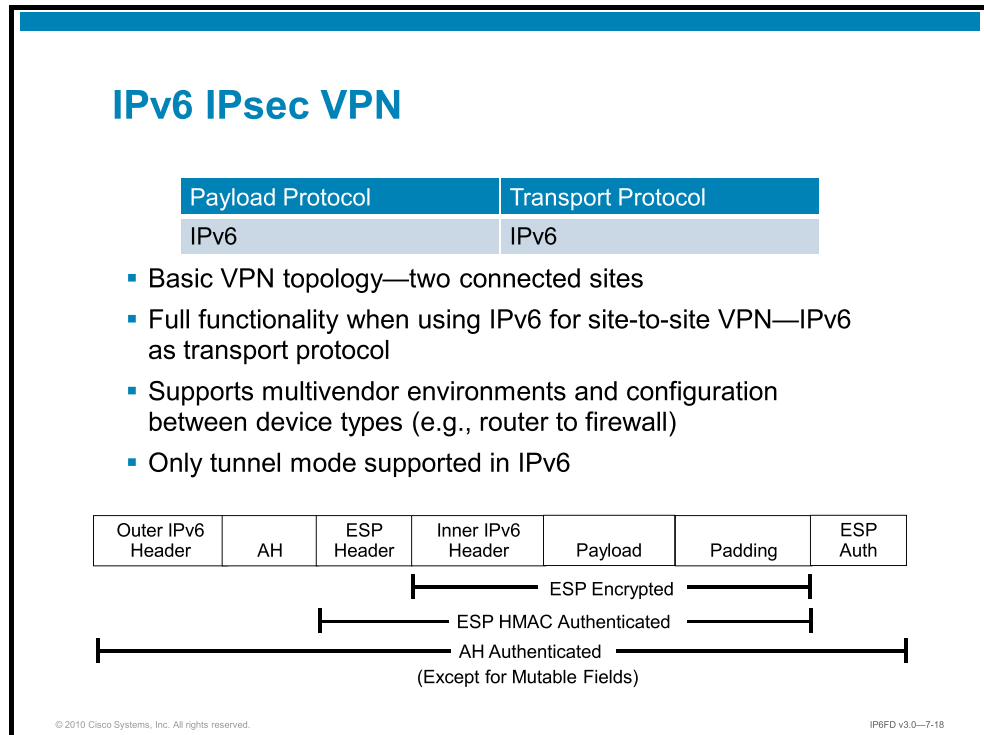
Jul 12 21:56:34.705: ISAKMP:(0): beginning Main Mode exchange
Jul 12 21:56:34.705: ISAKMP:(0): sending packet to 2001:DB8:A:6E::1 my port
500 peer port 500 (I) MM NO STATE
Jul 12 21:56:34.717: ISAKMP (0:0): received packet from 2001:DB8:A:6E::1 dport
500 sport 500 Global (I) MM NO STATE
Jul 12 21:56:34.721: ISAKMP:(0):Input = IKE MESG FROM PEER, IKE MM EXCH
Jul 12 21:56:34.721: ISAKMP:(0):Old State = IKE I MM1 New State = IKE I MM2

Jul 12 21:56:34.721: ISAKMP:(0): processing SA payload. message ID = 0
```

```
Jul 12 21:56:34.725: ISAKMP:(0):found peer pre-shared key matching
2001:DB8:A:6E::1
Jul 12 21:56:34.725: ISAKMP:(0): local preshared key found
Jul 12 21:56:34.725: ISAKMP : Scanning profiles for xauth ...
Jul 12 21:56:34.725: ISAKMP:(0):Checking ISAKMP transform 1 against priority
15 policy
Jul 12 21:56:34.725: ISAKMP:      encryption AES-CBC
Jul 12 21:56:34.725: ISAKMP:      keylength of 192
Jul 12 21:56:34.725: ISAKMP:      hash SHA
Jul 12 21:56:34.725: ISAKMP:      default group 2
Jul 12 21:56:34.729: ISAKMP:      auth pre-share
Jul 12 21:56:34.729: ISAKMP:      life type in seconds
Jul 12 21:56:34.729: ISAKMP:      life duration (basic) of 3600
Jul 12 21:56:34.729: ISAKMP:(0):atts are acceptable. Next payload is 0
```

VPN Connections Using IPv6

This topic explains the various VPN connections that can be made to carry or use the IPv6 protocol.



Site-to-site VPN is the most basic VPN topology that can be implemented. The topology consists of two sites with a protected link. The link is protected by IPsec algorithms, using ESP for encryption. Encryption key exchange is accomplished by using either IKE or manual key entry.

Depending on router interface type, the protected link (tunnel) can carry unicast or multicast traffic. If encryption is configured on a physical interface by using the **crypto map** command, then only unicast traffic is carried. If encryption is configured on a tunnel interface (such as Generic Routing Encapsulation [GRE], IPv6-IP, or a VTI), then multicast traffic is carried as well. This allows for routing protocols to work over a protected link.

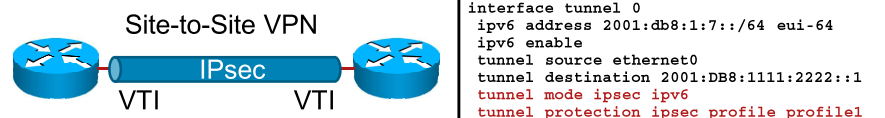
Since a pure site-to-site IPsec VPN is a completely standard technology, tunnels will work between devices of different vendors or types; for example, a tunnel between a firewall and a router.

Note Only site-to-site tunnel mode is supported in IPv6. You cannot use transport mode; that is, you cannot reuse existing original headers for the traffic. Traffic must be encapsulated.

IPv6 IPsec VPN (Cont.)

Configuration using a VTI:

- Most practical implementation; the interface carries unicast and multicast traffic.
- Possible use of routing protocols across the VTI.
- VTI interface appears in the routing table.
- An IPv6 VTI cannot use IPv4 as transport, and an IPv4 VTI cannot transport IPv6 packets.
- Functionally equivalent to crypto map configuration.



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-19

The IPsec VTI provides site-to-site IPv6 crypto protection of IPv6 traffic. Native IPv6 IPsec encapsulation is used to protect all types of IPv6 unicast and multicast traffic. The IPsec VTI allows IPv6 routers to establish IPsec tunnels between other security gateway routers and to provide crypto IPsec protection for traffic from internal networks. This functionality is like the security gateway model that uses IPv4 IPsec protection.

When the IPsec tunnel is configured, IKE and IPsec SAs are negotiated and set up before the line protocol for the tunnel interface is changed to the UP state. The remote IKE peer is the same as the tunnel destination address; the local IKE peer will be the address that is picked from the tunnel source interface that has the same IPv6 address scope as the tunnel destination address.

The configuration example configures a standard site-to-site IPsec VPN that uses a VTI. The **tunnel protection** command associates the tunnel interface with an IPsec profile. After an IPsec connection has been made (based on the protection profile), a virtual interface is spawned and is available for forwarding.

Note Only the **ipsec ipv6** configuration keywords are supported for IPv6.

IPv6 IPsec VPN (Cont.)

Configuration using IPsec-protected GRE tunnel:

- GRE can carry arbitrary Layer 3 protocols simultaneously.
- Possible use of routing protocols across the tunnel interface.
- Tunnel interface appears in the routing table.
- Can be used to transport IPv6 traffic over IPv4 network.
- Adds GRE overhead (IP header + GRE header).



```
interface tunnel 0
ipv6 address 2001:db8:1:7::/64 eui-64
ipv6 enable
tunnel source ethernet0
tunnel destination 2001:DB8:1111:2222::1
tunnel mode gre
tunnel protection ipsec profile profile1
```

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-20

If you want to carry multiple Layer 3 protocols over the same tunnel, you can use GRE encapsulation. GRE adds an additional 4-byte header, in which it specifies the tunneled protocol.

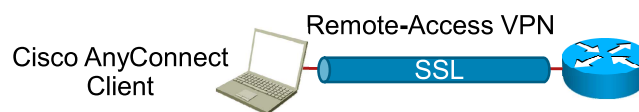
GRE also allows IPv6 IPsec tunnels over an IPv4 network.

With tunnel-mode, IPsec overhead is increased by 4 bytes of GRE, plus the additional IP header (which may be IPv4 or IPv6, depending on the transport protocol).

IPv6 SSL VPN

Payload Protocol	Transport Protocol
IPv6	SSL over TCP over IPv4

- Used to bypass firewalls that do not allow IPsec sessions.
- More stable remote-access VPN service.
- SSL is used for transport of IPv6 payload.
- SSL is TCP-based and works over firewalls or routers doing NAT.
- Session established using Cisco AnyConnect Client software available for 32-bit and 64-bit desktop operating systems.
- License-based feature on the Cisco ASA routers and ISRs.



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-21

The SSL VPN is an option to enable secure communications over an SSL session, as used for secure web browsing. The problem of some IPsec VPN implementations is that the connections were often blocked by firewalls in the traffic path from the user to the main site. SSL VPN appears to those firewalls as an SSL session.

The transported (payload) protocol in this case can be either IPv6 or IPv4, but an IPv4 network is used for transport. IPv6 is encapsulated into SSL, which uses TCP on the transport layer and IPv4 on the network layer. This approach has its advantages: An SSL VPN can run on most existing networks.

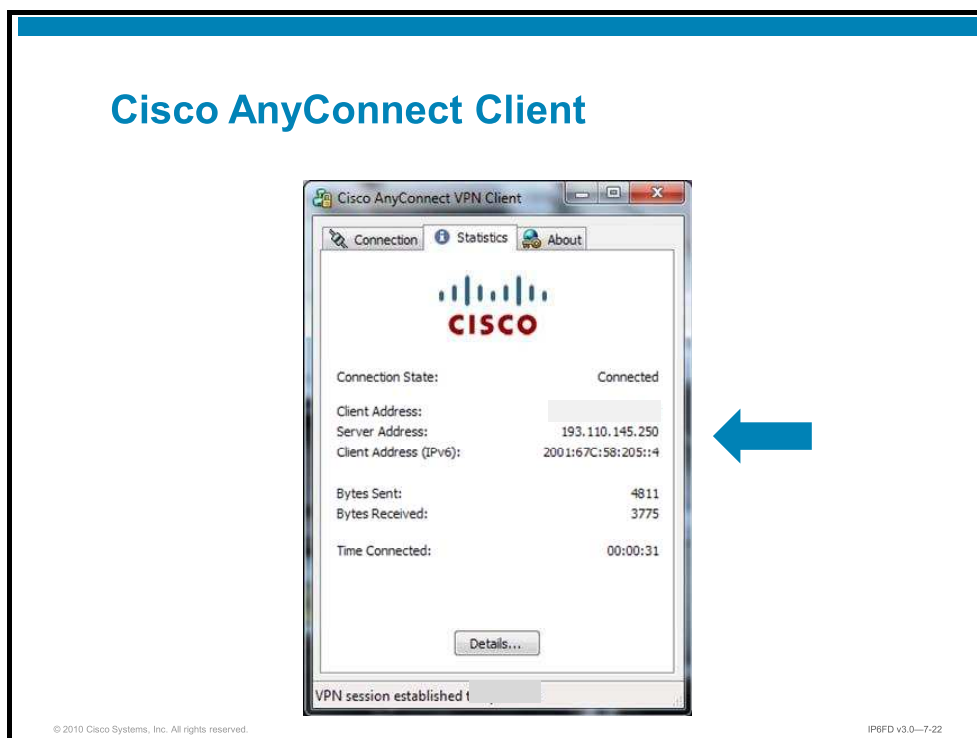
All SSL VPN communications run over TCP sessions, which are managed as normal TCP sessions by Network Address Translation (NAT) devices. There is no need for any traditional NAT traversal mechanism that adds a User Datagram Protocol (UDP) header to every packet to allow it to be forwarded on a regular firewall.

Because TCP, which is connection-oriented, is used, a session is established for the VPN connection, and the devices in the traffic path keep the state of the session (established or terminated). UDP-based VPN solutions are less stable and usually do not survive long timeout periods because the firewall might delete the session from its tables. This cannot happen with a TCP session, where the firewalls keep the connection state open until the session is terminated (that is, until they see the reset (RST) packet).

The SSL VPN feature and the Cisco AnyConnect VPN client require licenses to work.

Note SSL VPN can also be used in clientless mode; that is, without requiring any client software, except for a web browser, on the client. This way, the browser opens a protected environment in which the user can work and which is particularly suitable for public Internet places. This feature is called clientless SSL VPN (formerly named Cisco IOS WebVPN).

Cisco AnyConnect Client



The Cisco AnyConnect client establishes an SSL tunnel to transport both IPv4 and IPv6 traffic. An SSL VPN session can transport both IPv6 and IPv4 traffic.

The printout from the Windows **ipconfig** command looks like the following:

Windows IP Configuration

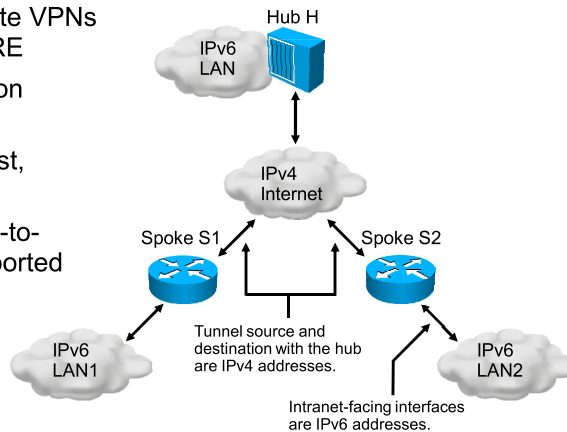
Ethernet adapter Local Area Connection 2:

```
Connection-specific DNS Suffix  . : cisco.com
IPv6 Address. . . . . : 2001:67c:58:205::4
Link-local IPv6 Address . . . . . : fe80::2948:928a:dfb3:e994%18
IPv4 Address. . . . . : 192.168.205.54
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : ::
```

IPv6 Dynamic Multipoint VPN (DMVPN)

Payload Protocol	Transport Protocol
IPv6	Multipoint GRE over IPv4

- A mesh of site-to-site VPNs using multipoint GRE
- Simpler configuration management
- Support for multicast, routing, and QoS
- On-demand, spoke-to-spoke tunnels supported



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--7-23

The Cisco DMVPN feature is an enhanced IPsec VPN technology that combines IPsec, multipoint GRE (mGRE) tunnels, and the Next Hop Resolution Protocol (NHRP), which is a dedicated protocol to resolve the next-hop router for a given network.

DMVPN provides users an ease of configuration via crypto profiles, which override the requirement for defining static crypto maps, and dynamic discovery of tunnel endpoints.

The foundation for DMVPN is mGRE, which allows multicast traffic to be transmitted and encrypted. Routing protocols use multicast traffic to propagate routes. Standard quality of service (QoS) mechanisms can be used for such VPN traffic.

Tunnels can be established either between a remote office (spoke) and the main location (hub), or between remote offices for direct data exchange without overloading the hub links.

In DMVPN for IPv6, the public network (the Internet) is a pure IPv4 network, and the private network (the intranet) is IPv6-capable.

DMVPN and NHRP

Next Hop Resolution Protocol:

- Based on the destination IPv6 subnet, NHRP resolves the IPv6 address of the next-hop router.
- Client and server protocol:
 - Server (hub) maintains public interface addresses of each spoke.
 - Client (spoke) registers with the server.
- Direct tunnels are possible between spokes.
- NHRP operation:
 - Packet is routed to a spoke subnet.
 - NHRP looks up the next hop (and initiates a query if required).
 - Next-hop adjacency is inserted in the switching path.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-24

The NHRP is a client and server protocol in which the hub is the server and the spokes are the clients.

- The hub maintains an NHRP database of the public interface addresses of each spoke.
- Each spoke registers its real address when it boots and queries the NHRP database for the real addresses of the destination spokes, to build direct tunnels.

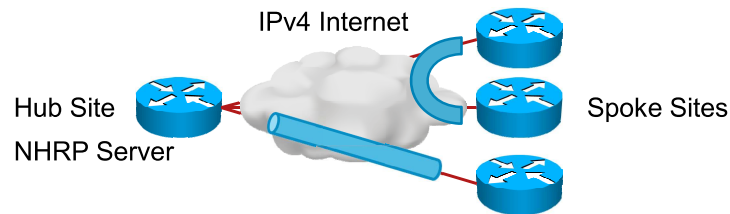
These are the main components that interoperate with NHRP:

- **mGRE tunnel interface:** An mGRE tunnel interface allows a single GRE interface to support multiple IPsec tunnels and simplifies the size and complexity of the configuration.
- **IPsec encryption:** An IPsec tunnel interface allows the protection of site-to-site IPv6 traffic with native encapsulation.

NHRP is automatically invoked for mGRE tunnels that carry the IPv6 passenger protocol. When a packet is routed and is sent to the switching path, NHRP looks up the given next hop and, if required, initiates an NHRP resolution query. If the resolution is successful, NHRP populates the tunnel endpoint database, which then populates the Cisco Express Forwarding adjacency table. The subsequent packets are Cisco Express Forwarding switched if Cisco Express Forwarding is enabled.

Configuring DMVPN

1. Configure an IPsec profile in DMVPN for IPv6.
2. Configure the hub for IPv6 over DMVPN.
3. Configure the spoke for IPv6 over DMVPN.



- Direct tunnels are established by using NHRP.
- IPsec encrypted connections exist between sites.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-25

These steps are necessary to configure DMVPN:

Step 1 Configure an IPsec profile in DMVPN for IPv6.

The IPsec profile shares most of the same commands with the crypto map configuration, but only a subset of the commands are valid in an IPsec profile. Only commands that pertain to an IPsec policy can be issued under an IPsec profile; you cannot specify the IPsec peer address or the access control list (ACL) to match the packets that are to be encrypted.

Step 2 Configure the hub for IPv6 over DMVPN.

The hub router for IPv6 over DMVPN must be configured for mGRE and IPsec integration: The tunnel must be associated with the configured IPsec profile. NHRP must also be configured on the hub.

Step 3 Configure the spoke for IPv6 over DMVPN.

The spoke must be configured for IPsec, mGRE, and NHRP.

Note More information about configuring DMVPN for IPv6 can be found in **Implementing Dynamic Multipoint VPN for IPv6** at

<http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-dmvpn.html>.

Verifying DMVPN (Example)

```
Router# show dmvpn ipv6 detail

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
N - NATed, L - Local, X - No Socket
# Ent --> Number of NHRP entries with same NBMA peer
NHS Status: E --> Expecting Replies, R --> Responding
UpDn Time --> Up or Down Time for a Tunnel
=====

Interface Tunnel1 is up/up, Addr. is 10.0.0.1, VRF ""
Tunnel Src./Dest. addr: 192.169.2.10/MGRE, Tunnel VRF ""
Protocol/Transport: "multi-GRE/IP", Protect "test profile"

IPv6 NHS: 2001::6 RE
Type:Spoke, Total NBMA Peers (v4/v6): 1
1.Peer NBMA Address: 192.169.2.9
   Tunnel IPv6 Address: 2001::6
   IPv6 Target Network: 2001::/112
   # Ent: 2, Status: NHRP, UpDn Time: never, Cache Attrib: S

IPv6 NHS: 2001::6 RE
Type:Unknown, Total NBMA Peers (v4/v6): 1
2.Peer NBMA Address: 192.169.2.9
   Tunnel IPv6 Address: FE80::1
   IPv6 Target Network: FE80::1/128
   # Ent: 0, Status: UP, UpDn Time: 00:00:24, Cache Attrib: D
```

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0-7-26

The **show dmvpn ipv6 detail** command gives you information that is obtained from NHRP. This sample shows target IPv6 networks that are connected to VPN peers in the IPv4 Internet. Communication between sites is maintained by using an mGRE tunnel, and traffic is encrypted using IPsec protocols.

The following command can be used as well to monitor DMVPN:

- **show dmvpn [ipv4 | ipv6] [peer [nbma | tunnel {ip-address | ipv6-address}] | network {ip-address mask}] [vrf vrf-name] [interface tunnel number] [detail] [static] [debug-condition]**
- **show ipv6 nhrp [dynamic [ipv6-address] | incomplete | static] [address | interface] [brief | detail] [purge]**
- **show ipv6 nhrp multicast [ipv6-address | interface]**
- **show ipv6 nhrp summary**
- **show ipv6 nhrp traffic [interface tunnel number]**

The following troubleshooting and debugging commands are available:

- **clear dmvpn session [peer {nbma | tunnel ipv4-address | ipv6-address}] [interface tunnel number] [vrf vrf-name] [static]**
- **clear ipv6 nhrp [ipv6-address | counters]**
- **debug dmvpn [condition [unmatched] | [peer [nbma | tunnel | ipv4-address | ipv6-address] | vrf [vrf-name] | interface {tunnel number} | error | detail | packet | all | nhrp | crypto | tunnel | socket | all]**
- **debug nhrp {ipv4 | ipv6} [cache | extension | packet | rate]**
- **debug nhrp condition [peer [nbma | tunnel | ip-address | ipv6-address]] | interface tunnel number | [vrf vrf-name]**
- **debug nhrp {ipv4 | ipv6} error**

Summary

This topic summarizes the key points that were discussed in this lesson.

Summary

- Security in IPv6 is implemented by using the IPsec security suite, which is mandatory for IPv6. You can still choose to use, or not to use, IPsec protections.
- IPv6 supports the AH and ESP security headers for packet integrity, source authentication, replay protection, confidentiality, and limited traffic flow confidentiality.
- IKE issues regarding security need to be considered before deployment. IKE is not required to be supported in IPv6 stacks and is not widely available. Cisco does support IKE for IPv6.
- Cisco IKE and IPsec for IPv6 is a robust security solution. Key rollover is provided, and various transforms, including AES-256, are available. Many diagnostic and troubleshooting commands are available.
- IPsec protocols are used for VPN. There are several variants of VPN connections, including site-to-site IPsec VPN, SSL VPN, and DMVPN.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--7-27

Discussing Security Issues in an IPv6 Transition Environment

Overview

Recommending the most appropriate integration mechanism to clients that are implementing IP version 6 (IPv6) is often necessary. If a client is unaware of the security risks that certain integration methods pose, then its network could be exposed to security breaches.

This lesson describes security issues in an IPv6 transition environment.

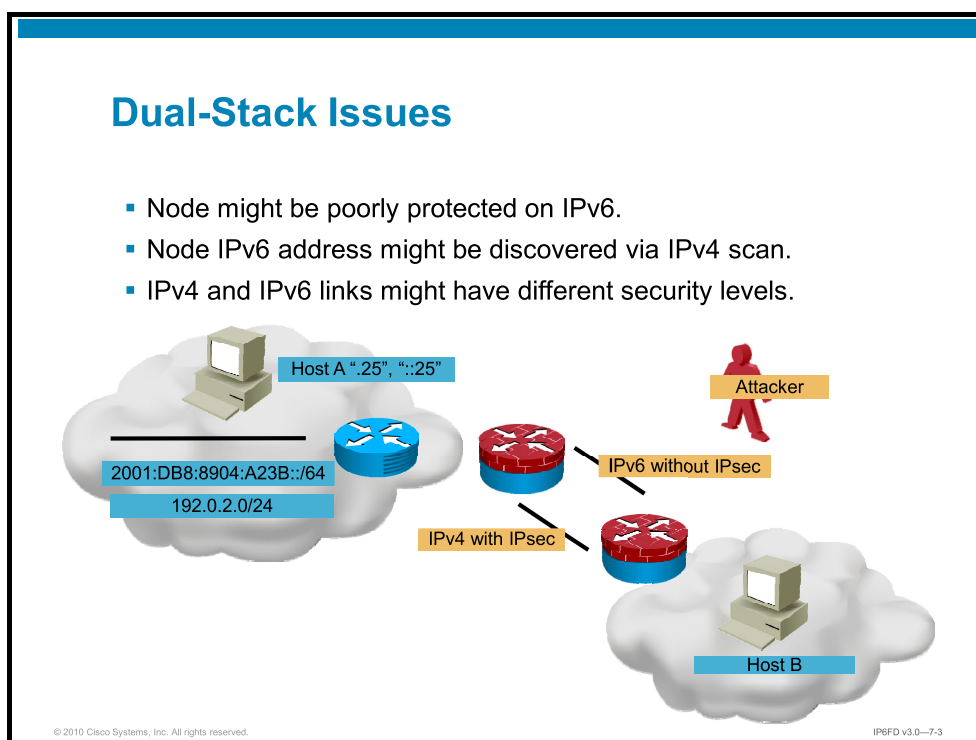
Objectives

Upon completing this topic, you will be able to identify security and transition issues in an IPv6 environment. This includes being able to meet these objectives:

- Describe dual-stack transition security issues
- Identify tunnel security issues in transition environments
- Describe the security issues of NAT-PT
- Describe the security issues that are caused by the increased need for ICMP

Dual-Stack Issues

This topic describes dual-stack issues.



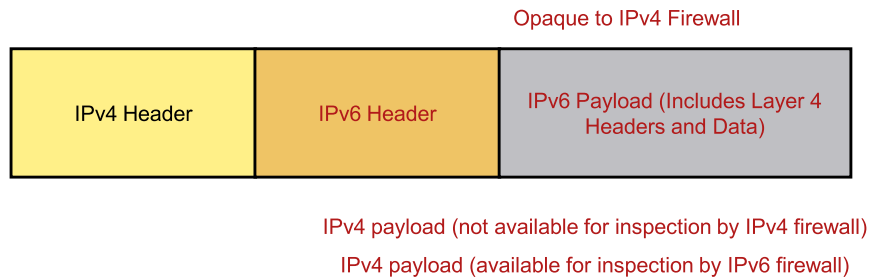
Dual-stack nodes differ from single-stack nodes in these ways:

- The node might be more hardened on IP version 4 (IPv4) than on IPv6, possibly because of the comparatively novel nature of IPv6 applications. For example, the Secure Shell (SSH) listener daemon on IPv4 might be overflow-proof, brought about by years of probes by the hacking community. The IPv6 counterpart, or code changes that are needed by a multiprotocol SSH daemon, might be vulnerable.
- Depending on local practices, a node at 192.2.0.25 (for example) might be assigned IPv6 static address 2001:DB8:2:0::25, as a way to remember the IPv6 address. This approach might be handy for system administrators, but makes it simple to discover a dual-stack node IPv6 address when the IPv4 address is known. Scanning attacks are more difficult in IPv6 because of the large, sparse nature of the /64 prefix. Using the same host ID for IPv4 and IPv6 compromises the advantage of IPv6.
- Users typically make connections to names, not addresses, and are indifferent and unaware of the IP version that carries their traffic, leading to inconsistent behavior or security issues. In the figure, a dual-stack environment is built by using IPv4 IPsec, but not IPv6 IPsec for a WAN link (perhaps because of code availability or local preference). It would be easy to believe that traffic is secure across the WAN, not realizing that a change in the Domain Name System (DNS) mapping for the name from an A to a QuadA record would defeat the WAN privacy measure by shifting traffic to IPv6.

Transition Mechanism Challenges

Inability to inspect tunneled packets:

- Firewalls typically have no visibility into the payload of a tunneled packet.



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-4

The introduction of IPv6 creates the inability to inspect tunneled packets, which in turn raises several questions. If end-to-end security is used, which role do firewalls play? How can end-system security be improved?

Many enterprises rely on edge devices (firewalls and routers doing packet filtering) to protect internal infrastructure and hosts from external, IP-based attacks. A tunneled packet consists of an outside header and its associated fields. Those associated fields are available for inspection by firewalls, but the firewall views the entire tunneled packet as the packet payload. The firewall is typically unable to inspect the embedded IPv6 packet or its payload.

For example, if an IPv6 packet that uses a destination options extension header is tunneled inside an IPv4 packet, and the tunnel passes through the firewall, then the inner packet is not inspected. If the destination option in the packet is known to cause an internal host problem, it cannot be filtered out by the firewall.

Note For more information on some of the issues that are associated with the integration of IPv6, see the white paper titled *Review of IPv6 Transition Scenarios for European Academic Networks* at <http://www.zdnet.com>.

Solutions

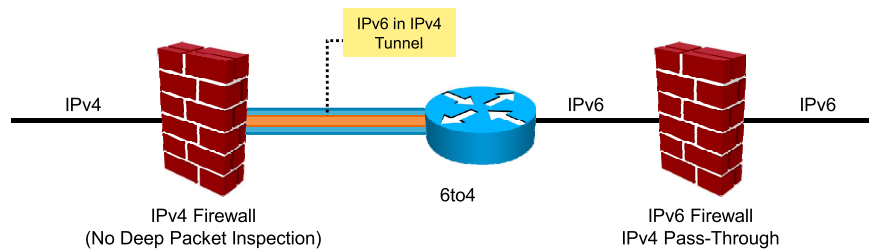
There are several solutions to the problem of tunneled packet inspection:

- Use firewalls in depth, inspecting traffic only when it has been de-encapsulated. De-encapsulate all tunneled packets in a bastion environment and then pass them through another firewall, such as an IPv6 firewall, for inspection.
- Use an edge device that provides both a firewall and a tunnel endpoint. This device acts as the tunnel endpoint and so can inspect de-encapsulated packets before passing them to the internal network. This approach requires using a product that combines firewall capability with 6to4 or Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) functions.

- Do not allow tunnels through the edge device, or firewall, at all. Most tunnels are built by using protocol 41 (IPv6-in-IPv4) or User Datagram Protocol (UDP). These packets can be completely prohibited, because these parameters are present in the outer IP header fields, which the firewall can inspect. Prohibiting protocol 41 tunnels will usually not provide acceptable services to internal devices.

Transition Mechanism Challenges (Cont.)

6to4 firewall solution



© 2010 Cisco Systems, Inc. All rights reserved.

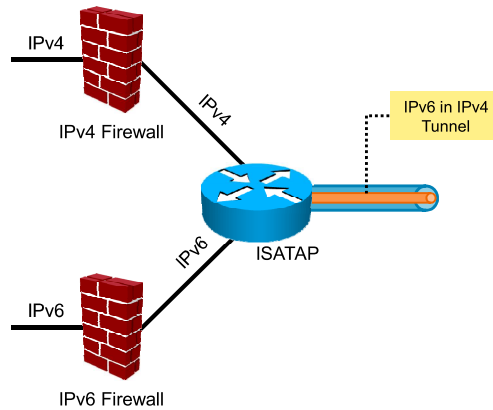
IPv6FD v3.0—7-5

For 6to4, the outside firewall inspects all packets: IPv4 without an embedded tunnel and IPv6-in-IPv4 tunnel packets. The 6to4 gateway de-encapsulates the IPv6-in-IPv4 packets, which are then inspected by an IPv6 firewall. The IPv6 firewall inside the enterprise passes IPv4 packets without inspection. The packets can also be split via routing rules so that only IPv6 traffic passes through the inside firewall.

Transition Mechanism Challenges (Cont.)

ISATAP firewall solution:

- In both the 6to4 and ISATAP firewall solutions, no packets arrive at the IPv6 transition device without inspection.



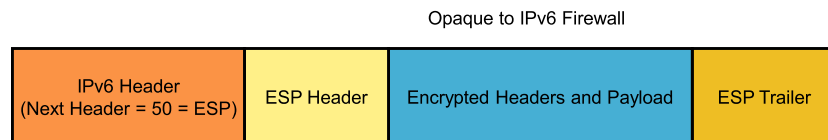
For ISATAP, a protocol-specific firewall inspects both IPv4 and IPv6 packets before they arrive at the ISATAP router. Encapsulation in ISATAP is an internal-facing function. There is no need for internal inspection, because all traffic has been effectively inspected.

Note that, in both the 6to4 and ISATAP firewall solutions, no packets arrive at the IPv6 transition device without inspection. This is because decapsulation servers are protected and located in a demilitarized zone (DMZ).

Transition Mechanism Challenges (Cont.)

Use of IPsec:

- Prevalent use of IPsec by internal hosts (peer-to-peer): Edge devices, or firewalls, cannot inspect ESP-encrypted payloads.



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-7

IPv6 encourages the use of peer-to-peer computing, in which peers have direct connections with each other rather than being shuttled through IP proxy environments, Network Address Translation (NAT) devices, or Layer 4 application gateways. Increasingly, these peers will choose to implement IPsec protection for their sessions, which prohibits the inspection of Layer 4 headers (TCP ports, for example) and packet payloads.

In these cases, the packets must transit the edge device (filtering router or firewall) without being inspected, because the edge device is not a member of the encryption regime. Because these packets are passed to the internal (enterprise) IPsec peer without any inspection, there is a risk that the packets are part of a malicious session or contain harmful data.

Solutions

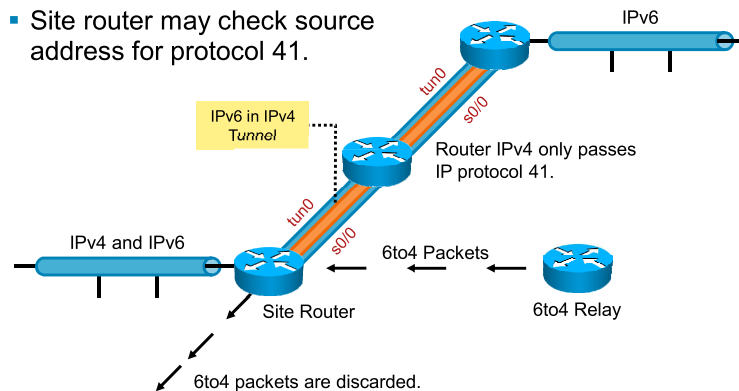
There are several solutions to the IPsec problem:

- Edge firewalls can entirely prohibit IP protocol 50 (Encapsulating Security Payload [ESP]) traffic. This solution often is unacceptable because peer-to-peer applications that use IPsec will not function.
- You can implement host firewalls on the hosts that are involved in the IPsec session. This solution is the best one because it does not prohibit connections but allows inspection of decrypted traffic. In enterprises that have the infrastructure to support host-based firewalls, this solution is identical to mechanisms that are used to secure IPv4 IPsec sessions in which host firewalls are used to inspect traffic after decryption.
- You can also require all IPsec sessions to terminate on the firewall, then inspect the traffic and route the inside IPv6 packets natively within the enterprise. This solution allows inspection, but it breaks the peer-to-peer model for the enterprise hosts.

Transition Mechanism Challenges (Cont.)

Manual tunnels from edge devices:

- Middle router may filter protocol 41, based on source and destination address.
- Site router may check source address for protocol 41.



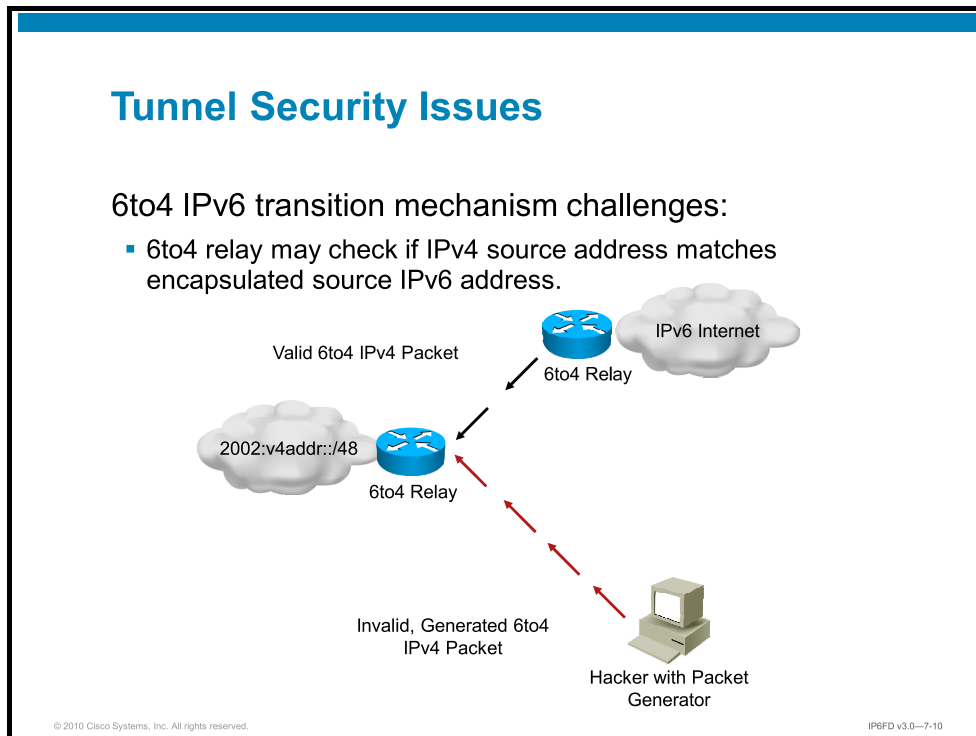
Manual tunnels (configured tunnels) improve site security by specifying the remote tunnel endpoint. In 6to4, the site edge tunnel device must accept encapsulated packets from any IPv4 address. 6to4 relays can be anywhere, as can 6to4 gateways. With manual tunnels, the remote endpoint is configured explicitly, so contracts and other trust relationships can be put in place.

The site receiving service obtains a routable IPv6 prefix (probably a /48 prefix) and, via the tun0 interface, treats the tunnel as an IPv6 interface. IPv6 traffic that is bound for the prefix of the site is encapsulated and sent to the tun0 interface of the site, where the traffic is de-encapsulated and routed into the site core. With an explicit tunnel endpoint, the site router should not receive any encapsulated packets for which the IPv6 packets inside are not actually destined for (have a destination address of) the site network. Routing on the ISP router tunnel endpoint should ensure this benefit. The packets might still contain spoofed IPv6 packets (in which the source address is not legitimate).

Because this scenario (configured tunnel) gives the site what looks like native IPv6 capability, 6to4 site traffic transits from the IPv6 side; the enterprise is not using 6to4 at all. In this scenario, the site router would, via an access control list (ACL), reject IP protocol 41 packets from any source other than the explicitly configured partner. Dropping these packets is a much less expensive (CPU-intensive) process than de-encapsulating and inspecting embedded packets.

Tunnel Security Issues

This topic describes tunnel security issues in transition environments.



By default, 6to4 traffic will be accepted and de-encapsulated from any source from which regular IPv4 traffic is accepted. If for any reason, doing so is a security risk (for example, if IPv6 spoofing is more likely than IPv4 spoofing), then you can apply additional source address-based packet filtering.

A possible plausibility check is whether the encapsulating IPv4 address is consistent with the encapsulated 2002:: address. Any 6to4 traffic (for example, from 2002:v4-addr::50) should come encapsulated in an IPv4 packet from the same IPv4 address. If this check is applied, exceptions to it must be configured to admit traffic from relay routers. If the 6to4 site needs only to accept traffic from specific other 6to4 sites, then an IPv4 ACL can be used to restrict tunnel endpoints. If the site accepts relay traffic, traffic must be accepted from any IPv4 address, because it is impossible to know in advance the locations of all 6to4 relays.

Example

The figure shows a simple example. The 6to4 router has no prior knowledge of where IPv6-in-IPv4 6to4 relay packets will come from. Therefore, the 6to4 router must de-encapsulate all packets from all senders. A malicious sender can fabricate bogus packets and, because decapsulation is a costly process, overwhelm the 6to4 router. This issue is especially true if 6to4 is managed in software, a common scenario for early products.

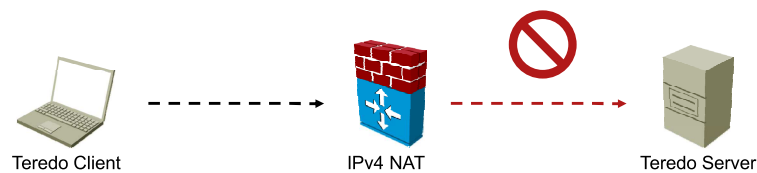
If the hacker in the example has compromised a valid 6to4 relay and is injecting bogus packets with valid packets, then blocking the source of the traffic will block both valid and invalid packets.

Note In any case, any 6to4 traffic whose source or destination address embeds an IPv4 address in a format other than a global unicast address must be silently discarded by both encapsulators and de-encapsulators. Specifically, IPv4 addresses that are defined in RFC 1918, *Address Allocation for Private Internets*, broadcast, subnet broadcast, multicast, and loopback addresses are unacceptable.

Tunnel Security Issues (Cont.)

Teredo:

- Teredo requires outbound UDP. In the enterprise, all outbound UDP traffic should be denied, with specific UDP services allowed to specific Internet sites.
- Other security issues exist, all of which can be partially mitigated and most of which are not new.



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--7.11

The Teredo protocol allows dual-stack nodes that are isolated on an IPv4-only network, using a private address and inside a NAT device, to become IPv6 peers on the IPv6 Internet. The client obtains a globally routable address and is logically considered an IPv6 node. Security issues relate to using Teredo, in terms of internal machines being compromised via IPv6. Teredo requires outbound UDP capability from inside the NAT environment. Enterprises can block this UDP traffic and effectively disable Teredo.

Note See the IETF draft *Tunneling IPv6 over UDP through NATs* (draft-huitema-v6ops-teredo-01.txt).

Security issues that relate to Teredo include the following:

- Security risks of directly connecting a node to the IPv6 Internet

Teredo nodes are subject to IPv6-based attacks, as are any native IPv6 node. Although IPv4 packet protection that the NAT device may have provided the Teredo node is still in effect, no such protection exists for IPv6 packets, which come encapsulated but unchecked in the path to the client. Use of a local IPv6 firewall on the Teredo client is recommended.

- Spoofing of Teredo servers to enable a man-in-the-middle attack

Teredo clients configure their global IPv6 address with assistance from the Teredo server. The client chooses a server either by looking up a well-known DNS name (which can be locally controlled) or by statically configuring the address of the desired server. The most damaging result of this attack is denial of service (DoS) to that Teredo client. The attacker can accomplish this DoS by establishing itself as the Teredo server and then not assisting the client with Teredo session setup. This attack can be mitigated by using IPsec or other mechanisms to secure the client-server communications.

- Potential attacks that are aimed at denying the Teredo service to a Teredo client

Numerous attacks are aimed at either overloading the Teredo servers, Teredo relays, or Teredo clients with too many connections or too much traffic. These attacks are varied, and each has a partial mitigation solution.

DoS attacks against non-Teredo participating nodes can be enabled by the Teredo service. These attacks are created as a result of deploying the Teredo service. For example, to perform a DoS attack against an IPv6 node, the attacker can generate a large volume of IPv6 traffic and send it to the node. To do so and not be traced, the attacker needs to use IPv6 source address spoofing.

Alternatively, an attack can generate many IPv6 packets, wrap them in an IPv4 packet, and send them to a Teredo relay, then let the relay pass the packets to the IPv6 target. To accomplish this, the attacker needs to perform IPv4 address spoofing and carefully construct the IPv4 packet. Otherwise, packets can be traced to the relay from IPv6 and then to the IPv4 source of the attack.

Summary of Tunnel Security Issues

In general, tunneling can break security or operational assumptions that are held by the team that deployed NAT.

IPv6-over-IPv4 (protocol 41) tunneling is typically explicitly allowed or implicitly disallowed through edge security devices. Tunneling IPv6 over IPv4 UDP is often a different situation, because UDP must usually be allowed through, at least partly in a stateful manner. Security managers should ensure that they know which UDP traffic, even if initiated from inside, is allowed through the firewall or NAT box.

All types of tunneling pose risks in home and small office, home office (SOHO) environments. In these environments, the hosts might not be as managed as in other networks (for example, network services might be enabled unnecessarily), leading to possible security break-ins or other vulnerabilities.

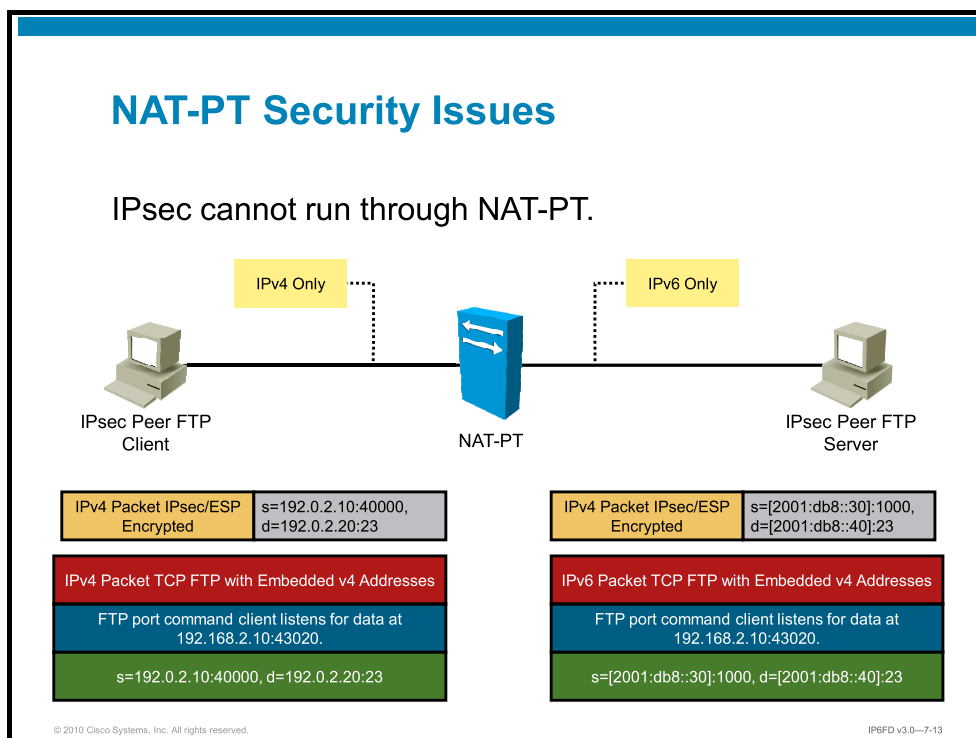
When you deploy tunneling solutions, especially tunneling solutions that are automatic and can be enabled easily by users who do not understand the consequences, you should take care not to compromise the security assumptions that are held by the users.

You should be aware of these tunnel security issues:

- Native IPv4/IPv6 dual-stack environments should be preferred. Avoid the use of tunneling if possible. Tunnels obscure the contents of encapsulated packets and make inspection and security filtering more difficult.
- Configured tunnels should be preferred over automatic tunnels. Avoid a situation in which the tunnel edge device (encapsulator or de-encapsulator) must process packets from any source. With configured tunnels, the far end of the tunnel is known, and filtering rules can be managed more easily. This approach also allows contractual agreement between tunnel peers. This solution is not possible with 6to4 or Teredo, in which some devices are chosen based on current network topology on the IPv6 side, such as IPv6 relays.

NAT-PT Security Issues

This topic describes security issues of NAT-Protocol Translation (NAT-PT).

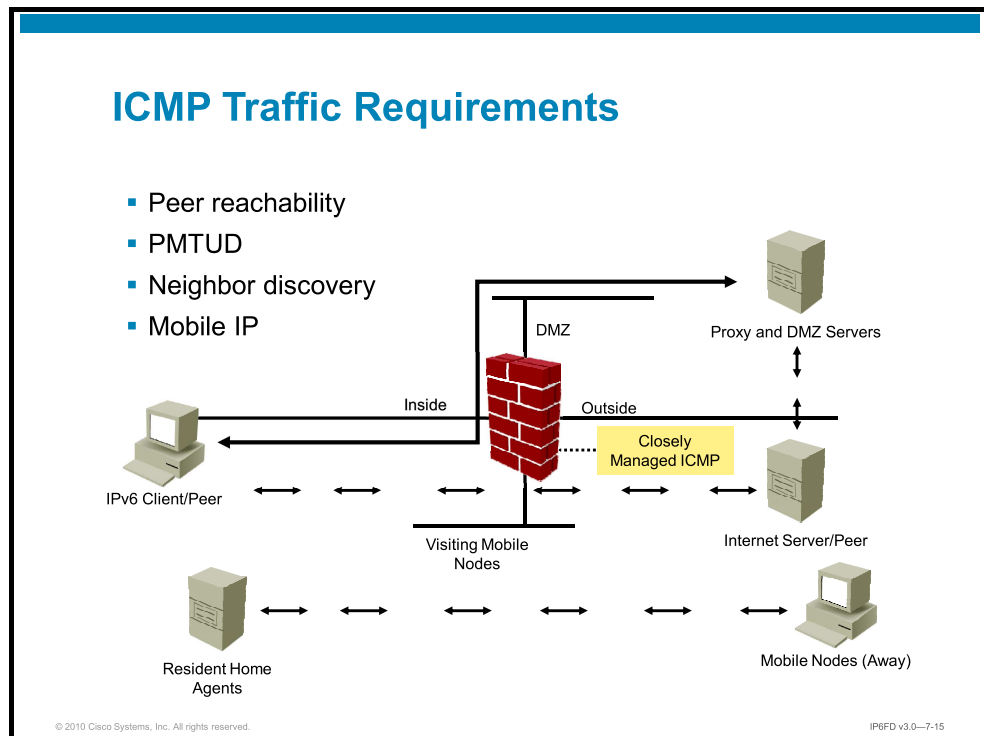


There are three major issues with NAT-PT:

- For the top set of IP packets (ESP) that are shown in the figure, the packets on either side of the NAT are markedly different. IPsec will not work through the environment, because the packet at the receiving peer will not pass the validation tests. These tests verify that the packet was not tampered with in transit, which it clearly was.
- The second set of packets shows what NAT-PT does to an FTP session. FTP embeds IP addresses in the payload of the TCP packet. For this reason, the FTP packet that tells the FTP server the address and port with which to connect back to transfer the data portion of the communication has embedded IPv4 addresses. Clearly, an IPv6 node cannot connect to that address. Even if the node could, the NAT-PT box has no knowledge of port 43020 and would not accept the packet. Application gateways that look deeper into the packet can be built to manage situations like this. However, such gateways are often slow to be developed, are application-specific by design, and use much processing power.
- NAT-PT devices are single points of failure for devices inside the NAT-PT boundary. Devices within the boundary can be isolated from outside IPv4 devices by a DoS attack on the NAT-PT box.

ICMP Traffic Requirements

This topic describes security issues that are caused by the increased need for Internet Control Message Protocol (ICMP).



Many enterprises block ICMP traffic at the edge device as a matter of policy, or they apply restrictions so that only specific systems (such as proxy servers) can exchange ICMP. With IPv6, more devices are peers with devices on the Internet, and IPv6 makes more use of ICMP. In configuring edge packet filtering, care must be taken to permit all needed functions.

These services rely on ICMP to function properly:

- **Peer reachability information:** Allows nodes to learn that the target node for a session is not reachable. Note that the destination node is reachable, but the specific protocol (such as UDP) or port (such as domain UDP), among others, does not respond. The receiving node also reports, via ICMP, packet field problems to the sending node.
- **Path maximum transmission unit (MTU) discovery:** In IPv6, only source nodes can fragment packets. Source nodes sometimes perform path discovery to learn how large a packet can be sent to a specific destination. This process depends on the ability of ICMP messages that are sourced from any router on the path to reach the sending (protected) node.
- **Neighbor discovery:** Neighbor discovery depends on ICMP traffic to function.
- **Mobile IPv6:** Mobile nodes and home agents exchange ICMP messages for various functions.

For IPv6 environments, careful engineering of ICMP packet flows at the enterprise edge is required, rather than the drop-all posture that is commonly adopted for IPv4 environments.

Summary

This topic summarizes the key points that were discussed in this lesson.

Summary

- Dual stack might lead to uneven security posture at Layer 3: one for IPv4 and one for IPv6.
- IPv6 transition mechanisms are unable to inspect tunneled packets.
- When using IPsec, edge devices cannot inspect ESP-encrypted payloads.
- Tunneling can compromise perimeter security.
- IPsec will not run through NAT.
- When configuring edge packet filtering, be sure to deny all by default while permitting needed functions.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-16

Understanding IPv6 Security Practices

Overview

You must consider several high-level security practices for IP version 6 (IPv6) networks, for your deployment strategy and network design.

This lesson describes security practices for IPv6 deployment.

Objectives

Upon completing this lesson, you will be able to describe security practices for IPv6 deployment. This ability includes being able to meet these objectives:

- Compare common network threats in IPv4 and IPv6 networks, and offers solutions to mitigate them
- Describe IPv6 security practices that you should consider
- Describe the need to hide topology information when possible, for security
- Secure a local link
- Manage ICMPv6 traffic at the IPv6 edge
- Develop a mobility support plan
- Use transition mechanisms as transport
- Secure the routing plane
- Describe an IPv6 early warning system

Threats in IPv6 Networks

This topic compares common network threats in IP version 4 (IPv4) and IPv6 networks, and offers solutions to mitigate them.

Threats in IPv6 Networks

- Several attacks in IPv4 can be used to target IPv6 hosts as well:
 - Network scans
 - Spoofing
 - Fragmentation
 - ARP and DHCP
 - Smurf
 - Source routing
 - Rogue devices
 - Traffic redirection
 - Attacks on services

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-3

Some network security threats that are present in IPv4 are still present in IPv6. Some of them are no longer an issue. There might also be a few new threats, not seen in IPv4.

These threats are discussed in this topic:

- Network scans
- Address spoofing
- Fragmentation-related attacks
- Address Resolution Protocol (ARP)- and DHCP-related attacks
- Smurf (broadcast amplification) attacks
- Source routing
- Rogue devices
- Traffic redirection
- Direct attack on services (Layer 7)

Threats in IPv6 Networks (Cont.)

Network scans:

- Attacker scans the network to find:
 - Active hosts on the network
 - Active services on the hosts
- In IPv4, most subnets can have between 254 and 65,534 possible hosts (25 sec to 109 min, at 10 hosts per sec).
- In IPv6, a /64 subnet has $2 \cdot 10^{19}$ possible hosts (scanning time at 10 hosts per second is incalculably great).
- For MAC-derived addresses, scan time can be reduced by limiting scanning range to know OUI ranges.
- Network scans in IPv6 can be used as a DoS-type attack: Router or firewall keeps state table entries for nonexistent hosts until they time out.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0-7.4

Network Scans

An attacker can target a well-known host, such as a web or mail server. Or the attacker can try to find a host that is not well known; for example a database server, which should be accessed only by web servers and not directly by end nodes on the Internet. If such servers are not protected by a host-based or network firewall, all the attacker needs to do is to get the IP address of the server.

An attacker who cannot acquire or guess the address might try to scan the entire range, to see whether any of the addresses appear live. In IPv4, the time that is required to scan a /24 subnet, at a rate of 10 hosts per second, is approximately 25 seconds. This is completed very quickly. Trying to scan a /16 subnet at the same rate might take almost 2 hours. Note that network delay can increase these times.

IPv6 uses /64 subnets, by default. The use of nonstandard subnet length is possible but not recommended. If an attacker scans a /64 subnet at the same rate (10 hosts per second), the magnitude of time that is needed to scan the entire address space is incalculably great. Scanning the address space for live hosts becomes effectively useless.

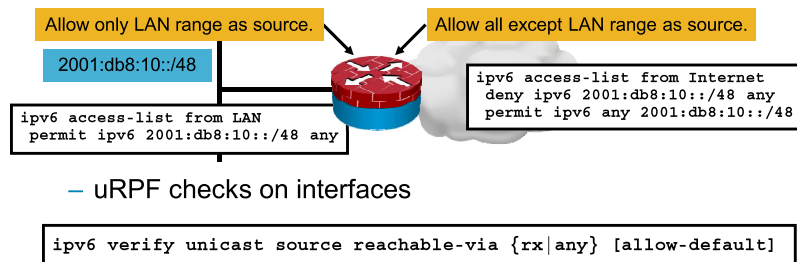
The previous calculation assumes that hosts use randomized addresses. If a host uses a modified EUI-64 host portion of the address, which is MAC derived, the scanning range can be reduced to known Organizationally Unique Identifier (OUI) prefixes.

These calculations show that IPv6 LAN scans are more or less useless. The scans can still be used as denial of service (DoS) attacks. By sending a stream of packets to nonexistent hosts, an attacker can fill up firewall state tables or router neighbor cache tables. In an IPv4 /24 network, even if one sends packets to all possible hosts, the result is 254 incomplete entries, a number that is well below dangerous levels. In an IPv6 /64, the number of legitimate entries exceeds all practical limits.

Threats in IPv6 Networks (Cont.)

Spoofing:

- Attacker modifies packet source address.
- Countermeasures include:
 - ACLs that filter incoming and outgoing packets



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7.5

Spoofing

Attackers can use address spoofing to make packets seem to come from one address so that they can bypass source address filters. In closed networks, network rules might permit local addresses and deny external addresses as sources. An attacker who spoofs the source address might fool the network filters into accepting illegitimate traffic.

To prevent such traffic, filters should be placed on network boundaries. Packets that have a LAN address as the source address must not be permitted to enter the network. Also, only those packets that have a LAN address as a source may exit the network. The latter rule enforces good behavior for LAN users; although it might not directly protect the network, it will prevent LAN nodes from initiating malicious or erroneous traffic.

Another check that can be implemented is Unicast Reverse Path Forwarding (uRPF). When uRPF is enabled, the router checks whether a route that points at the receive (Rx) or at any interface exists for any incoming traffic. The router can also consider the default route, when making this check.

Threats in IPv6 Networks (Cont.)

Fragmentation:

- Can be used to:
 - Circumvent firewall rules (fragments may not contain Layer 4 information)
 - Attack the network stack (malformed fragment data)
- Countermeasures include:
 - Not allowing fragments

```
deny ipv6 any any fragments
```

- Dropping all fragments (except last one) with size less than 1280 B

© 2010 Cisco Systems, Inc. All rights reserved.IP6FD v3.0--7-6

Fragmentation

IPv4 supports fragmentation of IP packets on transit nodes; IPv6 does not. Such fragments can be used either to circumvent Layer 4 firewall rules or to attack the network stack. The former can be achieved because TCP or User Datagram Protocol (UDP) header appears only in the first fragment, which is filtered by the access control list (ACL), whereas the rest of the fragments do not contain Layer 4 information and cannot be matched. Firewalls implement Virtual Fragment Reassembly (VFR) to combat this problem. VFR increases resource consumption because the firewall must keep all fragments in memory until it can reconstruct the entire segment. The latter is done with specially crafted, malformed packets, which the target node cannot manage and might crash.

Because IPv6 does not support intermediate fragmentation but only end-to-end fragmentation, Path Maximum Transmission Unit Discovery (PMTUD) is essential to discover the optimum path maximum transmission unit (PMTU). If all nodes refrain from using fragmentation, then the solution to combat fragment attacks might be to deny fragments completely. Doing so might have the undesirable effect of also matching packets that are initial packets but that lack Layer 4 information, if the total size of IPv6 headers exceeds 1280 B.

Another option, when implemented, would be to drop all except the last fragment, if the size is less than 1280 B.

Threats in IPv6 Networks (Cont.)

ARP and DHCP:

- ARP is replaced by ICMPv6 with neighbor discovery.
- DHCP can be used (DHCPv6).
- Default implementations use route advertisements for address autoconfiguration.
- Attackers can forge responses and overwrite legitimate cache entries.
- IPv4 equivalent countermeasures are not yet implemented.
- For critical equipment, configure static neighbor entries.

```
ipv6 neighbor 2001:db8:10::1 FastEthernet 0/0 0000.1234.abcd
```

© 2010 Cisco Systems, Inc. All rights reserved.IP6FD v3.0—7-7

ARP

ARP is used in IPv4 to provide mapping between Layer 2 and Layer 3 addresses. An attacker can send forged ARP packets and change legitimate mappings. These actions can be used to redirect traffic from the final destination or gateway to the attacker machine. In IPv6, Neighbor Discovery Protocol (NDP) has taken over the role of ARP. The secure neighbor discovery (SeND) extension can cryptographically secure exchange of neighbor discovery packets. If SeND is not implemented, then static neighbor entries should be used for critical hosts.

DHCP

DHCP is used to dynamically assign addresses to hosts. In IPv6, DHCP can still be used. However, route advertisements are more often used for address assignment. DHCP is used only in the form of DHCP-lite, to provide additional information, such as Domain Name System (DNS) servers.

Threats in IPv6 Networks (Cont.)

Smurf (broadcast amplification):

- Attacker can send a spoofed packet to a broadcast address, to initiate response to a victim.
- IPv6 uses multicast rather than directed broadcast.
- Countermeasures:
 - If nodes conform to RFC 2463, no smurf attacks are possible in IPv6.
 - Filter all packets with multicast sources on border gateways.

```
ipv6 access-list from outside
deny ipv6 FF00::/8 any log
permit ipv6 any any
```

© 2010 Cisco Systems, Inc. All rights reserved.IPv6FD v3.0—7-8

Smurf

Smurf attacks work in IPv4, when many hosts are configured so that they respond to packets that are sent to a directed broadcast address. IPv6 does not use broadcast addresses, so this attack is not feasible. Attacks that are directed to link-scoped multicast addresses do not work.

Multicast addresses should never appear in a source field, so filtering such packets poses no harm.

Threats in IPv6 Networks (Cont.)

Source routing:

- Source routing allows sender node to specify strict or loose path of the traffic.
- Attacker can use loose source routing to redirect return traffic while spoofing source address.
- IPv6 uses routing headers for this functionality.
- Countermeasures:

- To prevent routers from processing routing header

```
no ipv6 source-route
```

- To filter out packets with routing header

```
ipv6 access-list from outside  
deny ipv6 any any routing log  
permit ipv6 any any
```

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-9

Source Routing

Attackers can use source routing to change the path of the traffic, by specifying intermediate hops in the IP header. IPv4 uses options to specify hops; IPv6 uses routing headers. Like in IPv4, processing of routing headers in IPv6 can be stopped by configuring the **no ipv6 source-route** command.

This configuration does not prevent end hosts from processing routing headers. Those hosts must be protected locally.

As an alternative, you can filter all packets with a routing header, if this functionality is not required in your network.

Threats in IPv6 Networks (Cont.)

Rogue devices:

- Attacker brings a new node to a network:
 - End hosts act as attack platforms.
 - Network devices tie with existing network:
 - Routers
 - Switches
 - DHCP servers
 - DNS servers
 - Wireless access points
- Countermeasures:
 - IPsec
 - Authenticated network access (802.1x)
 - SeND

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-10

Rogue Devices

Rogue devices are devices that an attacker introduces into the network. If your access layer is not secured, then rogue devices can enter the network and extend the network range, act as attack platforms, or provide illegitimate services that can deliver incorrect information to clients.

The only safe way to prevent unauthorized entry is to authenticate network access by encrypting and authenticating all traffic with IPsec, by authenticating devices with 802.1x, or by implementing SeND.

Threats in IPv6 Networks (Cont.)

Traffic redirection:

- ICMP redirects
- Routing table modification (injection of host routes)
- FHRP attacks
- Attacker redirects traffic across malicious nodes where she can:
 - Sniff the traffic
 - Alter the packets
- Countermeasures:
 - Authentication of redirect messages
 - Authentication of routing protocols
 - Authentication of FHRP

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-11

Traffic Redirection

Traffic can also be redirected to the attacker device by doing the following:

- Using spoofed Internet Control Message Protocol (ICMP) redirects
- Attacking routing protocols and injecting more-specific routes
- Attacking the First Hop Redundancy Protocol (FHRP)

By redirecting traffic, the attacker can sniff the traffic and extract important information, corporate secrets, login credentials, and so on. By modifying or originating traffic, the attacker can cause the network to behave in an undesirable way.

To prevent this behavior, it is essential to authenticate as many of the network protocols as possible. Most routing protocols and FHRP offer authentication.

Threats in IPv6 Networks (Cont.)

Attacks on services:

- TCP, UDP, and other protocols that run on top of IP are unchanged.
- Attacker can craft packets in a way to exploit programming or configuration errors in the service software.
- Countermeasures:
 - Deny access to unneeded services.
 - Deny access from nonclients to services.
 - Watch for security updates and apply patches.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-12

Attacks on Services

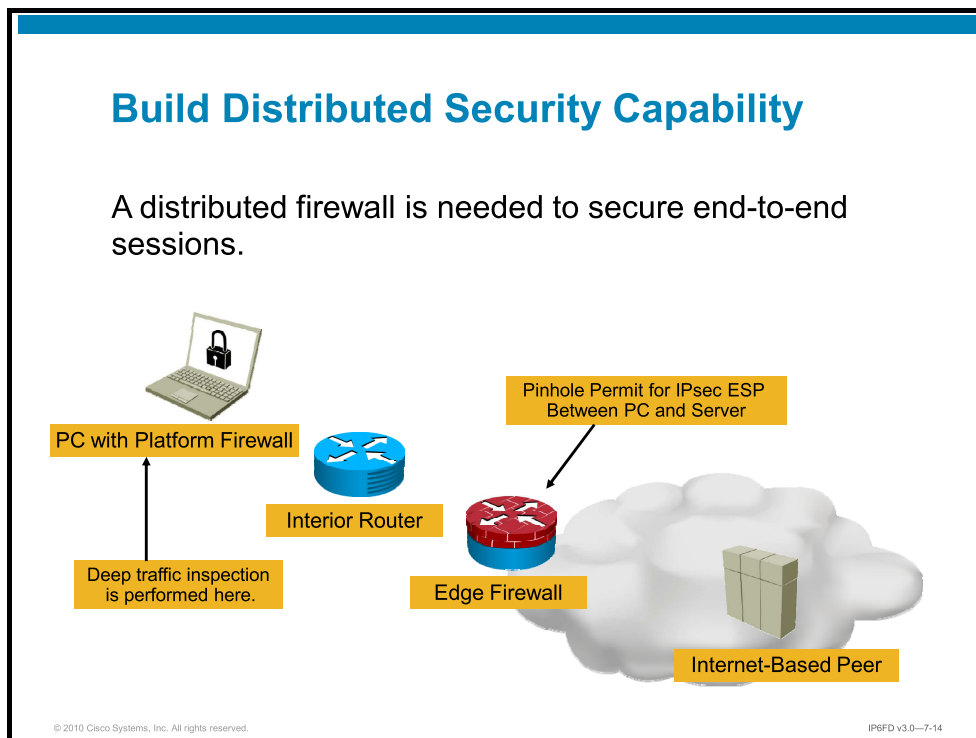
The attacker can attack end services directly, without bothering to attack the network. Still, the network can have configured filters that will give the attacker another barrier to cross.

If a service is not needed, access to it should be blocked. The guideline is to always try and restrict traffic as much as possible, by permitting access only to those services that must be accessible by a set of clients. For example, Internet clients might need access to web, email, and DNS services, whereas database and file servers should be accessible only to internal clients.

As with all software, watch for any security updates that your software vendor releases, and apply them as soon as they are released.

Build Distributed Security Capability

This topic describes IPv6 security practices that you should consider.



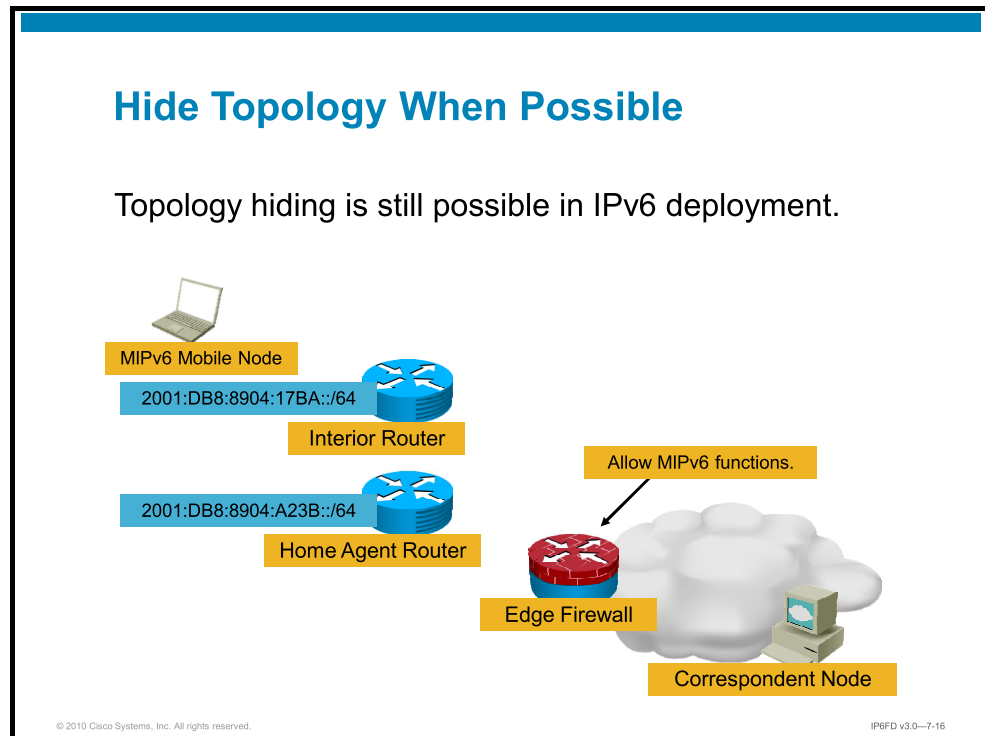
IPv6 will, over time, encourage the development and deployment of peer-to-peer applications. IPv6 is likely to have several impacts on traditional IP security practices:

- Peer-to-peer applications will be deployed from what are considered to be internal nodes to external, or Internet-based, nodes. Early in IPv6 deployment, security engineers are likely to try to refuse requests to allow peer-to-peer applications, but will be unable to do so as peer-to-peer applications become central to enterprise efforts. Many of these applications will be IPsec-protected (Encapsulating Security Payload [ESP]), and currently used edge devices, such as edge firewalls and network-based intrusion detection systems (IDSs), will be ineffectual. Inspection at the peer itself will be the only effective protection.
- All hosts that may be peers will need DNS entries, mapping names to addresses. As more nodes need to engage in peer-to-peer sessions, the number of hosts in the DNS will rise dramatically. Dynamic DNS (DDNS) will likely be needed, and dictionary attacks against node names might be effective. Hiding a host location will be difficult, and protection at the node level will be needed.
- Network Address Translation (NAT) is not defined for IPv6, although sometimes application-level proxies (such as HTTP proxy) allow internal-use-only nodes to gain access to some outside service. In general, many hosts will be reachable from the public network, at least for some specific services. Distributed protection and edge protection (defense in depth) will be crucial.

The figure shows a defense-in-depth arrangement. The edge firewall provides screening services but allows traffic between specific IPv6 hosts, even though it cannot perform deep inspection of the traffic. Deep inspection is done at the internal IPsec endpoint.

Hide Topology when Possible

This topic describes the need to hide topology information when possible, for security.



Topology hiding has benefits for high-security deployments but should not be relied on to protect systems and infrastructure. The security community is divided over whether topology hiding is important.

Topology hiding is the practice of keeping Internet-based systems from determining the internal IP subnet structure of the enterprise.

One advantage of IPv6 networks is that they make it difficult for scanning-based attacks to be successful. Compared to IPv4, in which many deployed networks are /24 networks with 254 possible addresses at which active hosts may be found, a /64 IPv6 prefix is a large space. If the enterprise is given a /48 (65,536 subnets) prefix, simply finding an active subnet presents a challenge for attackers; in a medium-size enterprise, perhaps 50,000 out of 64,000 subnets will be thinly populated. Assigning subnets in a nonobvious sequence, while still allowing use of engineering best practices such as route summarization, can also be part of a topology-hiding strategy. Given these considerations, topology hiding is a practice worth considering.

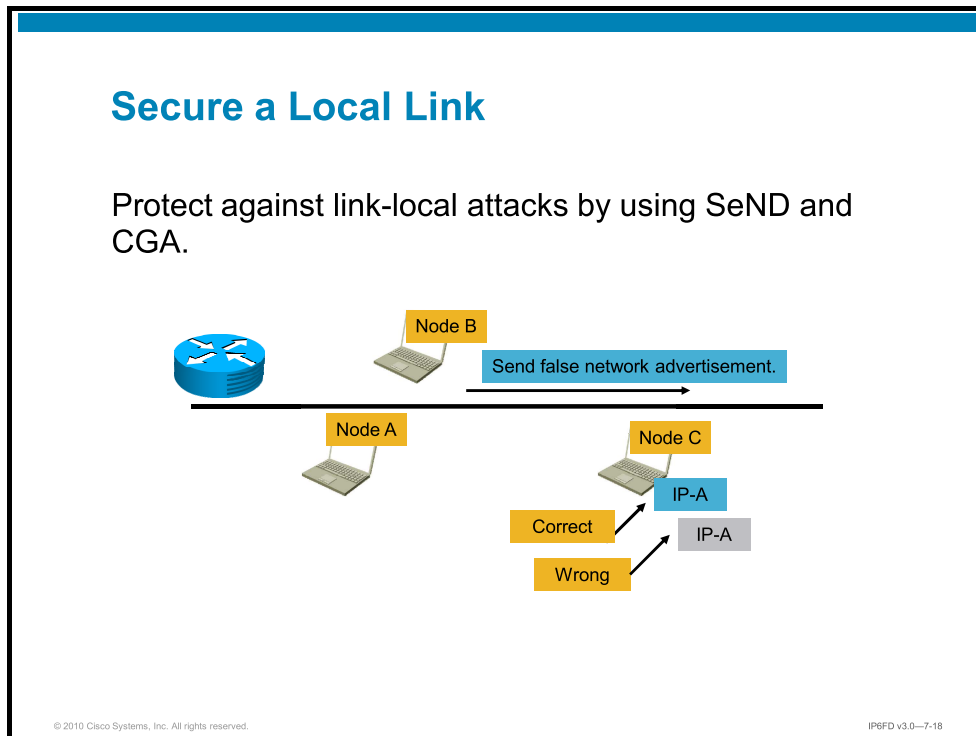
There are various ways to accomplish topology hiding:

- Use universal LAN addressing and proxy servers to provide Internet-based services to internal nodes. This approach works well as long as end-to-end services are not needed.
- Use MultiChannel Interface Processor version 6 (MIPv6). If the enterprise nodes are always mobile, then they can be associated with an IPv6 home agent on a known subnet and can roam internally onto different internal, cloaked subnets. Viewed from the outside, all hosts appear to be on a single or small set of subnets, regardless of the true number of active networks. Route optimization must not be used.

The figure shows the use of MIPv6 to hide the internal topology of the site. The internal mobile node moves from the home agent network to an internal foreign network. When MIPv6 is used, the outside world sees addresses only on the 2001:DB8:8904:A23B::/64 subnet, not on other internal subnets. Access can also be controlled more tightly at the edge screening router. Note that the mobile node and correspondent node can also use IPv6 IPsec for greater security.

Secure the Local Link

This topic describes how to secure a local link.



IP networks, including IPv6, assume a high level of trust on the local subnet.

There are several well-known subnet-local attacks in IPv4, and IPv6 is also vulnerable. For example, with ARP-cache positioning in IPv4, or neighbor cache-positioning in IPv6, an attacker can create an inconsistent mapping in a neighbor cache. This mapping can result in nodes sending packets to the attacker (which acts as a man-in-the-middle) rather than to the authentic destination node.

IPv6 has a new solution to this and related issues. The solution is called Secure Neighbor Discovery (SeND) and Cryptographically Generated Addresses (CGA). This scheme (which is standardized in RFC 3971 and RFC 3972) prevents neighbor-cache positioning. The first node to seize an address and install Layer 2 mapping in the neighbor cache is assured of keeping it. If another node tries to remap the relationship and redirect the packets, the first node detects the attack and does not change the mapping.

SeND and CGA are recently released RFCs, and implementations are uncommon at this time. In the future, however, subnet-local security will be enhanced via these mechanisms, and high-security organizations should consider them for deployment.

In the figure, node C has used neighbor discovery to obtain the Layer 2 address of node A and has placed it into the neighbor cache. Without SeND and CGA, node B (posing as node A) can send a false network advertisement, which node C will pick up and use to update the neighbor cache. Future packets that are sent by node C and intended for node A will be delivered to node B. SeND and CGA would have allowed node C to detect and ignore the false advertisement of node B.

CGA Configuration

- Configuration tasks:
 - General CGA parameters
 - Interface CGA parameters
- General CGA parameters

```
ipv6 nd secured sec-level minimum 1
ipv6 nd secured key-length minimum 1024
```

- Interface CGA parameters

```
interface FastEthernet 0/0
ipv6 cga rsakeypair CGA keys
ipv6 address 2001:db8:AE::/64 cga
```

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-19

CGAs are IPv6 addresses that are generated from the cryptographic hash of a public key and auxiliary parameters. This approach provides a method for securely associating a cryptographic public key with an IPv6 address in the SeND protocol.

The node that generates a CGA must first obtain a Rivest, Shamir, and Adleman (RSA) key pair. (SeND uses an RSA public/private key pair.) The node then computes the interface identifier part (which is the right-most 64 bits) and appends the result to the prefix to form the CGA.

CGA generation is a one-time event. A valid CGA cannot be spoofed, and the parameters that are associated to a particular CGA are reused. The message must be signed with the private key that matches the public key that is used for CGA generation, which only the address owner will have.

A user cannot replay the complete SeND message (including the CGA address, CGA parameters, and CGA signature) because the signature has only a limited lifetime.

Configuring General CGA parameters

This command configures the SeND security level:

ipv6 nd secured sec-level [**minimum value**]

This command configures the SeND key-length options:

ipv6 nd secured key-length [[**minimum** | **maximum**] **value**]

Example

```
ipv6 nd secured sec-level minimum 1
ipv6 nd secured key-length minimum 1024
```

Configuring a Host to Enable SeND

This command specifies which RSA key pair should be used on a specified interface:

ipv6 cga rsakeypair *key-label*

This command configures an IPv6 address, based on an IPv6 general prefix, and enables IPv6 processing on an interface. The **cga** keyword generates a CGA.

ipv6 address {*ipv6-address/prefix-length* [**cga**] [*prefix-name sub-bits/prefix-length* [**cga**]]}

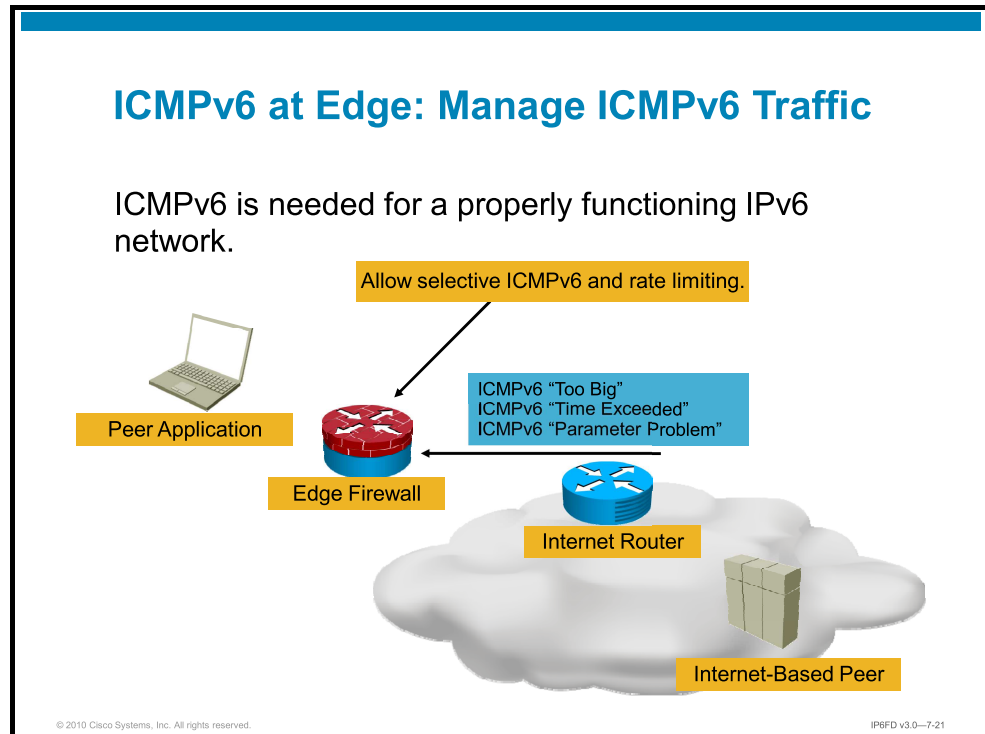
Note The CGA link-local addresses must be configured by using the **ipv6 address link-local** command.

Example

```
interface FastEthernet0/0
  ipv6 cga rsakeypair CGA keys\
  ipv6 address 2001:db8:AE::/64
```

ICMPv6 at Edge—Manage ICMPv6 Traffic

This topic describes the managing of ICMP version 6 (ICMPv6) traffic at the IPv6 edge.



ICMPv6 is important to the health of an IPv6 network. ICMPv6 must be allowed to flow more broadly than it is in IPv4 networks.

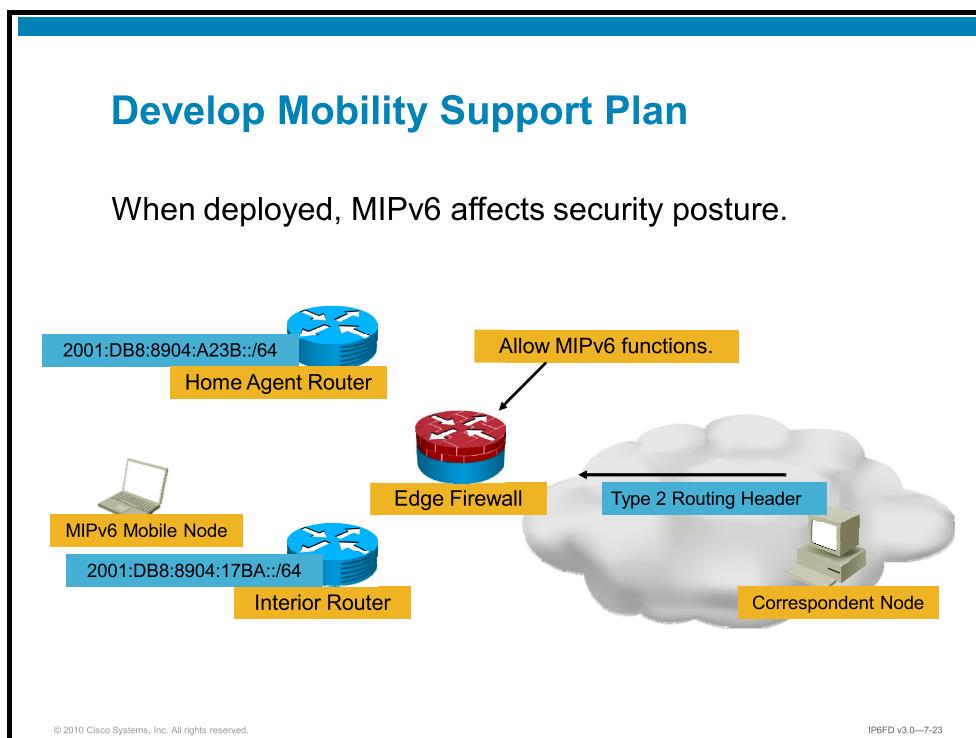
ICMPv6 is used for many crucial functions in IPv6, including the PMTUD function. For nodes that are involved in peer-to-peer sessions across the enterprise edge firewall, being able to discover the appropriate packet size and learning about changes in the path or unreachable nodes is crucial to providing the user with a high-quality experience. The traditional approach to ICMP traffic management—blockading all of it at the network edge—will not be appropriate.

Careful management of specific ICMPv6 message type and code combinations, unidirectional across the network border and at internal nodes that are protected by individual packet filters, will be common in the future. Rate limiting (allowing a prescribed number of messages of a certain type across a point in the network) will be a useful tool.

In the figure, an internal-based peer node needs to receive ICMPv6 error messages that are related to session failure. If the firewall blocks all ICMPv6 messages, the application might not recover gracefully. For the best network health, careful management, coupled with rate limiting, might be an appropriate solution.

Develop Mobility Support Plan

This topic describes the development of a mobility support plan.



MIPv6 requires a Type 2 routing header between Internet-based correspondent nodes and internal mobile nodes.

A common practice on current networks is to block source-routed packets at the enterprise edge; few, if any, current value-providing applications use the source-routing feature of IPv4. However, in IPv6 MIPv6 uses the routing header to deliver packets to the mobile node via the temporary care-of address (CoA) of the mobile node.

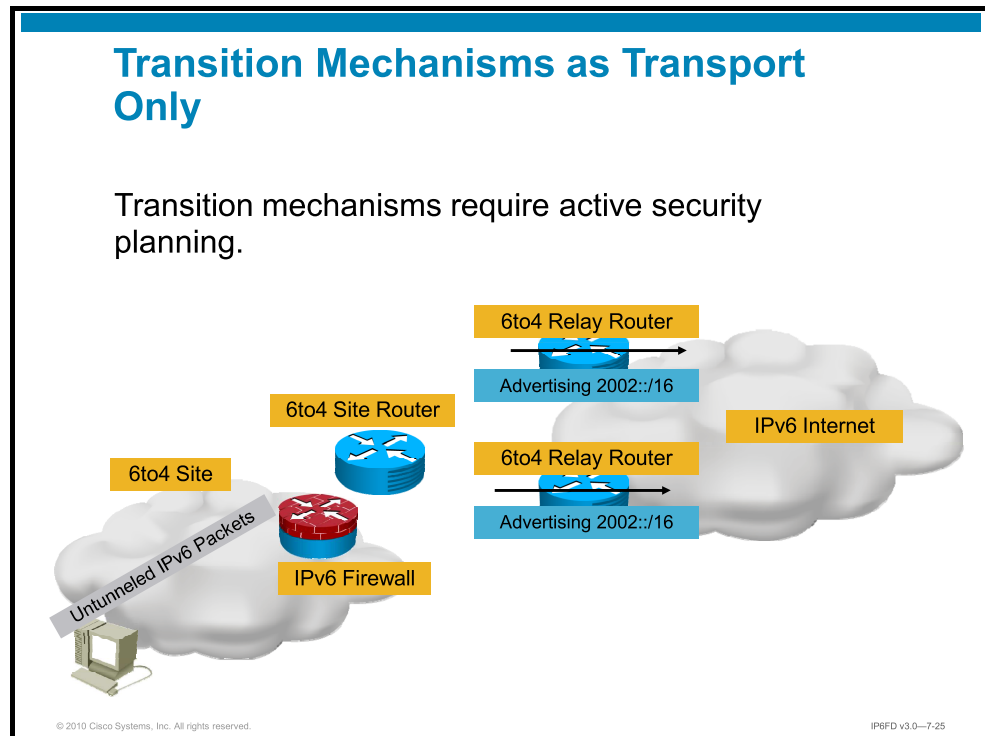
The Type 2 routing header is MIPv6-specific. Whereas the Type 0 routing header allows any number of intermediate hops, the Type 2 header allows only one intermediate hop—the CoA of the mobile node—and requires that the final destination is the home address of the mobile node.

Current generation firewalls do not distinguish between the Type 0 and Type 2 routing headers, but future releases of these devices will distinguish between them. At that time, it will be possible to selectively manage the two types. Understanding the enterprise support for MIPv6, how the protocol works, and engineering a solution that balances security and functionality will be required.

In the figure, a mobile node within an enterprise network exchanges packets with an Internet-based correspondent node. If the mobile node optimizes the path with the correspondent node, then the correspondent node will use the Type 2 routing header to deliver packets. Ideally, the enterprise can selectively manage source-routed packets at the edge.

Use Transition Mechanisms as Transport

This topic describes the use of transition mechanisms as transport.



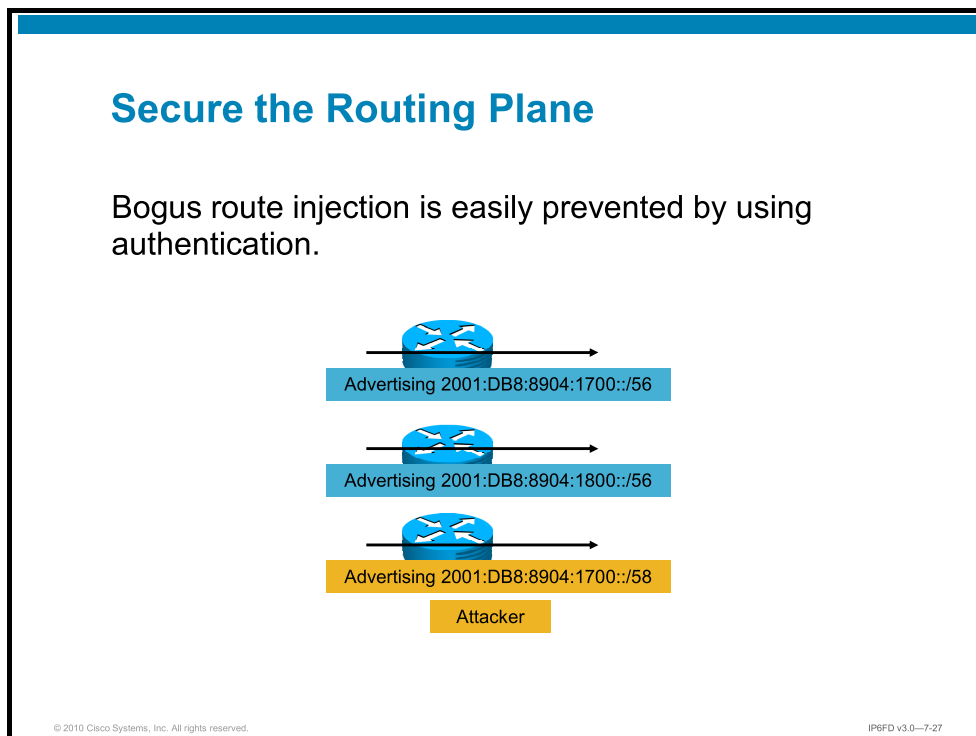
Several transition mechanisms, most notably 6to4 and Teredo, use infrastructure components as tunnel endpoints, in which the tunnels themselves are difficult to secure. For example, 6to4 tunnels traffic that is bound for the IPv6 Internet via anonymous 6to4 relay routers. These routers may be run by different parties on the Internet and should be considered to be public network infrastructure devices by 6to4 sites.

The 6to4 relay router might black-hole traffic, perform poorly, or be unreliable, or the operator might snoop traffic—all risks of any backbone Internet router. In this case, it is impossible to secure the 6to4 system. Instead, directly manage the IPv6 traffic that goes into and comes out of the tunnel, and consider the entire 6to4 infrastructure as simply a transport mechanism with the same security characteristics as the Internet backbone.

In the figure, the 6to4 relay routers are clearly Internet-based infrastructure and should be thought of like any other backbone router. IPv6 should be secured independently of the transport arrangement (in this case, via 6to4). Here, an IPv6-capable firewall is shown inspecting IPv6 traffic after the IPv4 tunnel wrapper is removed.

Secure the Routing Plane

This topic describes securing the routing plane.

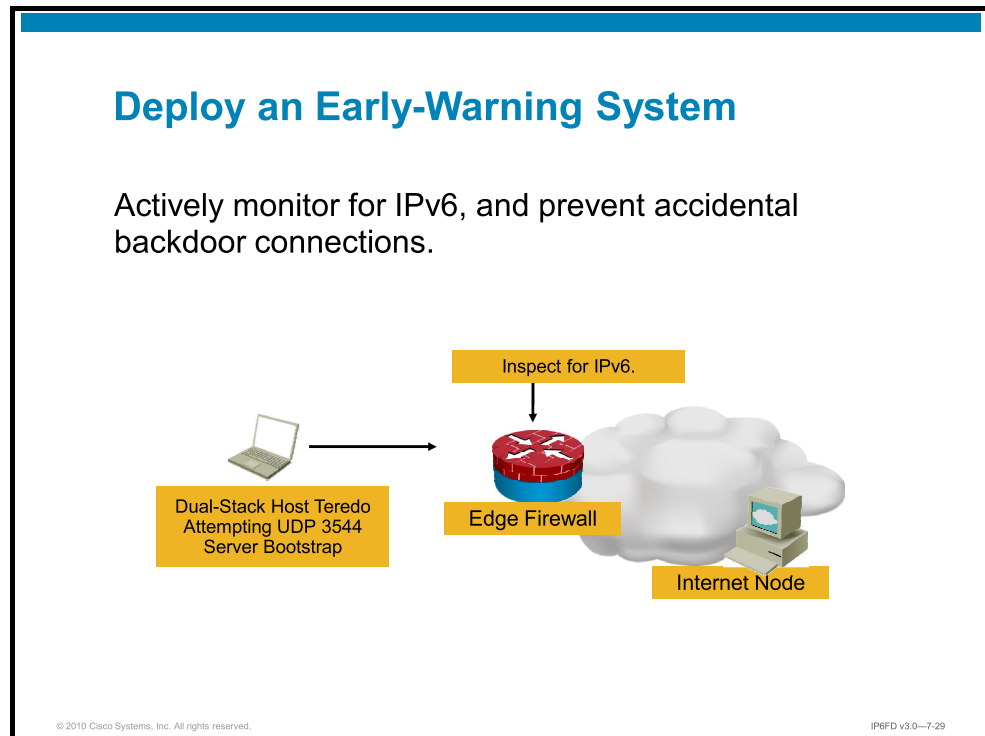


Routing protocol authentication is a well-known practice and should be deployed in most networks. Insider attacks are much more prevalent in large enterprises than are Internet-based attacks. It is not difficult to disrupt operations on a network by injecting spurious routing information. This type of attack is easily defended against by using authenticated routing protocols. Open Shortest Path First version 3 (OSPFv3) provides authentication support via the underlying IPsec mechanism. Enhanced Interior Gateway Routing Protocol (EIGRP) for IPv6 has integrated authentication. In most enterprises, turning on these protections makes sense.

In the figure, two legitimate routers are advertising /56 summary routes within a larger network. An attacker with a router, or probably a PC with a hacking application that manufactures standards-compliant OSPF messages, injects longer routes for a specific prefix. The routing system, if no authentication is performed, will believe that there are now two paths to the 2001:DB8:8904:1700/58 network, and will prefer the longer route. Authenticated routing protocols can blunt this attack.

Deploy an Early-Warning System

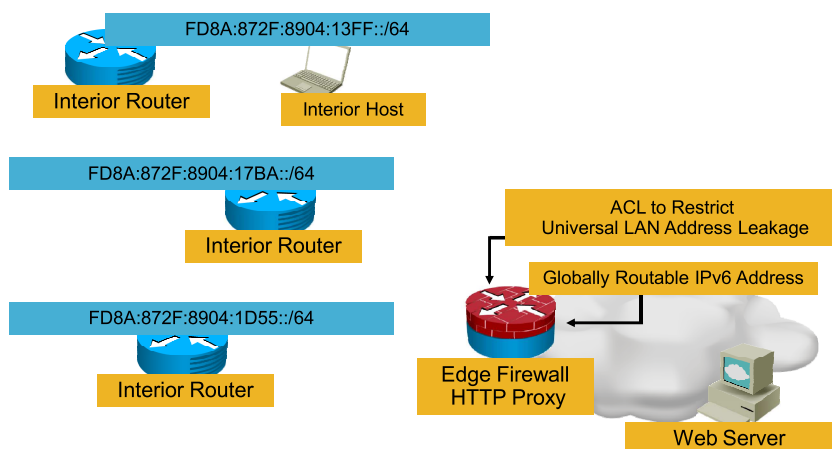
This topic describes an IPv6 early warning system.



Although some organizations are not using IPv6 or have deployed it in certain areas only, IPv6 might be present on other subnets as well. Networks usually contain people and systems from various organizations. Some use IPv6—sending IPv6 traffic (such as Neighbor Solicitations or Router Solicitations). Some IPv6 nodes use transition mechanisms that will attempt to circumvent simple edge firewalls; for example, in the case of Teredo. It would be a serious security concern to have dual-stack systems within the organization successfully obtain native, unfirewalled IPv6 capability. Such a situation can act as a back door to the IPv4 networks of the organization.

In the figure, an internal host has been dual-stacked by the user and is attempting to contact a Teredo server. If the host is successful, the Teredo client will be reachable on IPv6 through the enterprise IPv4 edge firewall.

Use Universal LAN Addresses for Internal-Use-Only Nodes



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-30

Some networks do not need Internet connectivity. Perhaps there is no reason for the network to communicate with the outside world, security policy demands no contact with the Internet, or there is no Internet access at the site.

In such cases, migration to IPv6 might not be mandatory. The size of the network and application support will be the determining factors when making this decision.

IPv6 has a set of addresses (FC00::/7) that are dedicated for use in private networks, as defined in RFC 4193. This block consists of two address groups:

- FC00::/8 is globally allocated in sizes of /48.
- FD00::/8 can be privately assigned by appending a 40-bit, random number. If there is no collision in this 40-bit space, then each network can have a unique /48 prefix, which allows the merging of networks if needed.

Range FC00::/7 is not routed on the Internet. Nodes that use these addresses cannot use Internet services even if they are connected to an Internet access router, because return traffic is never routed back to the network. Outgoing initiating traffic will be sent to the Internet hosts, unless the ISP implements some form of spoof protection. To prevent data leakage, the administrator should implement additional restrictions on the network, such as ACLs that prohibit local traffic from exiting the network.

Summary

This topic summarizes the key points that were discussed in this lesson.

Summary

- IPv6 security practices will be different from longstanding IPv4 practices, because the protocol works differently and offers a number of potential new services.
- Increased use of tunneling and IPsec makes edge firewalls less effective.
- Topology hiding is still possible and is sometimes desirable.
- New tools are becoming available to secure the local link.
- ICMPv6 must be allowed to flow for the network to perform optimally.
- MIPv6 is a powerful protocol with significant security-practice impacts.
- Transition mechanisms are intended to move IPv6 packets in mixed IPv4/IPv6 environments—security is not built in to these tools.
- Authenticating routing protocols is easy to do and provides real benefits (IPv4 and IPv6).
- Actively monitor for IPv6 activity inside the network, and seek to manage bootleg activity.

© 2010 Cisco Systems, Inc. All rights reserved. IPv6FD v3.0—7-31

References

For additional information, refer to this resource:

- Convery, Sean and Darrin Miller. *IPv6 and IPv4 Threat Comparison and Best-Practice Evaluation (v1.0)*. <http://seanconvery.com/v6-v4-threats.pdf>.

Configuring Cisco IOS Firewall for IPv6

Overview

One of the main features of the Cisco IOS Firewall Feature Set is Cisco IOS Firewall, which is formerly known as Context-Based Access Control (CBAC).

In this lesson, you will learn how Cisco IOS Firewall works and how to configure it.

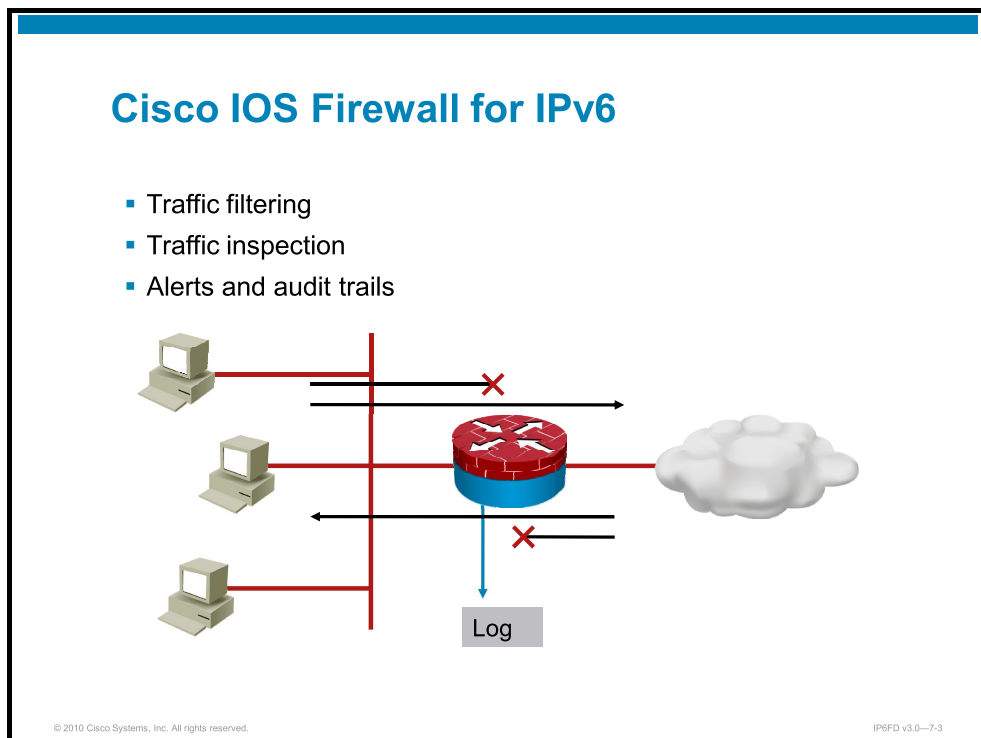
Objectives

Upon completing this lesson, you will be able to describe and configure Cisco IOS Firewall for IP version 6 (IPv6) traffic. This ability includes being able to meet these objectives:

- Describe Cisco IOS Classic Firewall
- Describe how Cisco IOS Classic Firewall works
- Implement IPv6 inspection on Cisco ISRs
- Describe the Zone-Based Policy Firewall for IPv6
- Configure and verify Zone-Based Policy Firewall zones and zone pairs
- Configure and verify a basic OSI Layer 3 to 4 interzone access policy
- List the commands that are used to verify and troubleshoot Cisco IOS Firewall configuration and operation

Cisco IOS Firewall for IPv6

This topic describes Cisco IOS Firewall for IPv6.



Cisco IOS Firewall works to provide network protection on multiple levels by using the following functions:

- Traffic filtering
- Traffic inspection
- Alerts and audit trails

What Cisco IOS Firewall Does Not Do

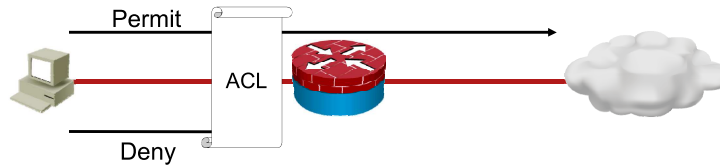
Cisco IOS Firewall does not provide intelligent filtering for all protocols; it works only for the protocols that you specify. If you do not specify a certain protocol for Cisco IOS Classic Firewall, then the existing access control lists (ACLs) determine how that protocol is filtered. No temporary openings are created for protocols that are not specified for Cisco IOS Firewall inspection.

Cisco IOS Firewall does not protect against attacks that originate from within the protected network, unless that traffic travels through a router on which the Cisco IOS Firewall Feature Set is deployed. Cisco IOS Firewall detects and protects only against attacks that travel through the firewall. This scenario is one in which you might want to deploy Cisco IOS Firewall on an intranet-based router.

Cisco IOS Firewall protects against certain types of attacks, but not every type of attack. Cisco IOS Firewall should not be considered a perfect, impenetrable defense. Determined, skilled attackers might be able to launch effective attacks. Although there is no such thing as a perfect defense, Cisco IOS Firewall detects and prevents most of the popular attacks on your network.

Traffic Filtering

- Traffic filtering is done with ACLs.
- All non-stateful rules can be applied.
- ACLs filter only initiating packets.



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6 v3.0—7-4

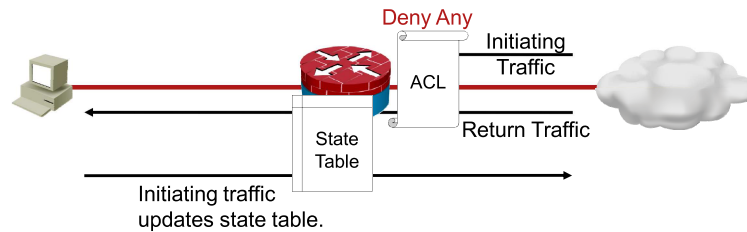
Traffic Filtering

Cisco IOS Classic Firewall intelligently filters TCP and User Datagram Protocol (UDP) packets that are based on application layer protocol session information. You can configure Cisco IOS Classic Firewall to permit specified TCP and UDP traffic through a firewall only when the connection is initiated from within the network that you want to protect. Cisco IOS Classic Firewall can inspect traffic for sessions that originate from either side of the firewall, and Cisco IOS Classic Firewall can be used for intranet, extranet, and Internet perimeters of your network.

Without Cisco IOS Classic Firewall, traffic filtering is limited to ACL implementations that examine packets at the network layer or, at most, the transport layer.

Traffic Inspection

- Inspection engine examines IP packets.
- Inspection engine updates state table.
- Inspection engine permits only return traffic that belongs to existing sessions.
- Return traffic bypasses all ACLs.



Traffic Inspection

Cisco IOS Classic Firewall inspects traffic that travels through the firewall, to discover and manage state information for TCP and UDP sessions. This state information is used to create temporary openings in the firewall, to allow return traffic and additional data connections for permissible sessions.

Inspecting packets at the application layer, and maintaining TCP and UDP session information, gives Cisco IOS Classic Firewall the ability to detect and prevent certain types of network attacks, such as TCP SYN flood attacks. A TCP SYN flood attack occurs when a network attacker floods a server with a barrage of requests for connection and does not complete the connection. The resulting volume of half-open connections can overwhelm the server, causing it to deny service to valid requests. Network attacks that deny access to a network device are called denial of service (DoS) attacks.

Cisco IOS Classic Firewall helps to protect against DoS attacks in other ways. Cisco IOS Classic Firewall inspects packet sequence numbers in TCP connections to determine whether they are within expected ranges and drops any suspicious packets. You can also configure Cisco IOS Classic Firewall to drop half-open connections, which require firewall processing and memory resources to maintain. Additionally, Cisco IOS Classic Firewall can detect unusually high rates of new connections and issue alert messages.

Alerts and Audit Trails

- Automatic alerts can be disabled; they are sent by default.
- Audit trail will generate log entries with session information:
 - Source address and port
 - Destination address and port
 - Amount of data transmitted
- Audit trail messages are sent when connection terminates.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-6

Alerts and Audit Trails

Cisco IOS Classic Firewall also generates real-time alerts and audit trails. Enhanced audit-trail features use syslog to track all network transactions: recording time stamps, source host, destination host, used ports, and the total number of transmitted bytes, for advanced, session-based reporting. Real-time alerts send syslog error messages to central management consoles after detecting suspicious activity.

End-to-End Connectivity

- Stateful inspection breaks end-to-end connectivity.
- Same limitations as IPv4 NAT:
 - Only traffic that belongs to existing sessions is permitted.
 - Outside nodes cannot initiate connections.
 - Layer 7 protocol-specific inspection is required for complex protocols.
- Solution:
 - Stateful inspection is performed only on end hosts.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-7

One of the fundamental advantages of IPv6 over IP version 4 (IPv4) is restoration of end-to-end connectivity. IPv4 has the same capability as IPv6, but networks started introducing Network Address Translation (NAT) because of insufficient addresses. Today, almost all large networks use NAT. Exceptions are some old academic networks, which were assigned addresses before it became apparent that rapid Internet growth would prevent all end hosts from using their own addresses.

With rapid growth, the Internet changed from a safe educational environment to a dangerous public network, where safety from potential threats is not implicit. Corporate networks hid behind firewalls, while smaller networks enjoyed state tracking of NAT.

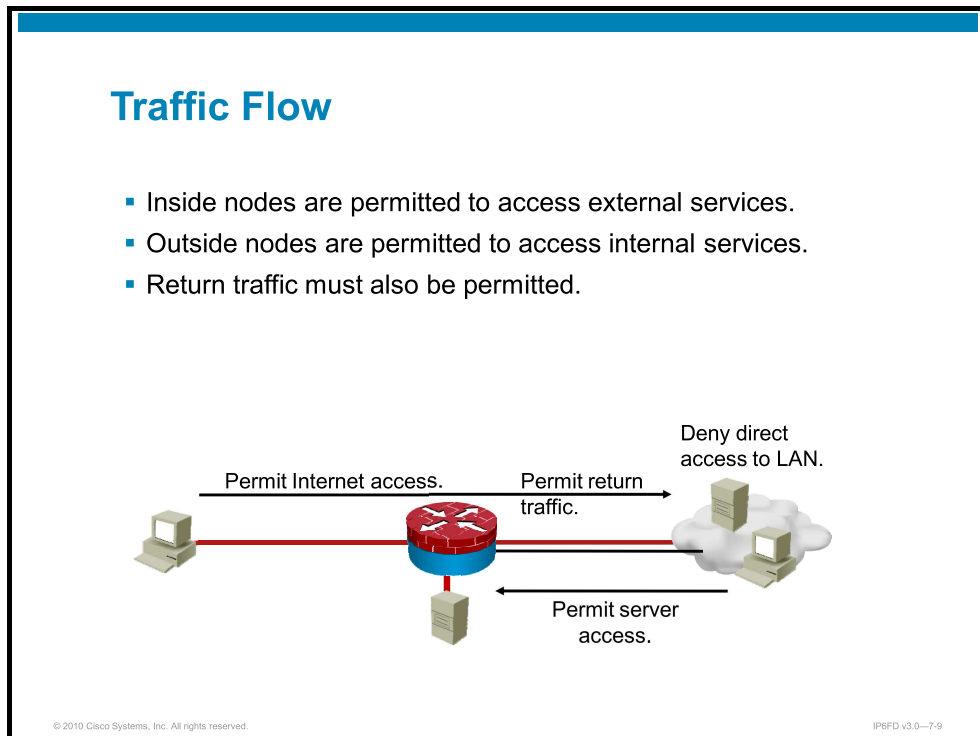
IPv6 restores end-to-end connectivity if and only if the borders of the networks are open. For the majority of corporate networks, this will not be the case, because they will still hide behind firewalls. These networks might no longer use NAT, but because stateful firewalls work basically on the same principle (nothing goes in, unless it is related to something that went out), they will break connectivity.

To provide connectivity for complex protocols that use dynamic port allocation, the firewall must have specific protocol support and must provide packet inspection. This approach consumes resources and might lower performance. However, this approach is the only way to maintain complete application support while still providing network edge firewall protection.

One solution to this problem is the deployment of stateful firewalls on the end hosts. Performance and management are issues that need to be addressed in this case.

IPv6 Inspection on ISRs

This topic describes how Cisco IOS Firewall works in IPv6.

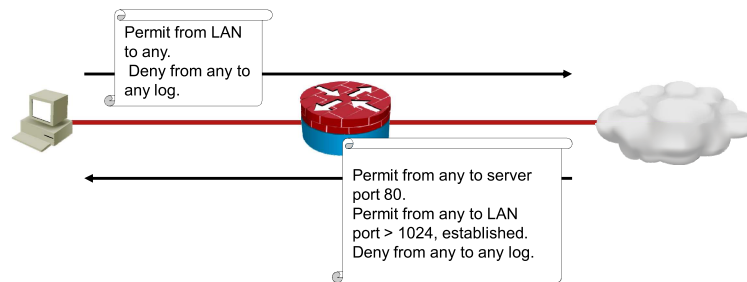


When you define and configure your security policy, you will most often divide your network into different security domains. In a very simple network, there will be just two domains: inside and outside. Usually, level of trust is different in different security domains: You will trust traffic more from one side than from the other.

ACLs are used to restrict traffic. Within ACLs, you define the type of traffic that you will permit to go through the interface. A common scenario is one in which internal hosts on the network are permitted more or less restricted access to the Internet. Policies vary from network to network and range from completely unrestricted access, to access policies that permit only a handful of services, to completely restricted access, in which the only way to connect to the outside host is through a proxy service. In any case, return traffic must be permitted for a service to work. In many cases, networks will permit access to a set of services on an inside host that is specifically designated for external access but will deny direct access to LAN devices.

Stateless ACLs

- Stateless ACLs can be used to implement desired functionality.
- Return traffic must be explicitly permitted.
- Very general rules are required.



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6 v3.0—7-10

One way of implementing security policies is with ACLs alone. This method is completely stateless, which means that the router does not store any state information about sessions that cross it. The greatest benefit of this approach is a considerably smaller performance impact when compared to stateful filtering.

The problem with stateless ACLs is that you need to explicitly deny return traffic. In a TCP session, the three-way handshake is performed first. Then, any traffic that one side sends over must be acknowledged by the other side.

Because a stateless router does not know which traffic belongs to existing sessions (because of a lack of connection state information), the router must permit all traffic that could belong to an existing session.

For TCP, there is a feature of all TCP sessions that can be exploited. All TCP sessions are started with a packet that has the synchronization (SYN) flag turned on and the acknowledgment (ACK) flag turned off. This initiating packet is followed by a SYN-ACK packet, and all subsequent packets, except a possible reset (RST) packet, have the ACK flag set. Therefore, you can safely assume that any packet with ACK or RST flags that are set is part of an existing connection or at least a response to a previous packet.

Cisco ACLs include the **established** keyword, which can be used to define these packets. Therefore, the usual practice is to permit return traffic that is defined by the **established** keyword. SYN packets are not matched and can be denied, thus protecting the inside network from direct connections.

Additionally, because source ports are almost exclusively from the range above 1023, ACLs can also check this parameter.

Any services that run on the inside network and must be reachable from the outside require SYN packets to go through. For these services, you must explicitly permit initiating traffic.

Stateful Packet Inspection

- Initial packets:
 - Must be permitted by ACL
 - Create connection entry in a state table
- Return traffic:
 - Is implicitly allowed
 - Bypasses all ACL restrictions
 - No need to specify in advance

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7.11

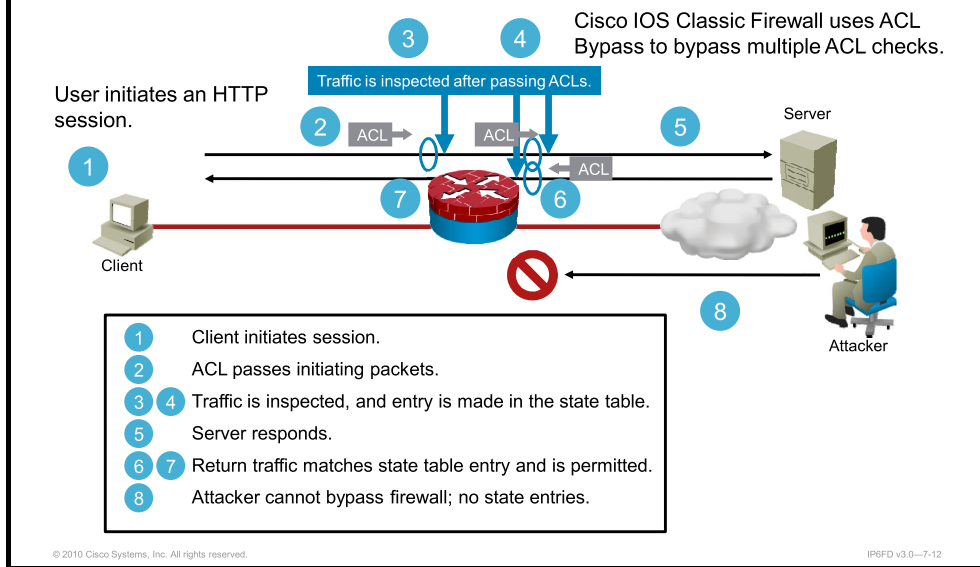
Stateful packet inspection adds connection-tracking capability to the router. With stateful packet inspection, a router will inspect all traffic that passes through it and check whether it is part of an existing connection. If so, the initial packet was permitted by an ACL and all related packets must be implicitly permitted for the session to work. If not, the ACLs are applied and if the traffic is permitted, a state table entry is created.

There are a couple advantages of using stateful packet inspection:

- **Simplified configuration:** Because only initial traffic must be explicitly listed in the ACL, the ACL is shorter, simpler, and easier to read. Less complexity reduces chances for errors.
- **Increased security:** The router now tracks all connections and permits return traffic only if an explicitly permitted initiating packet was sent in the opposite direction. Without stateful checking, any traffic that matches return traffic specification will be passed through and may reach protected internal nodes.

The disadvantage of stateful packet inspection is increased system utilization. All traffic must be checked by the stateful packet inspection engine—a process that can, depending on the platform, significantly reduce the throughput of the router.

Stateful Packet Inspection (Cont.)



This figure shows the actions that the router takes when traffic passes through it and traffic inspection is configured.

1. **Client initiates the connection:** The client sends out the first packet. If this is a TCP packet, then it will have the SYN flag set.
2. **Inbound ACL permits the initiating packet:** The initiating traffic must be explicitly permitted by the ACL, or else the traffic is dropped.
3. **Traffic is inspected after passing through the ACLs:** The traffic must pass all ACLs in the initial direction, or else the traffic is dropped. The traffic must also pass an **inspect** command on one of the interfaces in the right direction. This inspection creates a state table entry and allows related traffic.
4. **Server responds to the packet:** The server receives the initiating packet and responds to it.
5. **Return traffic matches the state table entry and is permitted:** Return packets match an entry in the state table. This entry is checked before any ACLs are processed. Because the traffic is related, it is permitted implicitly without any further checks.
6. **Attacker cannot bypass the firewall:** If a packet arrives at the router and has no matching entry in the state table, then normal ACLs checks are applied. The administrator can block any unwanted traffic by using ACLs. It is even possible to set up a deny-all policy on the outside interface. Although this approach permits all related traffic, nothing is permitted to enter the network from the outside.

TCP Session Inspection

- TCP is connection-oriented.
- TCP flags are checked by stateful packet inspection engine to build state table.
 - SYN, SYN-ACK, or ACK exchange builds connection.
 - FIN and RST tear down connections.
- Stateful packet inspection has the same knowledge of the state of connections as end hosts do.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-13

TCP is a session-oriented protocol. Because each session needs to be set up by using a three-way handshake, firewalls in the traffic path can monitor this handshake and set up a state table, which will store the session data and its state. By doing so, firewalls can maintain accurate information about all sessions that pass through them.

With TCP, source IP and port, destination IP and port, and TCP flags are checked and session state is tracked. SYN and ACK flags are involved with building a connection. RST and FIN flags are used to tear down a connection.

UDP and ICMP Traffic

- Connectionless protocols do not keep track of sessions.
- Hosts do not store the state of the connection.
- For UDP, stateful packet inspection briefly waits for return packets that match the criteria (address, port pairs).
- For ICMP, stateful packet inspection briefly waits for matching packets, based on outgoing traffic.

© 2010 Cisco Systems, Inc. All rights reserved.

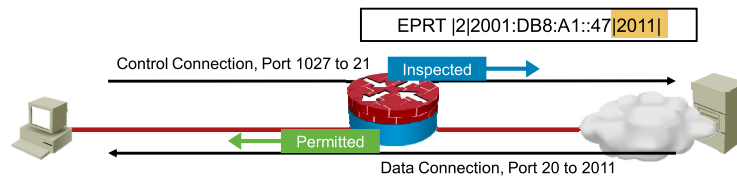
IP6FD v3.0—7-14

Connectionless protocols do not have session states. If a service is listening on a port, then that service can receive and process packets at any time. Because there is no defined state of a connection, firewalls use a trick: They briefly allow any traffic that matches the required criteria. For example, if a node sends out UDP traffic to a Domain Name System (DNS) server, a firewall will permit UDP packets that come from the DNS server, from port 53, to the node to the querying port, if they arrive soon after. The default time is 30 seconds.

An Internet Control Message Protocol (ICMP) inspection session is based on the source address of the inside host that originates the ICMP packet. Dynamic ACLs are created for return ICMP packets of the allowed types (destination unreachable, echo-reply, time-exceeded, and packet too large) for each session.

FTP Inspection

- FTP control session packets are inspected.
- Data connection information is extracted.
- Data connection packets are permitted by stateful packet inspection.
- Mechanism fails if encryption is used.



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-15

Cisco IOS Firewall for IPv6 uses Layer 7 support for applications. At the time of writing this document, the only application that is supported is FTP.

Any application layer protocol that is inspected will take precedence over the TCP or UDP packet inspection. For example, if inspection is configured for FTP, all control channel information will be recorded in the state table, and all FTP traffic will be permitted back through the firewall if the control channel information is valid for the state of the FTP session. The fact that TCP inspection is configured is irrelevant.

Cisco IOS Firewall for IPv6 uses RFC 2428 to garner IPv6 addresses and corresponding ports. If an address other than an IPv6 address is present, the FTP data channel is not opened.

Implement IPv6 Inspection on ISRs

This topic describes implementation of IPv6 inspection on Cisco Integrated Services Routers (ISRs).

Implementing IPv6 Inspection on ISRs

- Identify traffic that will be allowed out through the firewall. Be sure that ACLs permit legitimate traffic from the secure network to the unsecured network.
- Configure ACLs to block traffic from the unsecured network.
- Create inspection rules.
 - Define inspection rules:
 - Generic TCP and UDP
 - FTP inspection
 - Set global timeouts and thresholds (DoS protection).
- Apply the rule inbound to the inside interface or outbound to the outside interface.
- Configure audit trails and logging.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-17

When implementing IPv6 inspection on ISRs, you should follow these steps:

- Step 1** Identify traffic that will be allowed out through the firewall. Be sure that ACLs permit legitimate traffic from the secure network to the unsecured network.
- Step 2** Configure ACLs to block traffic from the unsecured network.
- Step 3** Create inspection rules.
- Define inspection rules:
 - Generic TCP and UDP
 - FTP inspection
 - Set global timeouts and thresholds (DoS protection).
- Step 4** Apply the rule inbound to the inside interface or outbound to the outside interface.
- Step 5** Configure audit trails and logging.

Configuring ACLs

- ACLs, in combination with stateful packet inspection, define initiating packets of the only traffic that can pass through the firewall.
- ACLs should:
 - Permit all initiating packets of permitted traffic
 - Deny everything else
- Apply these ACLs on the interfaces.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-18

For Cisco IOS Classic Firewall to work properly, you need to make sure that IP ACLs are configured appropriately and are applied at the appropriate interfaces.

Follow these two general rules when evaluating your IP ACLs at the firewall:

- **Start with a basic configuration:** A basic initial configuration allows all network traffic to flow from the protected networks to the unprotected networks, while blocking network traffic from any unprotected networks.
- **Permit Cisco IOS Classic Firewall traffic to leave the network through the firewall:** All ACLs that evaluate traffic that leaves the protected network should permit traffic that will be inspected by Cisco IOS Classic Firewall.

Note	If your firewall has only two connections, one to the internal network and one to the external network, then using all inbound ACLs on the external interface works well. This is because packets are stopped before they get a chance to affect the router itself.
-------------	---

Basic ACL Configuration

The first time you configure the Cisco IOS Firewall, it is helpful to start with a basic ACL configuration that makes the operation of the firewall easy to understand without compromising security. The basic configuration allows all network traffic from the protected networks to access the unprotected networks, while blocking network traffic (with some exceptions) from the unprotected networks to the protected networks.

Note	Any firewall configuration depends on your company security policy.
-------------	---

Use the following guidelines for configuring the initial firewall ACLs:

- Do not configure an ACL for traffic from the protected networks to the unprotected networks. (All traffic from the protected networks can flow through the interface.) This approach helps to simplify firewall management by reducing the number of ACLs that are applied at the interfaces. This configuration assumes a high level of trust for the users on the protected networks, and it assumes that there are no malicious users on the protected networks who might launch attacks from the inside. You can fine-tune network access for users on the protected networks as you gain experience with ACL configuration and the operation of the firewall.
- Configure an ACL that includes entries that permit certain ICMP traffic from unprotected networks.

Although an ACL that denies all IP traffic that is not part of a connection that is inspected by Cisco IOS Classic Firewall seems most secure, it is not practical for normal operation of the router. The router expects to see ICMP traffic from other routers in the network. Additionally, if ICMP traffic is not inspected by Cisco IOS Classic Firewall, then specific entries are needed in the ACL to permit return traffic for ICMP commands. For example, a user on a protected network uses the **ping** command to get the status of a host on an unprotected network; without entries in the ACL that permit echo-reply messages, the user on the protected network gets no response to the **ping** command.

Include ACL entries to permit the following ICMP messages:

- **Echo reply:** Outgoing **ping** commands require echo-reply messages to come back.
 - **Time-exceeded:** Outgoing **traceroute** commands require time-exceeded messages to come back.
 - **Packet-too-big:** Path Maximum Transmission Unit Discovery (PMTUD) requires packet-too-big messages to come back.
 - **Traceroute:** An ICMP traceroute message allows an incoming traceroute.
 - **Unreachable:** ICMP unreachable messages permit all unreachable messages to come back. If a router cannot forward or deliver a datagram, it sends an ICMP unreachable message back to the source and drops the datagram.
 - **Neighbor solicitation and neighbor advertisement:** These messages are needed for address resolution between Layers 2 and 3. These messages are implicitly permitted by IPv6 ACLs but may be denied if there is a "deny any any" statement before the end of the ACL.
- Add an ACL entry that denies any network traffic from a source address that matches an address on the protected network.

Note	This approach is known as antispoofing protection because it prevents traffic from an unprotected network from assuming the identity of a device on the protected network.
-------------	--

- By default, the last entry in an extended ACL is an implicit denial of all IP traffic that is not specifically allowed by other entries in the ACL, except for neighbor solicitation and neighbor advertisement.

External Interface ACL

Follow these guidelines for your ACLs when you configure Cisco IOS Classic Firewall on an external interface:

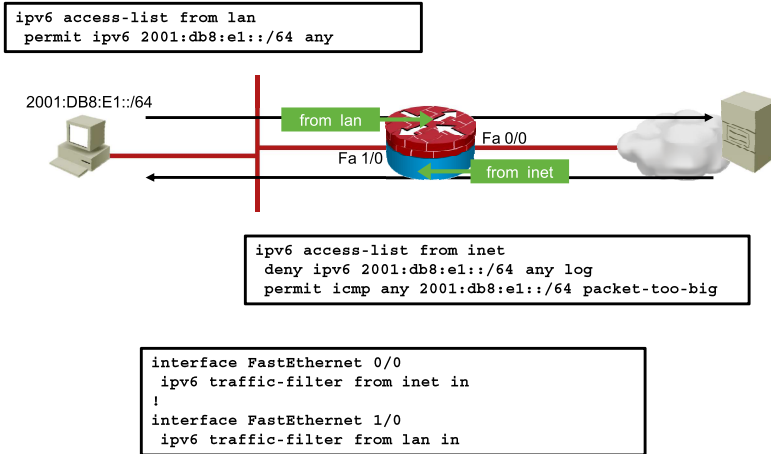
- If you have an outbound IP ACL at the external interface, the ACL can be a standard or extended ACL. This outbound ACL should permit traffic that you want to be inspected by Cisco IOS Classic Firewall. If traffic is not permitted, then it will not be inspected by Cisco IOS Classic Firewall, but will simply be dropped.
- The inbound IP ACL at the external interface must be an extended ACL. This inbound ACL should deny traffic that you want to be inspected by Cisco IOS Classic Firewall. (Cisco IOS Classic Firewall will create temporary openings in this inbound ACL, as appropriate to permit only return traffic that is part of a valid, existing session.)

Internal Interface ACL

Follow these guidelines for your ACLs when you configure Cisco IOS Classic Firewall on an internal interface:

- If you have an inbound IP ACL at the internal interface (or an outbound IP ACL at external interfaces), these ACLs can be either a standard or extended. These ACLs should permit traffic that you want to be inspected by Cisco IOS Classic Firewall. If traffic is not permitted, then it will not be inspected by Cisco IOS Classic Firewall, but will simply be dropped.
- The outbound IP ACL at the internal interface (or an inbound IP ACL at the external interface) must be extended ACLs. These access lists should deny traffic that you want to be inspected by Cisco IOS Classic Firewall. Cisco IOS Classic Firewall will create temporary openings in these outbound ACLs, as appropriate to permit only return traffic that is part of a valid, existing session. You do not necessarily need to configure an extended ACL at both the outbound internal interface and the inbound external interface, but at least one is necessary to restrict traffic that flows through the firewall into the internal protected network.

Configuring ACLs: Example



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-19

The figure presents an example of ACL configuration. The router has two interfaces: Fast Ethernet 1/0 on the LAN side and Fast Ethernet 0/0 on the Internet side. Two ACLs are configured. The first ACL is named `from lan` and permits the LAN IP range to access any other address. The second ACL is named `from inet` and denies the LAN IP range (a form of spoof protection) and permits any host to send ICMP packet-too-large messages to LAN hosts.

Inspection Commands

Router(config)#

```
ipv6 inspect name <inspection> <protocol> [alert  
{on|off} [audit-trail {on|off}] [timeout <seconds>]
```

- For the protocol to be inspected, an inspection must be configured for it.

Router(config-if)#

```
ipv6 inspect <inspection> {in|out}
```

- Inspection must be configured on an interface in the direction of initiating traffic:
 - Inside direction on the incoming interface
 - Outside direction on the outgoing interface

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-20

To configure inspection on a router, you must use two commands.

ipv6 inspect name

To define a set of IPv6 inspection rules, use the **ipv6 inspect name** command in global configuration mode. To remove the inspection rule for a protocol or to remove the entire set of inspection rules, use the **no** form of this command.

ipv6 inspect name *inspection-name protocol* [**alert** {**on** | **off**}] [**audit-trail** {**on** | **off**}] [**timeout** *seconds*]

To define a set of inspection rules, enter this command for each protocol that you want the Cisco IOS Firewall to inspect, using the same inspection name. Give each set of inspection rules a unique inspection name, which should not exceed the 16-character limit. Define either one or two sets of rules per interface. You can define one set to examine both inbound and outbound traffic, or you can define two sets: one for outbound traffic and one for inbound traffic.

To define a single set of inspection rules, configure inspection for all the desired application layer protocols and for TCP, UDP, or ICMP, as desired. This combination of TCP, UDP, and application layer protocols forms a single set of inspection rules with a unique name. (No application layer protocols are associated with ICMP.)

To remove the inspection rule for a protocol, use the **no** form of this command with the specified inspection name and protocol. To remove the entire set of named inspection rules, use the **no** form of this command with the specified inspection name.

In general, when inspection is configured for a protocol, return traffic that enters the internal network is permitted only if the packets are part of a valid, existing session for which state information is being maintained.

ipv6 inspect

To apply a set of inspection rules to an interface, use the **ipv6 inspect** command in interface configuration mode. To remove the set of rules from the interface, use the **no** form of this command.

ipv6 inspect *inspection-name* {**in** | **out**}

Use this command to apply a set of inspection rules to an interface.

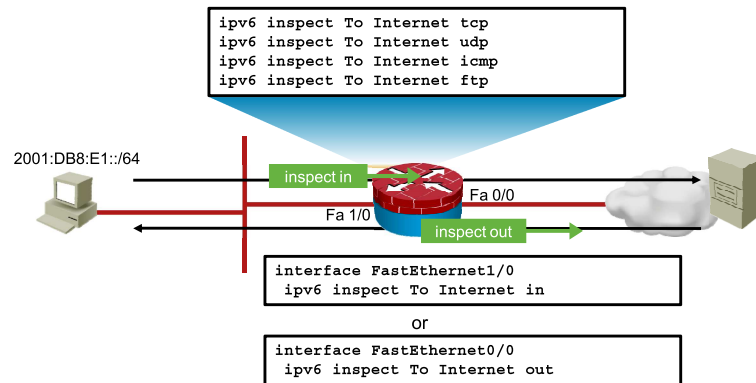
Typically, if the interface connects to the external network, apply the inspection rules to outbound traffic. If the interface connects to the internal network, apply the inspection rules to inbound traffic.

If you apply the rules to outbound traffic, then return inbound packets are permitted if they belong to a valid connection with existing state information. This connection must be initiated with an outbound packet.

If you apply the rules to inbound traffic, then return outbound packets are permitted if they belong to a valid connection with existing state information. This connection must be initiated with an inbound packet.

Inspection Configuration

- Define inspection.
- Apply inspection to an interface in initiating traffic direction.



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-21

The figure presents an example of inspection configuration on a router. Globally, there is one inspection, To Internet, which inspects TCP, UDP, ICMP, and FTP. This inspection can be applied either on the inside interface in the inbound direction or the outside interface in the outbound direction. The direction is important, because the traffic going out of the LAN needs to be inspected.

IPv6 Routing Header

- By default, packets that contain IPv6 routing header are dropped.

```
ipv6 inspect routing-header
```

- Enables processing of IPv6 routing header

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--7-22

To specify whether CBAC should inspect packets that contain an IPv6 routing header, use the **ipv6 inspect routing-header** command. To drop packets that contain an IPv6 routing header, use the **no** form of this command.

ipv6 inspect routing-header

An IPv6 source uses the routing header to list one or more intermediate nodes to be visited between the source and destination of the packet. The Cisco IOS Firewall uses this header to retrieve the destination host address. Cisco IOS Firewall will establish the appropriate inspection session, based on the retrieved address from the routing header.

The originating node lists all intermediate nodes that the packet must traverse. The source and destination address pair in the IPv6 header identifies the hop between the originating node and the first intermediate node. When the first intermediate node receives the packet, the node looks for a routing header. If the routing header is present, the next intermediate node address is swapped with the destination address in the IPv6 header, and the packet is forwarded to the next intermediate node. This sequence continues for each intermediate node that is listed in the routing header, until no more entries exist. The last entry in the routing header is the final destination address.

Troubleshooting

show ipv6 inspect

- **name** *inspection-name*
- **config**
- **interfaces**
- **session** [**detail**]
- **all**

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-23

To view CBAC configuration and session information, use the **show ipv6 inspect** command in privileged EXEC mode:

show ipv6 inspect {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all**}

The **name** *inspection-name* keyword and argument display the configured inspection rule with the name *inspection-name*.

The **config** keyword displays the complete Cisco IOS Firewall inspection configuration.

The **interfaces** keyword displays the interface configuration with respect to applied inspection rules and ACLs.

The **session** keyword displays existing sessions that are currently being tracked and inspected by Cisco IOS Firewall. The optional **detail** keyword causes additional details about these sessions to be shown.

The **all** keyword displays all Cisco IOS Firewall configurations and all existing sessions that Cisco IOS Firewall is tracking and inspecting.

Troubleshooting (Cont.)

debug ipv6 inspect

- **function-trace**
- **object-creation**
- **object-deletion**
- **events**
- **timers**
- **protocol**
- **detailed**

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-24

To display messages about Cisco IOS Firewall events, use the **debug ipv6 inspect** command in privileged EXEC mode. To disable debugging output, use the **no** form of this command.

debug ipv6 inspect {function-trace | object-creation | object-deletion | events | timers | protocol | detailed}

The **function-trace** keyword displays messages about software functions that the Cisco IOS Firewall calls.

The **object-creation** keyword displays messages about software objects that the Cisco IOS Firewall creates. Object creation corresponds to the beginning of Cisco IOS Firewall inspection sessions.

The **object-deletion** keyword displays messages about software objects that the Cisco IOS Firewall deletes. Object deletion corresponds to the closing of Cisco IOS Firewall inspection sessions.

The **events** keyword displays messages about Cisco IOS Firewall software events, including information about Cisco IOS Firewall packet processing.

The **timers** keyword displays messages about Cisco IOS Firewall timer events, such as when a Cisco IOS Firewall idle timeout is reached.

The **protocol** keyword displays messages about Cisco IOS Firewall inspected protocol events, including details about the protocol packets.

Use the **detailed** keyword in this command with other Cisco IOS Firewall debugging commands. This keyword causes detailed information to be displayed for all the other enabled Cisco IOS Firewall debugging.

Zone-Based Policy Firewall for IPv6 on ISRs

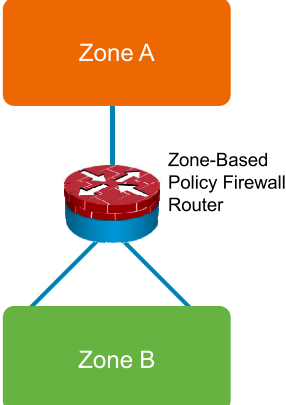
This topic describes the Zone-Based Policy Firewall for IPv6.

Deploying Basic Zone-Based Policy Firewall

Zone-Based Policy Firewall Overview

The Cisco IOS Software Zone-Based Policy Firewall is a feature that allows a router to act as a stateful firewall between zones (i.e., security domains):

- A Zone-Based Policy Firewall **zone** is a collection of networks that are reachable over one or a specific set of router interfaces.
- Zone-Based Policy Firewall allows you to configure **access rules between each zone** pair that is available on the router.



The diagram illustrates the Zone-Based Policy Firewall architecture. It shows two zones, Zone A (orange box) and Zone B (green box), connected by a central router labeled 'Zone-Based Policy Firewall Router' (red and blue box). The router is positioned between the two zones, with lines connecting it to both Zone A and Zone B, representing the firewall's role in controlling traffic between these security domains.

© 2010 Cisco Systems, Inc. All rights reserved. IPv6FD v3.0—7-26

The Zone-Based Policy Firewall is a Cisco IOS Software data plane security feature that allows a router to act as a powerful, flexible, stateful firewall between zones, which correspond to security domains that are created through network separation. The router that is configured with the Zone-Based Policy Firewall can act either as a standalone firewall system or as an element of a more complex firewall system.

A Zone-Based Policy Firewall zone is a collection of networks that are reachable over one or more specific router interfaces and that are designated to belong to the same zone. Zone-Based Policy Firewall access control policies then control access between two or more zones that are configured on the router. These policies use a flexible configuration language that allows you to specify simple or complex access policies in a manageable manner.

Deploying Basic Zone-Based Policy Firewall (Cont.)

Zone-Based Policy Firewall vs. Cisco IOS Classic Firewall (CBAC)

Zone-Based Policy Firewall	Cisco IOS Classic Firewall (CBAC)
Is based on stateful packet filtering and application inspection and control technology	Is based on stateful packet filtering and application inspection and control technology
Controls bidirectional access between zones	Controls inbound and outbound access on an interface
Uses the C3PL configuration language	Uses inspect statements and stateful ACLs
Provides simple configuration of routers with more than two zones	Can become complex in configuration of routers with more than two interfaces
Is supported from Cisco IOS Software Release 12.4(6)T	Is supported from Cisco IOS Software Release 11.2

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-27

Both the Cisco IOS Classic Firewall (CBAC) feature and the Zone-Based Policy Firewall feature are based on stateful packet filtering technology, enhanced with application inspection and control features.

The original implementation of Cisco IOS Classic Firewall stateful inspection used an interface-based configuration model, in which a stateful inspection policy was applied to an interface by using inspect statements and ACLs. All traffic that passed through that interface received the same inspection policy. This configuration model limited the granularity of the firewall policies and caused confusion of their proper application, particularly in scenarios in which the policies must be applied between multiple interfaces. Cisco IOS Classic Firewall support started in Cisco IOS Software Release 11.2.

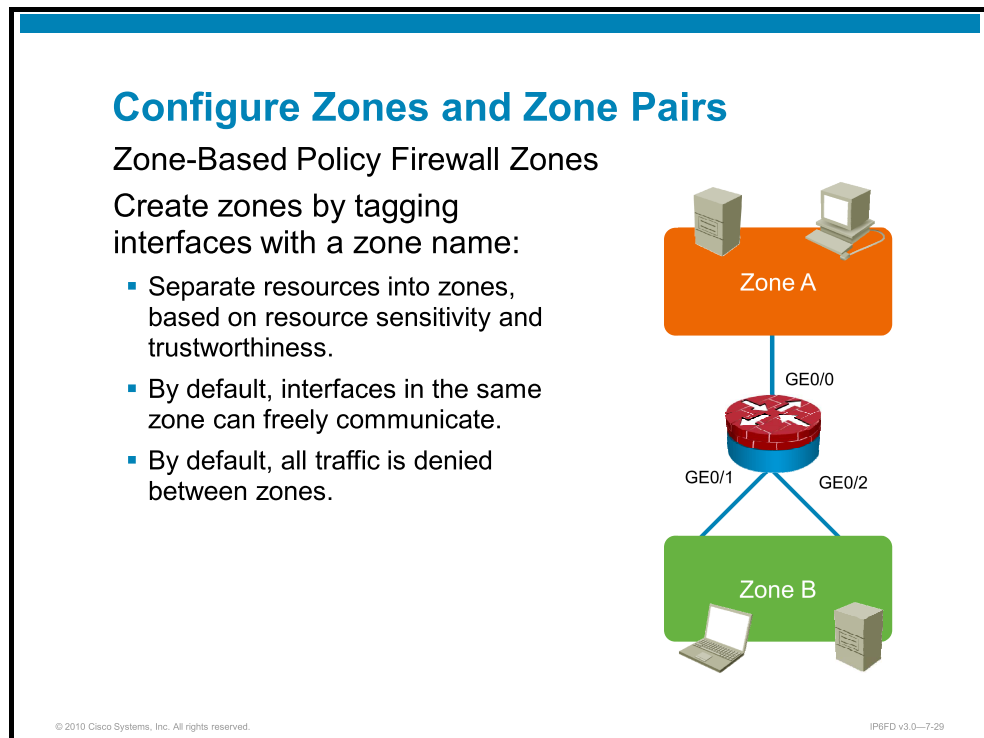
The Zone-Based Policy Firewall changes the firewall from the older interface-based model to a more flexible, more easily understood, zone-based configuration model. Interfaces are assigned to zones, and an inspection policy is applied to traffic that moves between the zones. Interzone policies offer considerable flexibility and granularity, which enables you to apply different inspection policies to multiple host groups that are connected to the same router interface. Zone-Based Policy Firewall is configured with the Cisco Common Classification Policy Language (C3PL), which employs a hierarchical structure to define inspection for network protocols and the groups of hosts to which the inspection will be applied. Zone-Based Policy Firewall is supported from Cisco IOS Software Release 12.4(6)T.

You can use the two configuration models concurrently on routers but not on the same interfaces; you cannot configure an interface as both a security zone member and for CBAC.

Note Interface ACLs are still relevant and are applied before the Zone-Based Policy Firewall when they are applied inbound. Interface ACLs are applied after the Zone-Based Policy Firewall when they are applied outbound.

Configuring Zones and Zone Pairs

This topic enables you to configure and verify Zone-Based Policy Firewall zones and zone pairs.



On a Zone-Based Policy Firewall-enabled router, you must create zones by tagging router network interfaces with a zone name. You should create as many zones as are needed by your access control requirements. Typically, zones (security domains) are created by grouping resources that have the same level of sensitivity or trustworthiness, or that require special isolation because of their criticality or vulnerability.

When you designate router interfaces as zone members, the default policy of the Zone-Based Policy Firewall is to permit all traffic between interfaces of the same zone (that is, intrazone traffic). Between different zones, the Zone-Based Policy Firewall denies all traffic by default.

The figure shows a Cisco ISR that is configured with three interfaces. Interface Gigabit Ethernet 0/0 (and all networks that are reachable over it) belongs to Zone A; interfaces Gigabit Ethernet 0/1 and Gigabit Ethernet 0/2 (and all networks that are reachable over them) belong to Zone B. By default, all traffic between interfaces Gigabit Ethernet 0/1 and Gigabit Ethernet 0/2 can flow freely, whereas traffic between interfaces Gigabit Ethernet 0/0 and Gigabit Ethernet 0/1 and between interfaces Gigabit Ethernet 0/0 and Gigabit Ethernet 0/2 is denied by default, and will need to be explicitly authorized by Zone-Based Policy Firewall rules.

Configure Zones and Zone Pairs (Cont.)

Configuration Tasks

1. Create zones.
2. Assign router interfaces to zones.
3. Create zone pairs.

© 2010 Cisco Systems, Inc. All rights reserved.

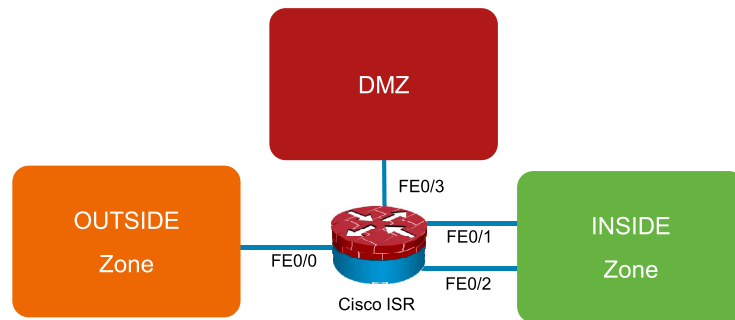
IPv6FD v3.0—7-30

To configure Zone-Based Policy Firewall zones on a Cisco ISR that runs Cisco IOS Software, you will need to perform the following configuration tasks:

- Step 1** Create security zones in Cisco IOS Software.
- Step 2** Assign router network interfaces to the configured zones.
- Step 3** Create zone pairs for each zone that requires unidirectional communication.

Configure Zones and Zone Pairs (Cont.)

Configuration Scenario



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-31

This figure presents the configuration scenario that is used in upcoming configuration tasks. You will create three zones, named OUTSIDE, DMZ, and INSIDE. The Cisco ISR uses four interfaces: interface Fast Ethernet 0/0 will be assigned to the OUTSIDE zone, interface Fast Ethernet 0/3 will be assigned to the DMZ zone, and interfaces Fast Ethernet 0/1 and Fast Ethernet 0/2 will be assigned to the INSIDE zone.

Configure Zones and Zone Pairs (Cont.)

Configuration

```
zone security INSIDE
zone security DMZ
zone security OUTSIDE
!
interface FastEthernet0/0
  zone-member security OUTSIDE
interface FastEthernet0/1
  zone-member security INSIDE
interface FastEthernet0/2
  zone-member security INSIDE
interface FastEthernet0/3
  zone-member security DMZ
!
zone-pair security IN-TO-OUT source INSIDE destination OUTSIDE
zone-pair security OUT-TO-DMZ source OUTSIDE destination DMZ
zone-pair security OUT-TO-IN source OUTSIDE destination INSIDE
```

Create named zones.

Assign router interfaces to zones.

Create only zone pairs that require interaction.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-32

In the first configuration task of this configuration sequence, you will globally create zones on the router. Use the **zone security** command to create a named zone. This example uses the INSIDE, DMZ, and OUTSIDE zones.

zone security

To create a security zone, use the **zone security** command in global configuration mode. To delete a security zone, use the **no** form of this command.

zone security {*zone-name* | **default**}

zone security Parameters

Parameter	Description
<i>zone-name</i>	Name of the security zone. You can enter as many as 256 alphanumeric characters.
default	Name of a default security zone. Interfaces that are not configured on any of the security zones belong to the default zone.

In the second configuration task, assign router network interfaces to zones. You do not need to assign all local interfaces to configured zones; assign only interfaces that you want to participate in the Zone-Based Policy Firewall system. To assign an interface to a zone, enter interface configuration mode and use the **zone-member security** command to specify the name of the zone to which the interface is assigned. You can assign a network interface to one zone only.

The Default Zone

All interfaces that are not tagged with a zone name become members of a system-defined **default** zone. You can use the default zone to establish an access policy between interfaces that are members of the Zone-Based Policy Firewall configuration system and all other interfaces on the router.

zone-member security

To attach an interface to a security zone, use the **zone-member security** command in interface configuration mode. To detach the interface from a zone, use the **no** form of this command.

zone-member security *zone name*

zone-member security Parameters

Parameter	Description
<i>zone name</i>	Name of the security zone to which an interface is attached.

A zone pair allows you to specify that a unidirectional firewall policy between two security zones will exist and will be configured by using C3PL. The direction of the traffic is indicated by specifying a source and destination security zone. If you want traffic to flow in both directions between two zones, you must create two zone pairs: one for each direction.

Create zone pairs by using the **zone-pair security** command, naming the zone pair, and specifying the source and destination zones, with respect to the direction in which application sessions will establish between zones. Only create zone pairs that will unidirectionally interact. If you do not specify a zone pair for a possible pair of zones, all sessions will be denied between these two zones (in that direction) by default.

zone-pair security

To create a zone pair, use the **zone-pair security** command in global configuration mode. To delete a zone pair, use the **no** form of this command.

zone-pair security *zone-pair-name* **source** {*source-zone-name* | **self** | **default**} **destination** {*destination-zone-name* | **self** | **default**}

zone-pair security Parameters

Parameter	Description
<i>zone-pair-name</i>	Name of the zone being attached to an interface.
source <i>source-zone-name</i>	Name of the router from which traffic originates.
default	Name of the default security zone. Interfaces without configured zones belong to the default zone.
destination <i>destination-zone-name</i>	Name of the router to which traffic is bound.
self	The system-defined zone. Indicates whether traffic will be going to or from a router.

Configure Zones and Zone Pairs (Cont.)

Verification

```
Router# show zone security

zone self
  Description: System defined zone

zone DMZ
  Member Interfaces:
    FastEthernet0/3

zone INSIDE
  Member Interfaces:
    FastEthernet0/1
    FastEthernet0/2

zone OUTSIDE
  Member Interfaces:
    FastEthernet0/0
```

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-33

You can use the **show zone security** command to display the zone description and the interfaces that are contained in the specified zone. If you do not specify the zone name, the command displays information for all of the zones that are configured.

In the output in the figure, interface Fast Ethernet 0/3 belongs to the DMZ zone, interfaces Fast Ethernet 0/1 and Fast Ethernet 0/2 belong to the INSIDE zone, and interface Fast Ethernet 0/0 belongs to the OUTSIDE zone.

show zone security

To display zone security information, use the **show zone security** command in user EXEC or privileged EXEC mode.

show zone security [*security-zone-name*]

show zone security Parameters

Parameter	Description
<i>security-zone-name</i>	(Optional) Name of the security zone.

Configuring a Basic OSI Layer 3 to 4 Interzone Access Policy

This topic enables you to configure and verify a basic Open Systems Interconnection (OSI) Layer 3 to 4 interzone access policy.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy

C3PL

C3PL is a Cisco IOS Software configuration language designed to:

- Classify traffic into classes
- Apply actions to traffic
- Allow modular reuse of objects
- Allow simpler creation of complex policies

Three main configuration constructs:

- **Class maps** provide traffic classification.
- **Policy maps** associate actions with traffic classes.
- **Parameter maps** tune inspection parameters for a class.

© 2010 Cisco Systems, Inc. All rights reserved.IP6FD v3.0—7-35

The second major change that is introduced by the Zone-Based Policy Firewall feature is the introduction of a new configuration policy language that is known as C3PL. The C3PL structure is like the Modular quality of service (QoS) command-line interface (CLI) (MQC) structure, in which class maps specify the traffic that is affected by the action that the policy map applies. C3PL is also highly modular, allowing reuse of configured objects in multiple policies, and flexible, allowing creation of complex policies in a manageable fashion.

C3PL uses three main configuration constructs:

- Class maps are objects that describe different classes of traffic and therefore provide classification of traffic, based on administrator-specified criteria.
- Policy maps specify the actions to take when traffic matches the criteria that are defined in class maps. For a router to use the information in a policy map and its associated class maps, the policy map must be associated with a zone pair.
- Parameter maps augment basic inspection with the capability to recognize and apply service-specific actions, such as tuning of the stateful engine or applying application layer inspection.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Class Maps

Class maps describe and group traffic into **classes**:

- Based on one or more **match** conditions
- Can match based on ACL, PAM protocol, or another class map
- Conditions can use OR (default), AND, and NOT operators

Telnet Traffic Between
2001:DB8:1::1 and 2001:DB8:2::2

FTP Traffic Between
2001:DB8:1::1 and 2001:DB8:2::2

Any Telnet, FTP, or HTTP Traffic

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-36

Class maps define the traffic that a Cisco IOS Software Zone-Based Policy Firewall selects for policy application. Basic class maps classify network traffic, based on the following criteria:

- **Access-group:** A standard, extended, or named ACL can select traffic, based on source and destination IP address, source and destination port, and other criteria that are available in the ACL. A permit action in the ACL selects traffic to belong to the class.
- **Protocol:** A protocol can select traffic, based on OSI transport layer protocols such as TCP, UDP, and ICMP, as well as application layer protocols such as HTTP, FTP, and DNS. Any well-known or user-defined service that is known to Cisco IOS Software Port-to-Application Mapping (PAM) can be specified.
- **Class map:** You can nest a subordinate class map that provides additional match criteria inside another class map, creating a hierarchy of classification.

Class maps can apply match-any (logical OR, which is the default) or match-all (logical AND) operators to determine how to apply the match criteria, if multiple criteria are specified. If match-any is specified, traffic must meet at least one of the match criteria in the class map. If match-all is specified, traffic must match all of the criteria in the class map to belong to that particular class. You can also use the negation (NOT) operator to describe traffic, based on traffic not having a particular property.

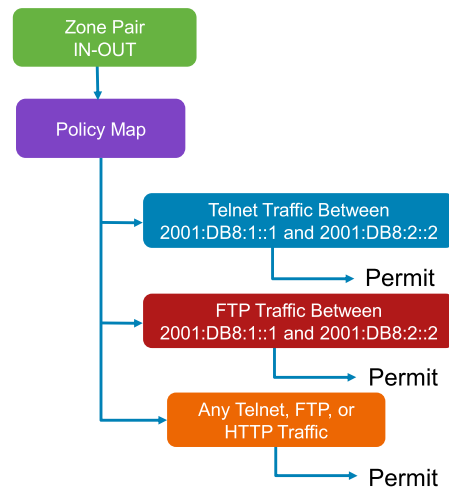
Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Policy Maps

Policy maps determine the firewall policy applied to a class:

- Permit, deny, log, or special actions
- Evaluated in class order

A policy map is applied to each configured zone pair.



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-37

A C3PL policy map is a configuration object that determines the actions (firewall policy) that will be applied to a particular class, for a particular zone pair to which the policy map will be applied. Generally, a policy map can permit access for a traffic class, deny access for a traffic class, and log access for a traffic class between zones.

A policy map is a sequential list of classes and the actions that are applied to them. When a session is establishing between zones, the policy map is consulted in sequence (class order), and the action that is assigned to the first matching class is applied to the session.

Policy maps must be applied to zone pairs. You can apply the same policy map to one or more zone pairs. However, most configurations will use a single, unique policy map for each zone pair.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Policy Actions

Policy Action	Description	Use
Inspect	Permit and statefully track session of this class	Permit protocols that Zone-Based Policy Firewall can statefully inspect
Pass	Statelessly (unidirectionally) permit packet of this class between zones	Permit protocols that Zone-Based Policy Firewall does not statefully inspect
Drop	Deny traffic of this class	Deny specific traffic classes
Log	Log packet of this class (used with drop action)	Log denied traffic

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-38

A Cisco IOS Zone-Based Policy Firewall can take four possible actions for a particular class, as dictated by the policy map:

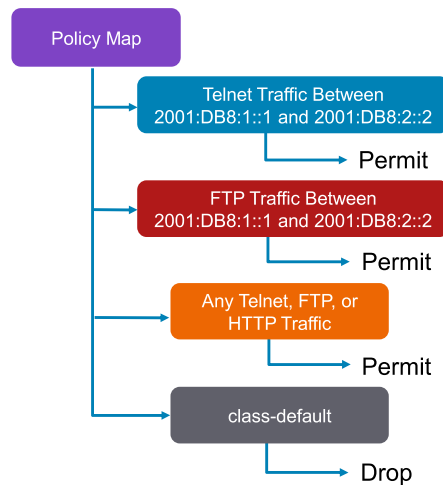
- **Inspect:** This action permits the class and enables stateful tracking for all inspectable sessions of the class. You should use this action for traffic classes that should be permitted through the firewall and that contain protocols that the Zone-Based Policy Firewall can inspect.
- **Pass:** This action permits the class and statelessly (without keeping state) allows the packet to be delivered to the destination zone. The pass action allows the traffic in one direction only. A corresponding policy must be applied to allow return traffic to pass in the opposite direction. You should use this action for traffic classes that should be permitted through the firewall and that contain protocols that the Zone-Based Policy Firewall cannot inspect statefully; for example, asynchronous ICMP messages (such as ICMP unreachable), Generic Routing Encapsulation (GRE) or Open Shortest Path First (OSPF) traffic, and so on.
- **Drop:** This action denies the class between the zone pair.
- **Log:** This action can be used with the drop action only, to log packets that are denied by the Zone-Based Policy Firewall configured policy.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Default Class

There is always an implicit class called **class-default** as the last class in each policy map:

- The default action in class-default is **drop**.
- You can customize actions by explicitly specifying the class-default inside the policy.



© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-39

Zone-Based Policy Firewall policy maps always include a special, implicitly present class, which is called class-default, as the last class in the policy map. The class-default class is a catch-all class that matches any traffic that was not previously matched by other, more specific classes.

The default action of class-default is to drop traffic. However, you can tune the class-default policy by explicitly listing it inside your policy map; for example, to also include the log action.

Note You cannot assign the inspect or pass actions to the class-default class.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Default Access Policy

Source Interface Member of Zone?	Destination Interface Member of Zone?	Zone Pair Exists?	Policy Map Exists?	Result
No	No	—	—	Permit
Yes (zone A)	Yes (zone A)	— *	—	Permit
Yes	No	—	—	Deny
No	Yes	—	—	Deny
Yes (zone A)	Yes (zone B)	No	—	Deny
Yes (zone A)	Yes (zone B)	Yes	No	Deny
Yes (zone A)	Yes (zone B)	Yes	Yes	Apply policy map actions

* Can be optionally inspected from Cisco IOS Software Release 15.0(1)M

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-40

The zone membership of the router network interfaces is subject to several rules that govern interface behavior, as with the traffic that moves between the router interfaces:

- You must configure a zone before you can assign interfaces to it.
- You can assign an interface to only one security zone.
- Traffic is implicitly allowed to flow by default among interfaces that are members of the same zone. However, you can change this behavior, starting with Cisco IOS Software Release 15.0(1)M.
- To permit traffic to and from a zone member interface, you must configure a policy that allows or inspects traffic between that zone and any other zone.
- Traffic cannot flow between a zone member interface and any interface that is not a zone member. You can apply pass, inspect, and drop actions only between two zones.
- Interfaces that have not been assigned to a zone function as classical router ports and might still use classical stateful inspection (CBAC) configuration.
- If you do not want an interface on the router to be part of the zone-based firewall policy, you might still need to put that interface in a zone and configure a pass all policy (a dummy policy) between that zone and any other zone to which traffic flow is desired.
- From the preceding rules it follows that, if traffic is to flow among all of the interfaces in a router, all of the interfaces must be part of the zoning model (each interface must be a member of a zone).

The figure shows several examples of different interface and configuration combinations.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Configuration Tasks

1. Configure classes.
2. Configure an access policy.
3. Apply a policy to a zone pair.

© 2010 Cisco Systems, Inc. All rights reserved.

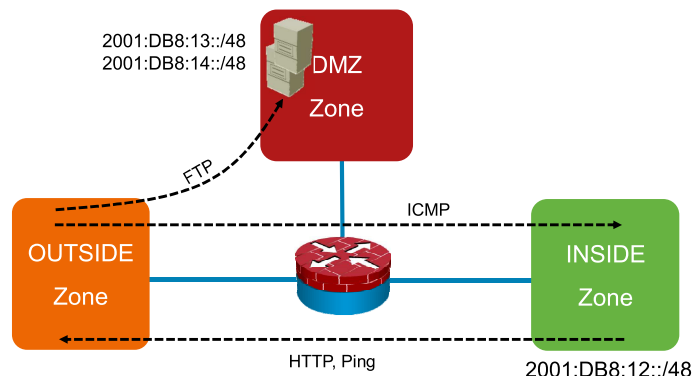
IPv6FD v3.0—7-41

To configure the Cisco IOS Software Zone-Based Policy Firewall with a basic (OSI Layer 3 to 4) interzone access control policy, you need to perform the following configuration tasks:

- Step 1** Configure traffic classes by using C3PL class maps.
- Step 2** Configure access policies by using C3PL policy maps for all configured zone pairs.
- Step 3** Apply the configured policy maps to zone pairs.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Configuration Scenario



© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-42

This figure presents the configuration scenario that is used in upcoming configuration tasks. You will create the following basic interzone access policy:

- Hosts in the INSIDE zone, with addresses in the 10.0.0.0/8 network, can access any host in the OUTSIDE zone, by using the HTTP and ping (ICMP echo) protocols.
- Any hosts in the OUTSIDE zone can access two FTP servers (172.31.1.10 and 172.31.1.11) in the DMZ zone, by using the FTP protocol.
- Any hosts in the OUTSIDE zone can send any ICMP destination unreachable messages to hosts in the INSIDE zone, with addresses in the 10.0.0.0/8 network (this is typically required for TCP/IP stacks to work optimally, including enabling their PMTUD functions).
- All other traffic between zone pairs is prohibited and should be dropped.

The configuration steps in the upcoming configuration sequence assume that the Cisco ISR is already configured with all required Zone-Based Policy Firewall zones (INSIDE, OUTSIDE, DMZ), and that appropriate network interfaces are assigned to these zones, as specified in the previous topic.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Task 1: Configure Classes.

```
ipv6 access-list extended IN-TO-OUT-ACL
 permit tcp 2001:db8:12::/48 any eq http
 permit icmp 2001:db8:12::/48 any echo
!
ipv6 access-list extended OUT-TO-IN-ACL
 permit icmp any 2001:db8:12::/48
!
class-map type inspect IN-TO-OUT-CLASS
 match access-group name IN-TO-OUT-ACL
!
class-map type inspect OUT-TO-IN-CLASS
 match access-group name OUT-TO-IN-ACL
```

Create ACLs that describe the first packets of sessions for which Zone-Based Policy Firewall will rely on PAM to identify the required inspection routines.

Create a class map that defines a class based on the ACL.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-43

In the first task of the configuration sequence, configure class maps that describe traffic that must be permitted across the Zone-Based Policy Firewall-enabled router.

To describe HTTP and ping traffic that will be permitted between the INSIDE and OUTSIDE zone, IN-TO-OUT-ACL describes the initial packets of HTTP and ping sessions from the 10.0.0.0/8 network. IN-TO-OUT-CLASS matches the configured ACL to classify these two applications into the same class. You could alternatively configure two distinct classes for HTTP and ping.

To describe ICMP destination unreachable packets that will be permitted between the OUTSIDE and INSIDE zones, OUT-TO-IN-ACL describes the required ICMP packet type. OUT-TO-IN-CLASS matches the configured ACL to classify these packets into a separate class.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Task 2: Configure an Access Policy.

```
policy-map type inspect IN-TO-OUT-POLICY
  class type inspect IN-TO-OUT-CLASS
    inspect
  class class-default
    drop log
!
policy-map type inspect OUT-TO-DMZ-POLICY
  class type inspect OUT-TO-DMZ-CLASS
    inspect
  class class-default
    drop log
!
policy-map type inspect OUT-TO-IN-POLICY
  class type inspect OUT-TO-IN-CLASS
    pass
  class class-default
    drop log
```

For each zone pair, create a named policy map.

For classes that contain statefully inspectable protocols, use the inspect action to permit and track sessions.

You can optionally include the class-default class, which you can use to log packets that are denied by default.

Use the pass action to statelessly permit protocols that are not statefully tracked by Zone-Based Policy Firewall.

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-44

In the second configuration task, you configure policy maps to determine the access policy for each required and configured zone pair.

First, configure a policy map to control access between the INSIDE and OUTSIDE zones. Use the **policy-map type inspect** command to create a named policy map (IN-TO-OUT-POLICY). In the policy map configuration mode, use the **class type-inspect** command to refer to a previously configured IN-TO-OUT-CLASS. In the class submode, specify the **inspect** action to permit and statefully inspect the class protocols (HTTP and ping). Exit the class submode, and create an optional reference to the class-default class, by using the **class class-default** command. In the class-default class, tune the default **drop** action to also **log** denied traffic, by specifying both actions (**drop log**) in the class policy.

Next, configure a policy map to control access between the OUTSIDE and DMZ zones (OUT-TO-DMZ-POLICY). In the policy map configuration mode, use the **class type-inspect** command to refer to a previously configured OUT-TO-DMZ-CLASS. In the class submode, specify the **inspect** action to permit and statefully inspect the class protocol (FTP). Exit the class submode, and assign the **drop log** actions to the class-default class of this policy map.

Finally, configure a policy map to control access between the OUTSIDE and INSIDE zones (OUT-TO-IN-POLICY). In the policy map configuration mode, use the **class type-inspect** command to refer to a previously configured OUT-TO-IN-CLASS. In the class submode, specify the **pass** action to statelessly permit the class protocol (ICMP destination unreachable messages), which cannot be treated statefully. Exit the class submode, and assign the **drop log** actions to the class-default class of this policy map.

policy-map type inspect

To create a Layer 3 and Layer 4 or a Layer 7 (protocol-specific) inspect type policy map, use the **policy-map type inspect** command in global configuration mode. To delete an inspect type policy map, use the **no** form of this command.

Layer 3 and Layer 4 (Top Level) Policy Map Syntax

policy-map type inspect *policy-map-name*

Layer 7 (Application-Specific) Policy Map Syntax

policy-map type inspect *protocol-name policy-map-name*

policy-map type inspect Parameters

Parameter	Description
<i>policy-map-name</i>	Name of the policy map. The name can be a maximum of 40 alphanumeric characters.
<i>protocol-name</i>	Layer 7 application-specific policy map. These are the supported protocols: ■ h323: H.323 version 4 (H.323v4) ■ http: HTTP ■ im: Instant messaging (IM) protocol; the supported IM protocols include: — America Online (AOL) Version 5 and later — I Seek You (ICQ) Version 2003b.5.56.1.3916.85 — MSN Messenger Version 6.x and 7.x — Windows Messenger Version 5.1.0701 — Yahoo Messenger Version 9.0 and later ■ imap: Internet Message Access Protocol (IMAP) ■ p2p: PPP ■ pop3: Post Office Protocol version 3 (POP3) ■ sip: Session Initiation Protocol (SIP) ■ smtp: Simple Mail Transfer Protocol (SMTP) ■ sunrpc: Sun Remote Procedure Call (SunRPC)

inspect

To enable Cisco IOS stateful packet inspection, use the **inspect** command in policy map class configuration mode. To disable stateful packet inspection, use the **no** form of this command.

inspect [*parameter-map-name*]

inspect Parameters

Parameter	Description
<i>parameter-map-name</i>	(Optional) Name of a previously configured parameter map. If you do not specify a parameter map name, the software uses the default values for all the parameters.

drop (zone-based policy)

To drop packets that are sent to the router, use the **drop** command in policy map class configuration mode.

drop [**log**]

drop (zone-based policy) Parameters

Parameter	Description
log	(Optional) Displays logging messages about dropped packets.

log (policy-map and class-map)

To generate a log of messages, use the **log** command in policy map configuration mode or class map configuration mode. To disable the log, use the **no** form of this command.

log

pass

To allow packets to be sent to the router without being inspected, use the **pass** command in policy map class configuration mode.

pass

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Task 3: Apply Policy Maps to Zone Pairs.

```
zone-pair security IN-TO-OUT source INSIDE destination OUTSIDE
service-policy type inspect IN-TO-OUT-POLICY
!
zone-pair security OUT-TO-DMZ source OUTSIDE destination DMZ
service-policy type inspect OUT-TO-DMZ-POLICY
!
zone-pair security OUT-TO-IN source OUTSIDE destination INSIDE
service-policy type inspect OUT-TO-IN-POLICY
```

Apply a policy map to each zone pair.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6 v3.0—7-45

In task three of this configuration sequence, you apply the configured policy maps to their respective zone pairs.

Use the **zone-pair security** global configuration command to enter the zone pair configuration mode. In this mode, use the **service-policy type inspect** command to associate a named policy map with each configured zone pair. In this example, associate the IN-TO-OUT-POLICY policy that you configured in the previous task with the IN-TO-OUT zone pair, the OUT-TO-DMZ-POLICY policy that you configured in the previous task with the OUT-TO-DMZ zone pair, and the OUT-TO-IN-POLICY policy that you configured in the previous task with the OUT-TO-IN zone pair.

service-policy type inspect

To attach a firewall policy map to a zone-pair, use the **service-policy type inspect** command in zone pair configuration mode. To disable this attachment to a zone pair, use the **no** form of this command.

service-policy type inspect *policy-map-name*

service-policy type inspect Parameters

Parameter	Description
<i>policy-map-name</i>	Name of the policy map. The name can be a maximum of 40 alphanumeric characters.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Verify Class Maps

```
Router# show class-map type inspect
Class Map type inspect match-all OUT-TO-DMZ-CLASS (id 3)
  Match access-group name OUT-TO-DMZ-HOSTS
  Match protocol ftp

Class Map type inspect match-all IN-TO-OUT-CLASS (id 1)
  Match access-group name IN-TO-OUT-ACL

Class Map type inspect match-all OUT-TO-IN-CLASS (id 2)
  Match access-group name OUT-TO-IN-ACL
```

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-46

To verify your configuration, you can use Cisco IOS Software commands to verify the contents of your class maps, policy maps, and service policies.

To display a specific class map, use the **show class-map type inspect** *class-map-name* command in privileged EXEC mode. If a class map name is not specified, all class maps are displayed. The example in the figure displays all of the inspection class maps that were created in the preceding configuration sequence.

show class-map type inspect

To display Layer 3 and Layer 4 or Layer 7 (application-specific) inspect type class maps and their matching criteria, use the **show class-map type inspect** command in privileged EXEC mode.

show class-map type inspect [*protocol-name*] [*class-map-name*]

show class-map type inspect Parameters

Parameter	Description
<i>protocol-name</i>	(Optional) Layer 7 application-specific class map. These are the supported protocols: ■ aol: AOL IM ■ edonkey: eDonkey peer-to-peer ■ fasttrack: FastTrack traffic peer-to-peer ■ gnutella: Gnutella version 2 traffic peer-to-peer ■ h323: H.323 protocol ■ http: HTTP ■ icq: ICQ IM ■ imap: IMAP ■ kazaa2: Kazaa version 2 peer-to-peer ■ msnmsgr: MSN Messenger IM protocol ■ pop3: POP3 ■ sip: SIP ■ smtp: SMTP ■ sunrpc: SunRPC ■ winmsgr: Windows IM ■ ymsgr: Yahoo IM
<i>class-map-name</i>	(Optional) Name of the inspect type class map. The name can be a maximum of 40 alphanumeric characters.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Verify Policy Maps

```
Router# show policy-map type inspect
Policy Map type inspect OUT-TO-DMZ-POLICY
  Class OUT-TO-DMZ-CLASS
    Inspect
  Class class-default
    Drop log

Policy Map type inspect IN-TO-OUT-POLICY
  Class IN-TO-OUT-CLASS
    Inspect
  Class class-default
    Drop log

Policy Map type inspect OUT-TO-IN-POLICY
  Class OUT-TO-IN-CLASS
    Pass
  Class class-default
    Drop log
```

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-47

To display a specific policy map, use the **show policy-map type inspect *policy-map-name*** command in privileged EXEC mode. If a policy map name is not specified, all policy maps are displayed. The example in the figure displays all of the inspection policy maps that were created in the preceding configuration sequence.

show policy-map type inspect

To display a specified policy map, use the **show policy-map type inspect** command in privileged EXEC mode.

show policy-map type inspect [*policy-map-name*] [**class** *class-map-name*]

show policy-map type inspect Parameters

Parameter	Description
<i>policy-map-name</i>	(Optional) Name of the policy map.
<i>class class-map-name</i>	(Optional) Name of the class map.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Verify Policy Assignment

```
Router# show zone-pair security
Zone-pair name IN-TO-OUT-PAIR
  Source-Zone INSIDE Destination-Zone OUTSIDE
  service-policy IN-TO-OUT-POLICY
Zone-pair name OUT-TO-DMZ
  Source-Zone OUTSIDE Destination-Zone DMZ
  service-policy OUT-TO-DMZ-POLICY
Zone-pair name OUT-TO-IN
  Source-Zone OUTSIDE Destination-Zone INSIDE
  service-policy OUT-TO-IN-POLICY
```

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-48

To display the source zone, destination zone, and policy that are attached to the zone pair, enter the **show zone-pair security** [*source source-zone-name*] [*destination destination-zone-name*] command. If you do not specify a source and a destination, all of the zone pairs and their associated policies are displayed. If you specify only a source or a destination zone, all of the zone pairs that contain that zone as the source or destination (depending on which one you specify) are displayed.

show zone-pair security

To display the source zone, destination zone, and policy that are attached to the zone pair, use the **show zone-pair security** command in privileged EXEC mode. To disable the display, use the **no** form of this command.

show zone-pair security [*source source-zone-name*] [*destination destination-zone-name*]

show zone-pair security Parameters

Parameter	Description
source <i>source-zone-name</i>	(Optional) Name of the source zone.
destination <i>destination-zone-name</i>	(Optional) Name of the destination zone.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Verify Classification and Policy Enforcement

```
Router# show policy-map type inspect zone-pair
policy exists on zp IN-TO-OUT-PAIR
Zone-pair: IN-TO-OUT-PAIR
Service-policy inspect : IN-TO-OUT-POLICY
Class-map: IN-TO-OUT-CLASS (match-all)
  Match: access-group name IN-TO-OUT-ACL
  Match: protocol telnet
Inspect
  Session creations since subsystem startup or last reset 845647433
  Current session counts (estab/half-open/terminating) [72:2:9]
  Maxever session counts (estab/half-open/terminating) [4386:15:563]
< some output omitted>

Class-map: class-default (match-any)
  Match: any
  Drop
    46579346 packets, 9568934789 bytes
```

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-49

To display statistics of an applied policy and to verify that it is matching traffic and enforcing a policy, you can use the **show policy-map type inspect zone-pair** command. This command displays a policy map along with its class maps and displays session statistics, including the number of dropped packets in the class-default class.

show policy-map type inspect zone-pair

To display the runtime inspect-type policy-map statistics and other information, such as sessions that exist on a specified zone pair, use the **show policy-map type inspect zone-pair** command in privileged EXEC mode.

show policy-map type inspect zone-pair [*zone-pair-name*] [*sessions*]

show policy-map type inspect zone-pair Parameters

Parameter	Description
<i>zone-pair-name</i>	(Optional) Zone pair for which the system will display the runtime inspect-type policy-map statistics. Default: The requested information is shown for all zone pairs.
sessions	(Optional) Displays the Cisco IOS Software stateful packet inspection sessions that are created because of the policy-map application on the specified zone pair.

Configure a Basic OSI Layer 3 to 4 Interzone Access Policy (Cont.)

Verify Sessions

```
Router# show policy-map type inspect zone-pair sessions

policy exists on zp IN-TO-OUT-PAIR
Zone-pair: IN-TO-OUT-PAIR
Service-policy inspect : IN-TO-OUT-POLICY
Class-map: IN-TO-OUT-CLASS (match-all)
Match: access-group name IN-TO-OUT-ACL
Match: protocol telnet
Inspect
Number of Half-open Sessions = 0
Sessions
  Session 673E1FE0 [2001:db8:12::1]:52227=>[2001:db8:F100::47]:80 tcp SIS OPEN
    Created 00:00:12, Last heard 00:00:12
    Bytes sent (initiator:responder) [534:6786]
  < rest of sessions omitted >
```

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-50

To display all individual sessions (which can be limited to a specific zone pair), you can extend the **show policy-map type inspect zone-pair sessions** command with the **sessions** argument. In the output of this command, the router displays all statefully tracked sessions between a particular zone pair, allowing you to verify that a particular session is statefully managed by the router.

IPv6 Routing Header

```
parameter-map type inspect ipv6-header
ipv6 routing-header-enforcement loose
```

- Zone-Based Policy Firewall will by default drop all packets with routing header.
- Loose header enforcement in a parameter map may be used to change default behavior.

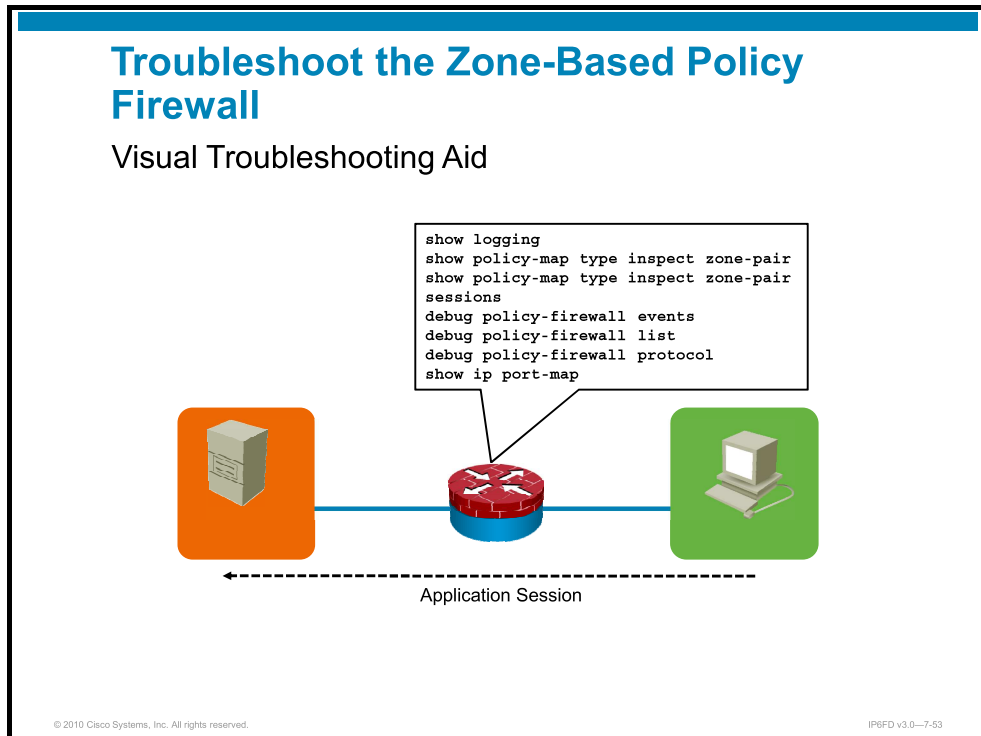
© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0—7-51

The **ipv6 routing-header-enforcement loose** command in a parameter map can be used in the Zone-Based Policy Firewall the same way that the **ipv6 inspect routing-header** command can be used with CBAC. The **ipv6 routing-header-enforcement loose** command allows packets with a routing header through the firewall. Default behavior is to drop such packets.

Troubleshooting the Zone-Based Policy Firewall

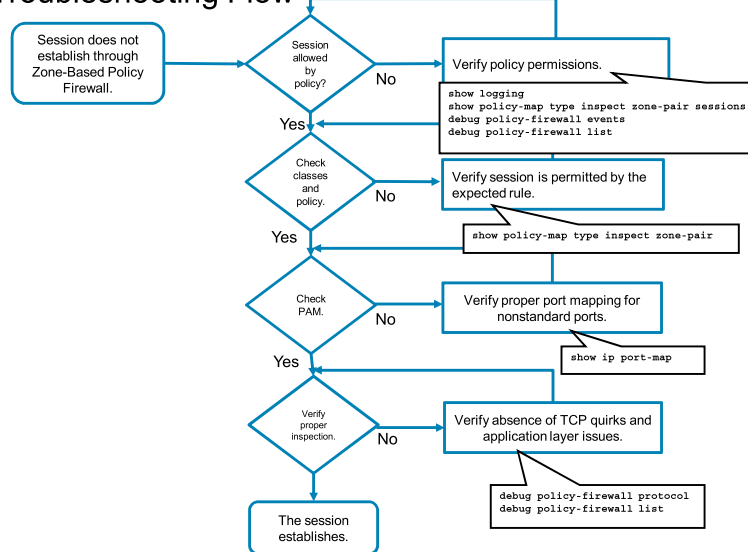
This topic enables you to troubleshoot the operation of basic Zone-Based Policy Firewall functions.



To troubleshoot issues that are related to basic access control, you can use a set of Cisco IOS Software CLI verification and debugging features. This figure lists some common commands and features that you might find useful during the troubleshooting session. Typically, you will examine why a particular session cannot establish through a Zone-Based Policy Firewall-enabled router.

Troubleshoot the Zone-Based Policy Firewall (Cont.)

Troubleshooting Flow



The figure shows the recommended task flow for troubleshooting basic access control through a Zone-Based Policy Firewall-enabled router. Follow these steps, and use the provided commands to troubleshoot events when a host appears to have connectivity issues because of an implemented security policy:

- Step 1** First, verify that the problematic connection is allowed by an interzone or intrazone policy. If you have configured logging in the class-default class, use the **show logging** command to observe possible events that indicate a denied connection. Use the **show policy-map type inspect zone-pair sessions** command to see whether your session enters the state table of the router. Additionally, consider using the **debug policy-firewall events** command to see whether the Zone-Based Policy Firewall manages the session. This debug output will indicate creation of state table objects and denied packets, even if logging has not been configured in the class-default class.
- Step 2** If the session appears to be permitted, verify that it is permitted by the correct class and policy-map combination. Use the **show policy-map type inspect zone-pair** to observe matching statistics (if possible; other sessions might also increase counters), or manually examine the configured class maps and policy maps.
- Step 3** If the problematic session is using a nonstandard port, verify that this port is defined by the PAM module. Use the **show ip port-map** command if the session requires special inspection; for example, if it is a dynamic application, such as FTP or Oracle SQL*Net or TNS.
- Step 4** Finally, verify that the Zone-Based Policy Firewall properly inspects the session, by examining the Zone-Based Policy Firewall inspection routines in depth. Use the **debug policy-firewall protocol** command, and debug either the application or transport protocol to observe possible issues with either protocol or router behavior.

If all these steps do not resolve your issue, you might need to deploy troubleshooting tools that are beyond the scope of this course.

Troubleshoot the Zone-Based Policy Firewall (Cont.)

Troubleshooting Policies

- Classes that log dropped packets (for example, class-default) will indicate packet drops.

```
%FW-6-LOG SUMMARY: 1 packet were dropped from [2001:db8:1::1]:0 => [2001:db8:2::1]:3  
(target:class)-(IN-TO-OUT:class-default)
```

- Debugging Zone-Based Policy Firewall events will show all permitted or denied sessions.

```
Router# debug policy-firewall events  
FIREWALL*: FW CCE dropping pak 0x66B1FEEC in fast path  
FIREWALL sis 672ED2E0: [2001:db8:1::1]:52232=> [2001:db8:2::1]:22, tcp flag:0x4,  
pak:0x66EALDF8, fso:0x672ED2E0, tcp seq:4001713440, tcp ack:0,  
tcp window:4128, FastEthernet0/1
```

- It is strongly recommended to use debug conditions, based on traffic-selecting ACLs, to minimize debug output.

```
Router# debug policy-firewall list DEBUG
```

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0--7.55

The first printout indicates the message that a Zone-Based Policy Firewall-enabled router will generate if the class-default class is configured with a log action, and a packet or session is denied.

The second printout shows an example output of the **debug policy-firewall events** command, showing a dropped and permitted session.

Limiting debugging to specific sessions is strongly recommended; otherwise, the amount of debug output is likely to overwhelm you in presence of other traffic. Use the **debug policy-firewall list** command to specify an extended ACL that will limit the flows that are subject to debugging (that is, the flows that are permitted by the specified ACL).

debug policy-firewall

To display messages about Cisco IOS Firewall events, including details about the packets of the protocol, use the **debug policy-firewall** command in EXEC mode. To disable debugging output, use the **no** form of this command.

debug policy-firewall {function-trace | object-creation | object-deletion | list {access-list | extended-access-list} | events | timers | protocol protocol-name | packet-path | L2-transparent | control-plane | detailed}

debug policy-firewall Parameters

Parameter	Description
function-trace	Displays messages about software functions that the Cisco IOS Firewall calls.
object-creation	Displays messages about software objects that the Cisco IOS Firewall creates. Object creation corresponds to the beginning of Cisco IOS Firewall inspected sessions.
object-deletion	Displays messages about software objects that the Cisco IOS Firewall deletes. Object deletion corresponds to the closing of Cisco IOS Firewall inspected sessions.

Parameter	Description
<code>list</code>	Displays messages about policy firewall conditional debugging.
<code>access-list</code>	Filters the basic list of policy firewall conditional debugging messages. The valid range is from 1 to 199.
<code>extended-access-list</code>	Filters the extended range of policy firewall conditional debugging messages. The valid range is from 1300 to 2699.
<code>events</code>	Displays messages about Cisco IOS Firewall software events, including information about Cisco IOS Firewall packet processing or MIB special events.
<code>timers</code>	Displays messages about Cisco IOS Firewall timer events, such as when the Cisco IOS Firewall idle timeout is reached.
<code>protocol protocol-name</code>	Displays messages about Cisco IOS Firewall inspected protocol events, including details about the packets of the protocol. These are the supported protocols: ■ aol: AOL IM ■ cuseeme: CU-SeeMe ■ dns-resolver: DNS resolver ■ dns-timer: DNS timer ■ ftp-cmd: FTP commands and responses ■ ftp-token: FTP token (enables tracing of the parsed FTP tokens) ■ h225ras: H.225 Registration, Admission, and Status (RAS) configuration ■ h323: H.323v4 ■ http: HTTP ■ icmp: ICMP ■ icq: ICQ IM ■ imap: IMAP ■ mgcp: Media Gateway Control Protocol (MGCP) ■ msn-msgr: MSN Messenger IM protocol ■ msrpc: Microsoft Remote Procedure Call (Microsoft RPC) ■ netshow: Microsoft NetShow ■ p2p: PPP ■ pop3: POP3 ■ rcmd: UNIX R commands (rlogin, rexec, rsh) ■ realaudio: RealAudio ■ rpc: RPC ■ rtsp: Real Time Streaming Protocol (RTSP) ■ sip: SIP ■ skinny: Skinny Client Control Protocol (SCCP) ■ smtp: SMTP ■ sqlnet: SQL*Net ■ streamworks: StreamWorks ■ stun-ice: STUN-ICE ■ tcp: TCP ■ tftp: Trivial File Transfer Protocol (TFTP) ■ udp: UDP ■ vdolive: VDOLive

Parameter	Description
	■ winmsgr: Windows IM ■ yahoo: Yahoo IM
packet-path	Displays messages about the packet-path functions.
L2-transparent	Displays messages about Layer 2 transparent (firewall) bridge mode events.
control-plane	Displays messages about the control plane routines.
detailed	Displays detailed information for all the other enabled Cisco IOS Firewall debug commands. Use this form of the command with the other Cisco IOS Firewall debug commands.

Troubleshoot the Zone-Based Policy Firewall (Cont.)

Troubleshooting Policies (Cont.)

- You can troubleshoot Zone-Based Policy Firewall inspection routines for dynamic protocols (FTP, H.323, and so on) to verify compatibility.

```
Router# debug policy-firewall protocol ftp-cmd
```

- You can troubleshoot Zone-Based Policy Firewall TCP inspection routines to verify TCP stateful engine and normalizer operation.

```
Router# debug policy-firewall protocol tcp
FIREWALL* sis 672F36E0: pak 66B1FEBC SIS CLOSED/LISTEN TCP SYN
SEQ 2916869111 LEN 0 [2001:db8:3::1]:52234 => [2001:db8:2::7]:23
FIREWALL* sis 672F36E0: Found TCP window scale option in SYN
FIREWALL* sis 672F36E0: Scale window size by 2 bits for sis
FIREWALL* sis 672F36E0: pak 66B1FEBC SIS OPENING/SYNSENT TCP SYN ACK
2916869112 SEQ 3500288275 LEN 0 [2001:db8:3::1]:52234 <= [2001:db8:2::7]:23
FIREWALL sis 672F36E0: pak 664C2E84 SIS OPENING/SYNRCVD TCP SYN ACK
2916869112 SEQ 711524500 LEN 0 [2001:db8:3::1]:52234 <= [2001:db8:2::7]:23
FIREWALL sis 672F36E0: L4 result: OoO may process the packet 0x664C2E84
[2001:db8:2::7]:23 [2001:db8:3::1]:52234 bytes 24 ErrStr = Out-Of-Order Segment
```

© 2010 Cisco Systems, Inc. All rights reserved.

IP6FD v3.0—7-56

The **debug policy-firewall protocol** command allows for very extensive debugging of low-level inspection routines and should be used as a last-resort option. The two printouts show the resulting output of application layer protocol (FTP) debugging and of transport layer protocol (TCP) debugging.

Summary

This topic summarizes the key points that were discussed in this lesson.

Summary

- Cisco IOS Firewall uses stateful packet inspection to inspect traffic and create temporary openings at firewall interfaces.
- Inspection rules specify which traffic will be inspected.
- Zone-Based Policy Firewall groups interfaces into zones, with rulesets applied to interzone traffic.
- Stateful firewalls break end-to-end connectivity.

Module Summary

This topic summarizes the key points that were discussed in this module.

Module Summary

- Implementing IPv6 security features in Cisco IOS Software involves creating ACLs.
- IPsec is the baseline protocol established for enabling packet security during transmission.
- If a client that is implementing IPv6 is unaware of the security risks posed by certain integration methods, its network could be exposed to security breaches. Therefore, it is necessary to recommend the most appropriate integration mechanism to such a client.
- You must consider several high-level security practices for IPv6 networks, for your deployment strategy and network design.
- One of the main features of the Cisco IOS Firewall Feature Set is CBAC.

© 2010 Cisco Systems, Inc. All rights reserved.

IPv6FD v3.0--7-1

Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

- Q1) Which three fields can Cisco IOS Software extended IPv6 ACLs match? (Choose three.) (Source: “Configuring IPv6 ACLs”)
- A) destination IPv6 address
 - B) TCP source port
 - C) DSCP value
 - D) IPv6 hop limit
- Q2) What is the key capability of a reflexive ACL over a standard ACL? (Source: “Configuring IPv6 ACLs”)
- A) All return traffic for an established session is denied by the reflexive ACL.
 - B) All return traffic for an established session is allowed by default, without needing an inbound ACL.
 - C) Outbound traffic that originates within the enterprise network is rate-limited across the egress interface.
 - D) Return traffic that is associated with an established outside session is allowed back into the enterprise, if the outbound reflexive ACL and the incoming reflexive ACL are both configured.
- Q3) How are the header filtering keywords “undetermined transport” used? (Source: “Configuring IPv6 ACLs”)
- A) to match IPv6 packets when there is an unrecognized Next Header value
 - B) to match malformed IPv6 packets when the packet appears to be incomplete
 - C) to match fragmented IPv6 packets when the initial fragment does not contain Layer 4 information (such as if the packet is a TCP packet with destination port 80)
 - D) with PMTUD, to advise the source node when an unrecognized Layer 4 protocol makes it impossible for IPv6 to determine the length of the Layer 3 payload
- Q4) What is the function of IKE in an IPv6 network? (Source: “Using IPsec, IKE, and VPNs”)
- A) to encrypt IPsec data sessions between two peer nodes
 - B) to negotiate IPsec SAs for peer nodes through a protected IKE tunnel
 - C) to compete with AES as an encryption type in high-security deployments
 - D) to negotiate ISAKMP relationships that are then used as secret shared keys when IPsec AH is used between nodes

- Q5) What does the **rtr(config)#crypto isakmp key GreenCableRE address ipv6 2001:DB8:A:6E::1/128** command do? (Source: “Using IPsec, IKE, and VPNs”)
- A) establishes GreenCableRE as the ISAKMP bidirectional preshared secret for the peer at the given address
 - B) establishes GreenCableRE as the IPsec SA for the given peer
 - C) specifies GreenCableRE as the preshared secret for the ISAKMP hash algorithm
 - D) specifies GreenCableRE as the IPsec preshared secret for all sessions to and from the given peer address
- Q6) Allowing inbound ICMPv6 traffic through the enterprise edge firewall is necessary to support which two IPv6 processes or situations? (Choose two.) (Source: “Discussing Security Issues in an IPv6 Transition Environment”)
- A) an Internet-based router that reports that a destination address is unreachable
 - B) an Internet-based router that reports that a packet TTL expired in transit
 - C) an Internet-based router that reports that a routing change has resulted in the acceptance of a larger MTU size
 - D) Internet-based nodes that send link-local ICMPv6 traffic back to source nodes that have in-flight sessions
- Q7) Why is it an insecure practice to use the same host ID for IPv6 as for IPv4 (such as 192.0.2.230, 2001:DB8:4:1::230) in a dual-stack environment? (Source: “Discussing Security Issues in an IPv6 Transition Environment”)
- A) The given IPv6 address is not legitimate: It overlaps with interface identifiers that are reserved for ISATAP.
 - B) EUI-64 interface identifiers for static addresses must have bit 7 of the identifier set to 1 to indicate local significance, making it impossible to configure addresses as shown.
 - C) A scanning attack on IPv4 that locates the node on the subnet makes it easy to determine the IPv6 address of the node as well.
 - D) Not all hosts are dual-stacked, so it is misleading for network administrators to use this practice.
- Q8) What are two reasons that IPv6-in-IPv4 tunnels can cause problems for network security administrators that run the enterprise edge firewall? (Choose two.) (Source: “Discussing Security Issues in an IPv6 Transition Environment”)
- A) It is impossible to inspect an IPv6 packet within an IPv4 tunnel because protocol 41 tunneling obscures the UDP header.
 - B) It is difficult to inspect the IPv6 packet within an IPv4 tunnel because few available firewalls have this capability, although the capability should not be difficult to implement.
 - C) Few firewalls terminate IPv6-in-IPv4 tunnels, making their use in the enterprise impossible.
 - D) There are no available IPv6 firewalls, so IPv6 packets cannot be inspected after being removed from the IPv4 tunnel wrapper.

- Q9) Why are distributed firewalls—platform firewalls—likely to be a primary enabling technology for IPv6? (Source: “Understanding IPv6 Security Practices”)
- A) Firewalls cannot perform a complete inspection of IPv6 packets because of the extension header chaining mechanism.
 - B) IPv6 packets are more streamlined, making it much more efficient to evaluate packet payloads at each source than at the network level.
 - C) The IPv6 large address space and other features encourage peer-to-peer computing, and making IPsec mandatory should increase the volume of encrypted traffic, with network-based middle boxes that gateway firewalls cannot inspect.
 - D) The IPv6 large address space and broad support for IPsec make gateway firewalls more important than ever, but centrally managed gateway firewalls are needed to allow deep inspection of IPv6-in-IPv4 IPsec traffic.
- Q10) IPv6 proxy services have several impacts on network security. What is one advantage of using proxy services for a specific service, such as HTTP? (Source: “Understanding IPv6 Security Practices”)
- A) End-to-end computing is maintained for Layer 3 (IPv6) and Layer 4 (TCP/UDP) sessions.
 - B) All outbound traffic appears to come from a single IPv6 address, making the enterprise invisible on the public network.
 - C) All outbound traffic appears to come from the proxy server address, so the internal subnet topology of the network is not seen by Internet-based nodes.
 - D) New inbound sessions are all sent to the proxy server address, regardless of the true address of the internal node or the port that the application is running on, and are redirected from that point, allowing a single DNS entry for the proxy server to support all services.
- Q11) What does the Cisco IOS Firewall do? (Source: “Using Cisco IOS Firewall for IPv6”)
- A) protects against attacks that originate from within the protected network
 - B) provides network protection on multiple levels
 - C) provides intelligent filtering for all protocols
 - D) protects against every type of attack

Module Self-Check Answer Key

Q1)	A, B, C
Q2)	D
Q3)	C
Q4)	B
Q5)	A
Q6)	C, D
Q7)	B
Q8)	A, B
Q9)	C
Q10)	C
Q11)	B