25. IPv6 ACLs

ccnacookbook.com

The general rule is that you can have one Access Control List (ACL) per interface per direction per protocol. That means that you can have both IPv4 and IPv6 ACLs on the same interface when you're double-stacking.

Security Filtering Practices

- An inbound ACL can prevent a "bad" packet from even entering your edge router
- Filter outbound packets at the last possible moment, outbound on the one interface that leaves your network. That will prevent later reconfigurations from introducing pathways that bypass your filtering
- It's more secure to explicitly permit that which is allowed and allow the implicit deny to auto-block everything else
- Some IPv6 ICMP packets are important to IPv6 working properly, like Neighbor Discovery Protocol (NDP) and MTU discovery, so don't recklessly filter all ICMP, like some people do in IPv4. [See also: RFC 4890, link at end.]

Matchable Items in IPv6 Header Include

- Source & Destination Addresses
- TCP/UDP Port numbers & TCP flags (SYN, ACK, etc.)
- Traffic Class (DSCP values range 0-63)
- Flow Label (values range 0-1 048 575)
- Next Header Field (type & number of the extension header)
- IPv6 Extension Header (e.g. IPSec) Type & Value
- ICMPv6 Type & Code

Limitations

- Can't filter an IPv6 Packet that is encapsulated, e.g. within IPv4 for tunneling
- IPv6 ACLs use a prefix length instead of a wildcard, so you can't have discontiguous masks, e.g. "all even numbered IP addresses"
- The "log" keyword triggers on the first match and logs it immediately. Subsequent matches are written to the log every 5 minutes to save CPU
- Router-generated packets are exempt (as in IPv4)

CONFIGURATION

All IPv6 ACLs are named (no numbered ACLs) and the IPv4 distinction between a "standard" and "extended" ACL is dropped. Some options include:

```
R1(config)# ipv6 access-list myACL
R1(config-ipv6-acl)# remark Same as IPv4
R1(config-ipv6-acl)# {permit | deny} {ipv6 | tcp | udp | icmp} <src phrase>
                  <dest phrase> [log]
```

Where both the source and destination phrases look like:

```
{any | host 2001:db8:ff::1 | 2001:db8:ff::/64} [eq 80]
```

The eq phrase can be used with TCP or UDP to denote a port number. You can also use "neq," etc.

Predictably, ACLs are assigned to interfaces much like in IPv4. Only the keyword "traffic-filter" is really new, replacing IPv4's "access-group." You expected the "ipv6" keyword.

```
R1(config)# interf gi0/0
R1(config-if)# ipv6 traffic-filter MyACL {in | out}
```

VERIFICATION

```
R1# show ipv6 access-list
```

The command now uses singular "access-list," unlike ipv4's plural

```
IPv6 access list myACL
    permit tcp any host 2001:DB8:FF::1 eq www sequence 20
    deny tcp any any eq www sequence 30
    permit tcp any any eq 443 sequence 40
```

Sequence numbers are now at the end, begin at 20 instead of 10, and are preceded by the keyword "sequence." This is also how you would type a sequence number if you were modifying an ACL.

```
R1# show ipv6 interf gi0/2
```

IPv6 will need to be enabled on the interface (IPv6 address or "ipv6 enable") before this command will show any output.

```
GigabitEthernet0/2 is administratively down, line protocol is down
  IPv6 is tentative, link-local address is FE80::A693:4CFF:FEF4:12AA [TEN]
  No Virtual link-local address(es):
  No global unicast address is configured
  Joined group address(es):
    FF02::1
    FF02::2
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  Input features: Access List
  Inbound access list MyACL
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  ND advertised default router preference is Medium
  Hosts use stateless autoconfig for addresses.
```

Mr. Odom points out a shorthand command to show all applied access-lists. This can be especially useful on L3 switches, where dozens of interfaces are the norm.

```
R1# show ipv6 interface | include line|list
```

The first vertical line is a "pipe" which feeds the output of the show command to the input of the include command. The second vertical line (same keyboard character) is an "or" that causes the "include" command to grep for either "line" or "list." You may find IOS picky about whitespace—Pipes require spaces on either side, "ors" don't.

```
GigabitEthernet0/0 is administratively down, line protocol is down
GigabitEthernet0/1 is administratively down, line protocol is down
GigabitEthernet0/2 is administratively down, line protocol is down
  Inbound access list MyACL
```

To clear the "hit counters" for an IPv6 ACL,

```
R1# clear access-list counters MyACL
```

Here, unlike IPv4 ACLs, the "ipv6" keyword isn't optional and creates a different command

EXAMPLES

```
R1(config)# ipv6 access-list WEB_SERVER_IN
R1(config-ipv6-acl)# permit icmp any 2001:db8:42::/64 echo-request
                    In IPv4, the echo request message (ping) was simply called "echo."
R1(config-ipv6-acl)# permit tcp any host 2001:db8:42::19 eq 80
R1(config-ipv6-acl)# permit tcp any host 2001:db8:42::19 eq 443
R1(config-ipv6-acl)# permit tcp 2001:db8:22::/64 host 2001:db8:42::19 eq 22 log

R1# show ipv6 access-list WEB_SERVER_IN
IPv6 access list WEB_SERVER_IN
    permit icmp any 2001:DB8:42::/64 echo-request sequence 10
    permit tcp any host 2001:DB8:42::19 eq www sequence 20
    permit tcp any host 2001:DB8:42::19 eq 443 sequence 30
    permit tcp 2001:DB8:22::/64 host 2001:DB8:42::19 eq 22 log sequence 40

R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ipv6 access-list WEB_SERVER_IN
R1(config-ipv6-acl)# deny tcp host 2001:db8:22::12 any eq 22 log sequence 35
                    This inserts a line between 30 and 40. In IPv4, the sequence number started the line.
```

NDP IMPLICIT PERMITS

IPv4 ACLs include an implicit "deny everything else" at the end. IPv6 ACLs have three implied (and invisible) statements:

```
1 permit icmp any any nd-na
2                     Allow NDP Neighbor Advertisements
3 permit icmp any any nd-ns
4                     Allow NDP Neighbor Solicitations
5 deny ipv6 any any
```

This added complexity [Lines 1 and 3, above] exempt some of the ICMPv6 and multicast traffic that IPv6 requires to operate correctly. These permit statements are necessary because

- ICMPv6 is needed for NDP (Neighbor Discovery Protocol), used in neighbor discovery (replacement for IPv4's ARP) and router discovery (used in host self-configuration)
- Several protocols, including NDP, use multicast addresses (IPv6's way to avoid broadcasts), which are vulnerable to the last statement's "any any" keywords. These protocols include:

PROTOCOL OR PROCESS	ASSOCIATED MULTICAST ADDRESSES
ICMPv6 NDP (Neighbor Discovery Protocol) [see CCENT, Chapter 31]	FF02::1 (all nodes)—Used by NDP RA (Router Advertisement) FF02::2 (all routers)—Used by NDP RS (Router Solicitation) FF02:0:0:0:0:1:FF00::/104 (Solicited Mode Multicast)—Used in Neighbor MAC Discovery (ARP replacement) and DAD (Duplicate Address Detection)
DHCPv6 [CCENT, Chapter 31]	FF02::1:2 (All DHCP Agents—Local Scope) FF05::1:3 (Site Scope—requires multicast routing)
OSPFv3 [CCNA, Chapter 23]	FF02::5—Destination for Hellos FF02::6 (Designated Routers)—Destination for Updates
EIGRP [CCNA, Chapter 24]	FF02::A
RIPng	FF02::9

If applied inbound, the three implicit statements don't permit router advertisements (RA), but a router probably won't need to dynamically configure itself based on the advertisement of a neighboring router. Even applied outbound, an ACL won't deprive hosts of the router's own advertisements because internally generated traffic is immune to outbound ACLs.

Pratfall Danger—If you want to log unexpected packets with an explicit "deny ipv6 any any log" line at the end of your ACL, you'll need to also explicitly permit the nd-na and nd-ns messages first, to keep the ordering of permits and denies correct.

IPV6 ACLS IN THE CONTROL-PLANE

IPv6 ACLs can be applied to a vty in the same way as IPv4 ACLs, using "access-class."

RESOURCES

Document	Link
RFC 4890 "Recommendations for Filtering ICMPv6 Messages in Firewalls"	http://www.ietf.org/rfc/rfc4890.txt
NIST Publication 800-119 "Guidelines for the Secure Deployment of IPv6"	http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-119.pdf