IPv6 Security

Filtering and Traffic Analysis

ISP/IXP Workshops

Before we begin…

- Enabling IPv6 on a router means that:
  - The router is accessible by IPv6
  - Interface filters already present in IPv4 must be replicated for IPv6
  - Vty filters already present in IPv4 must be replicated for IPv6
- Failure to protect the router after enabling IPv6 means that it is wide open to abuse through IPv6 transport
  - Even though the IPv4 security is in place

IPv6 Filters

- IPv6 access-lists (ACL) are used to filter traffic and restrict access to the router
  - Used on router interfaces
  - Used to restrict access to the router
  - ACLs matching source/destination addresses, ports and various other IPv6 options
- IPv6 prefix-lists are used to filter routing protocol updates
  - Used on BGP peering
  - Matching source and destination addresses

IPv6 Extended ACL

- Adds support for IPv6 option header and upper layer filtering
- Only named access-lists are supported for IPv6
- IPv6 and IPv4 ACL functionality

  Implicit **deny any any** as final rule in each ACL.

  A reference to an empty ACL will permit any any.

  ACLs are NEVER applied to self-originated traffic.

IPv6 Extended ACL overview

- CLI mirrors IPv4 extended ACL CLI
- Implicit permit rules, enable neighbor discovery
- ULP, DSCP, flow-label,… matches
- Logging
- Time-based
- Reflexive
- CEFv6 and dCEFv6 ACL feature support

IPv6 ACL Implicit Rules

- Implicit permit rules allow neighbor discovery

  The following implicit rules exist at the end of each IPv6 ACL to allow ICMPv6 neighbor discovery:

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

IPv6 Extended ACL Match

- TCP/UDP/SCTP and ports (eq, lt, gt, neq, range)
- ICMPv6 code and type
- Fragments
- Routing Header
- Undetermined transport

  The first unknown NH can be matched against (numerically rather than by name).

  Since an unknown NH cannot be traversed, the ULP cannot be determined.

IPv6 Extended ACL

- Logging

```
ipv6 access-list in-filter
  permit tcp any any log-input
  permit ipv6 any any log
```

- Time based ACLs

```
time-range bar
 periodic daily 10:00 to 13:00
!
ipv6 access-list tin
 deny tcp any any eq www time-range bar
 permit ipv6 any any
```

IPv6 ACL Reflexive

- Reflect

  A reflexive ACL is created dynamically, when traffic matches a permit entry containing the reflect keyword.

  The reflexive ACL mirrors the permit entry and times out (by default after 3 mins), unless further traffic matches the entry (or a FIN is detected for TCP traffic).

  The timeout keyword allows setting a higher or lower timeout value.

  Reflexive ACLs can be applied to TCP, UDP, SCTP and ICMPv6.

- Evaluate

  Apply the packet against a reflexive ACL.

  Multiple evaluate statements are allowed per ACL.

  The implicit deny any any rule does not apply at the end of a reflexive ACL; matching continues after the evaluate in this case.

Cisco IOS IPv6 ACL CLI (1)

- Entering address-family sub-mode

  [no] ipv6 access-list <*name*>

  Add or delete an ACL.

- IPv6 address-family sub-mode

  [no] permit | deny ipv6 | <*protocol*> any | host <*src*> | src/len [*sport*] any | host <*dest*> | dest/len [*dport*] [reflect <*name*> [timeout <*secs*>]] [fragments] [routing] [dscp <*val*>] [flow-label <*val*>][time-range <*name*>] [log | log-input] [sequence <*num*>]

  Permit or deny rule defining the acl entry. Individual entries can be inserted or removed by specifying the sequence number.

  Protocol is one of TCP, UDP, SCTP, ICMPv6 or NH value.

Cisco IOS IPv6 ACL CLI (2)

`[no] evaluate`

Evaluate the dynamically created acl via the permit reflect keyword.

`[no] remark`

User description of an ACL.

- Leaving the sub-mode

  `exit`

- Showing the IPv6 ACL configuration

  `show ipv6 access-list [name]`
  `show access-list [name]`

- Clearing the IPv6 ACL match count

  `clear ipv6 access-list [name]`
  `clear access-list [name]`

Cisco IOS IPv6 ACL CLI (3)

- Applying an ACL to an interface

```
interface fastethernet 0/0
 ipv6 traffic-filter v6in-filter in
 ipv6 traffic-filter v6out-filter out
```

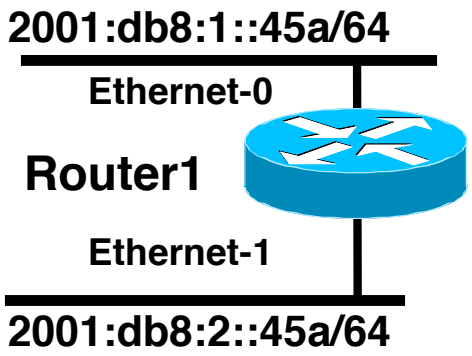
- Restricting access to the router

```
line vty 0 4
 ipv6 access-class vty-filter in
```

- Applying an ACL to filter debug traffic

```
debug ipv6 packet [access-list <acl_name>] [detail]
```

Cisco IOS IPv6 Reflexive ACL



Allow www traffic via a Reflexive ACL, based on time of day

```
Router1#
interface ethernet-0
  ipv6 address 2001:db8:1::45a/64
  ipv6 traffic-filter In in
  ipv6 traffic-filter Out out

interface ethernet-1
  ipv6 address 2001:db8:2::45a/64
  ipv6 traffic-filter Ext-out out

ipv6 access-list In
  permit tcp host 2001:db8:1::1 eq www host 2001:db8:2::2
    time-range tim reflect myp
  permit icmp any any router-solicitation

ipv6 access-list Out
  evaluate myp
  evaluate another

time-range tim
  periodic daily 16:00 to 21:00
```

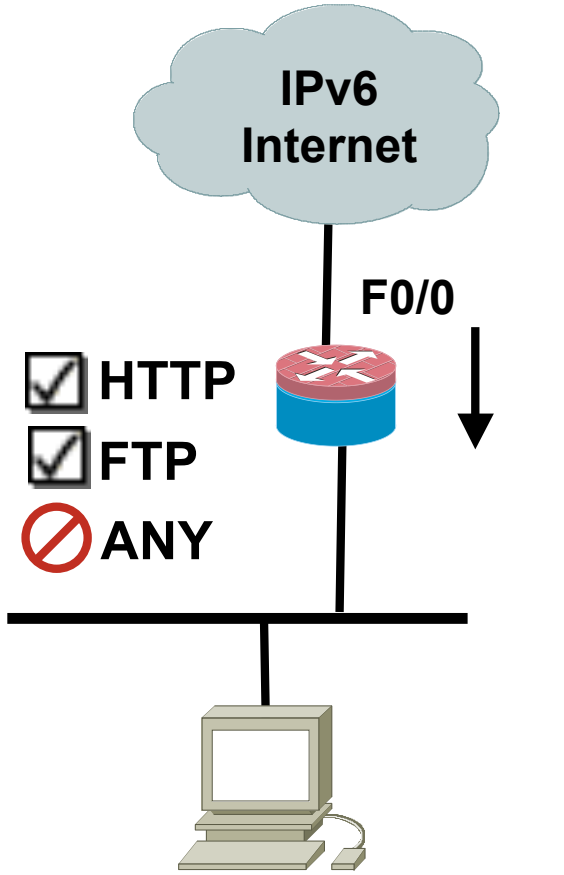
Cisco IOS IPv6 ACL Display

```
brum-45c#show ipv6 access-list
IPv6 access list In
   permit tcp host 2001:db8:1::1 eq www host 2001:db8:2::2 time-range tim (active)
   reflect myp (1 match)
IPv6 access list Out
   evaluate myp
   evaluate another
IPv6 access list myp (Reflexive)
   permit tcp host 2001::2 2432 host 2000::1 eq www (timeout 180)
```

Cisco IOS IPv6 Firewall Feature Set Example: Nothing New from IPv4

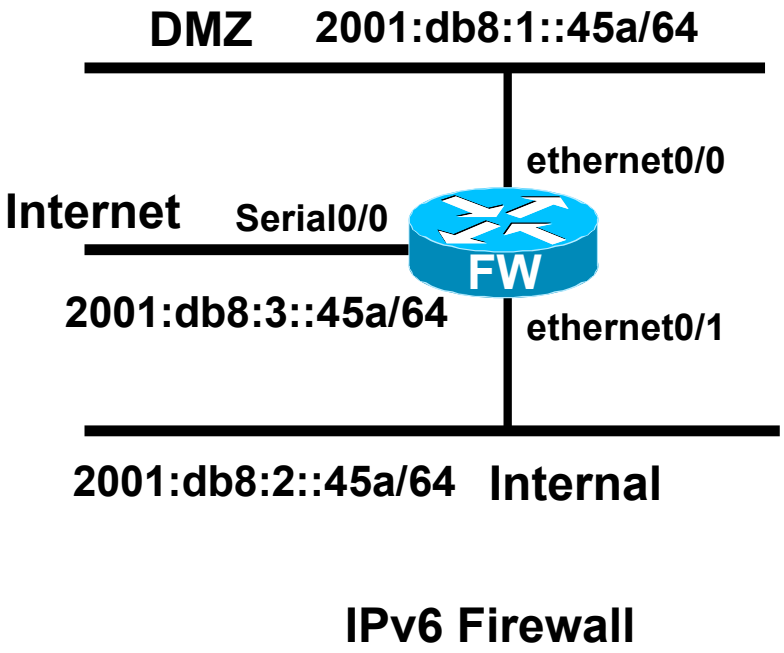
- **Cisco IOS Firewall released 12.3(7)T**

```
ipv6 unicast-routing
ipv6 cef
!
ipv6 inspect audit-trail
ipv6 inspect max-incomplete low 150
ipv6 inspect max-incomplete high 250
ipv6 inspect one-minute low 100
ipv6 inspect one-minute high 200
ipv6 inspect name V6FW tcp timeout 300
ipv6 inspect name V6FW udp
ipv6 inspect name V6FW icmp
!
interface FastEthernet0/0
ipv6 address 2001:DB8:C003:1112::2/64
ipv6 cef
ipv6 traffic-filter EXAMPLE in
ipv6 inspect V6FW in
!
ipv6 access-list EXAMPLE
permit tcp any host 2001:DB8:C003:1113::2 eq www
permit tcp any host 2001:DB8:C003:1113::2 eq ftp
deny ipv6 any any log
```



http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/ps5761/index.html

Cisco IOS IPv6 Firewall (1)



IPv6 Firewall

```
FW#
  interface ethernet0/0
   ipv6 address 2001:db8:1::45a/64
   ipv6 traffic-filter dmz-in6 in
  interface ethernet0/1
   ipv6 address 2001:db8:2::45a/64
   ipv6 traffic-filter internal-in6 in
   ipv6 traffic-filter internal-out6 out
  interface serial0/0
   ipv6 address 2001:db8:3::45a/64
   ipv6 traffic-filter exterior-in6 in
   ipv6 traffic-filter exterior-out6 out

  ipv6 access-list vty
   deny ipv6 any any log-input

  line vty 0 4
   ipv6 access-class vty in

  ipv6 access-list dmz-in6
   permit ipv6 host 2001:db8:1::100 any
```

Cisco IOS IPv6 Firewall (2)

```
ipv6 access-list internal-in6
   permit tcp 2001:db8:2::/64 any reflect internal-tcp
   permit udp 2001:db8:2::/64 any reflect internal-udp
   permit icmp 2001:db8:2::/64 any
   permit icmp any any router-solicitation
ipv6 access-list internal-out6
   evaluate internal-tcp
   evaluate internal-udp
   permit icmp any 2001:db8:2::/64 echo-reply
ipv6 access-list exterior-in6
   evaluate exterior-tcp
   evaluate exterior-udp
   remark Allow access to ftp/http server on the DMZ
   permit tcp any host 2001:db8:1::100 eq ftp
   permit tcp any host 2001:db8:1::100 eq www
   permit tcp any host 2001:db8:1::100 range 49152 65535
   permit icmp any any echo-reply
   permit icmp any any unreachable
   deny ipv6 any any log-input
ipv6 access-list exterior-out6
   permit tcp 2001:db8:2::/64 any reflect exterior-tcp
   permit udp 2001:db8:2::/64 any reflect exterior-udp
```

Cisco IOS IPv6 ACL Troubleshooting

- Access-list status:

  `show ipv6 access-list [<name>]`

  Hit count for matching entries.

  (In)active time-based entries.

- To reset the hit counts for an ACL:

  `clear ipv6 access-list [<aclname>]`

- Configure logging for an ACL entry
- To determine which packets are being dropped by an ACL

  `debug ipv6 packet detail`

Cisco IOS IPv6 NetFlow

- Netflow now supports IPv6
  - Type 9 flow records
  - Available from 12.4 IOS releases
- Activated by:
  - Interface subcommands:
  - **`ipv6 flow ingress`**
  - **`ipv6 flow egress`**
- Status:
  - **`show ipv6 flow cache`**

IPv6 NetFlow

```
gw>show ipv6 flow cache
IP packet size distribution (60682 total packets):
   1-32   64   96  128  160  192  224  256  288  320  352  384  416  448  480
   .000 .060 .260 .482 .013 .007 .003 .005 .002 .002 .001 .001 .001 .001 .002

   512  544  576 1024 1536 2048 2560 3072 3584 4096 4608
   .000 .000 .000 .003 .148 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 475168 bytes
  12 active, 4084 inactive, 34851 added
  566713 ager polls, 0 flow alloc failures
  Active flows timeout in 30 minutes
  Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 33928 bytes
  0 active, 1024 inactive, 0 added, 0 added to flow
  0 alloc failures, 0 force free
  1 chunk, 1 chunk added
SrcAddress         InpIf    DstAddress        OutIf    Prot SrcPrt DstPrt Packets
2001:50...E::2:30 Tu2002   2002:CA...7:FDB2 Fa0/0    0x11 0x0035 0x8001 2
2002:CA...17:FDB2 Fa0/0    2001:50...::2:30 Tu2002   0x11 0x8001 0x0035 2
2002:CA...D8:1::1 Local    2002:CA...7:FDB2 Fa0/0    0x3A 0x0000 0x8800 1
2002:CA...17:FDB2 Fa0/0    FE80::2...6:5580 Local    0x3A 0x0000 0x8800 1
FE80::2...17:FDB2 Fa0/0    FE80::2...6:5580 Local    0x3A 0x0000 0x8800 1
FE80::2...46:5580 Local    FE80::2...7:FDB2 Fa0/0    0x3A 0x0000 0x8800 1
2002:CA...D8:1::1 Local    2002:CA...7:FDB2 Fa0/0    0x11 0xE47C 0x0035 1
```

IPv6 Security

Filtering and Traffic Analysis

ISP/IXP Workshops