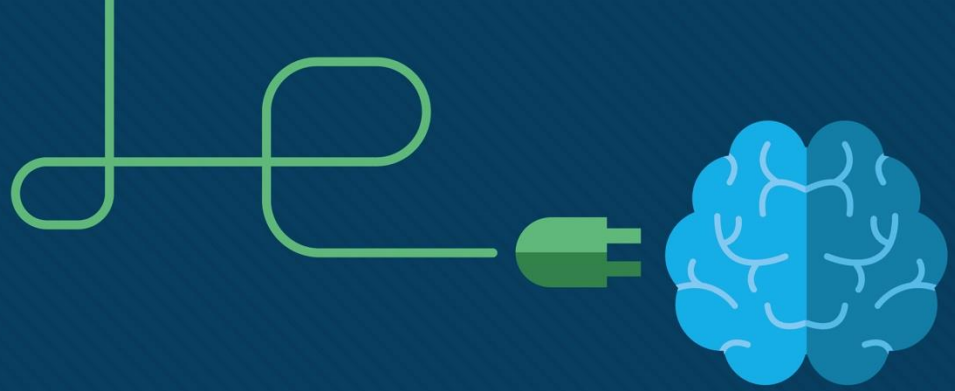


Module 4: ACLs for IPv4 and IPv6 Configuration

Enterprise Networking, Security, and Automation v7.0 (ENSA)

4.1 Configure Extended IPv4 ACLs

Extended ACLs

Extended ACLs provide a greater degree of control. They can filter on source address, destination address, protocol (i.e., IP, TCP, UDP, ICMP), and port number.

Extended ACLs can be created as:

- **Numbered Extended ACL** - Created using the **access-list** *access-list-number* global configuration command.
- **Named Extended ACL** - Created using the **ip access-list extended** *access-list-name*.

Protocols and Ports

Extended ACLs can filter on internet protocols and ports. Use the **?** to get help when entering a complex ACE. The four highlighted protocols are the most popular options.

Protocol Options

```
R1(config)# access-list 100 permit ?
  <0-255>        An IP protocol number
  ahp            Authentication Header Protocol
  dvmrp          dvmrp
  eigrp          Cisco's EIGRP routing protocol
  esp            Encapsulation Security Payload
  gre            Cisco's GRE tunneling
  icmp           Internet Control Message Protocol
  igmp           Internet Gateway Message Protocol
  ip             Any Internet Protocol
  ipinip         IP in IP tunneling
  nos            KA9Q NOS compatible IP over IP tunneling
  object-group   Service object group
  ospf           OSPF routing protocol
  pcp            Payload Compression Protocol
  pim            Protocol Independent Multicast
  tcp            Transmission Control Protocol
  udp            User Datagram Protocol
R1(config)# access-list 100 permit
```

Configure Extended IPv4 ACLs
Protocols and Ports (Cont.)

Selecting a protocol influences port options. Many TCP port options are available, as shown in the output.

```
R1(config)# access-list 100 permit tcp any any eq ?
  <0-65535>    Port number
  bgp          Border Gateway Protocol (179)
  chargen      Character generator (19)
  cmd          Remote commands (rcmd, 514)
  daytime      Daytime (13)
  discard      Discard (9)
  domain       Domain Name Service (53)
  echo         Echo (7)
  exec         Exec (rsh, 512)
  finger       Finger (79)
  ftp          File Transfer Protocol (21)
  ftp-data     FTP data connections (20)
  gopher       Gopher (70)
  hostname     NIC hostname server (101)
  ident        Ident Protocol (113)
  irc          Internet Relay Chat (194)
  klogin       Kerberos login (543)
  kshell       Kerberos shell (544)
  login        Login (rlogin, 513)
  lpd          Printer service (515)
  msrpc        MS Remote Procedure Call (135)
  nntp         Network News Transport Protocol (119)
  onep-plain   Onep Cleartext (15001)
  onep-tls     Onep TLS (15002)
  pim-auto-rp  PIM Auto-RP (496)
  pop2         Post Office Protocol v2 (109)
  pop3         Post Office Protocol v3 (110)
  smtp         Simple Mail Transport Protocol (25)
  sunrpc       Sun Remote Procedure Call (111)
  syslog       Syslog (514)
  tacacs       TAC Access Control System (49)
  talk         Talk (517)
  telnet       Telnet (23)
  time         Time (37)
  uucp         Unix-to-Unix Copy Program (540)
  whois        Nicname (43)
  www          World Wide Web (HTTP, 80)
```

Protocols and Port Numbers Configuration Examples

Extended ACLs can filter on different port number and port name options.

This example configures an extended ACL 100 to filter HTTP traffic. The first ACE uses the **www** port name. The second ACE uses the port number **80**. Both ACEs achieve exactly the same result.

```
R1(config)# access-list 100 permit tcp any any eq www
!or...
R1(config)# access-list 100 permit tcp any any eq 80
```

Configuring the port number is required when there is not a specific protocol name listed such as SSH (port number 22) or an HTTPS (port number 443), as shown in the next example.

```
R1(config)# access-list 100 permit tcp any any eq 22
R1(config)# access-list 100 permit tcp any any eq 443
R1(config)#
```

Apply a Numbered Extended IPv4 ACL

In this example, the ACL permits both HTTP and HTTPS traffic from the 192.168.10.0 network to go to any destination.

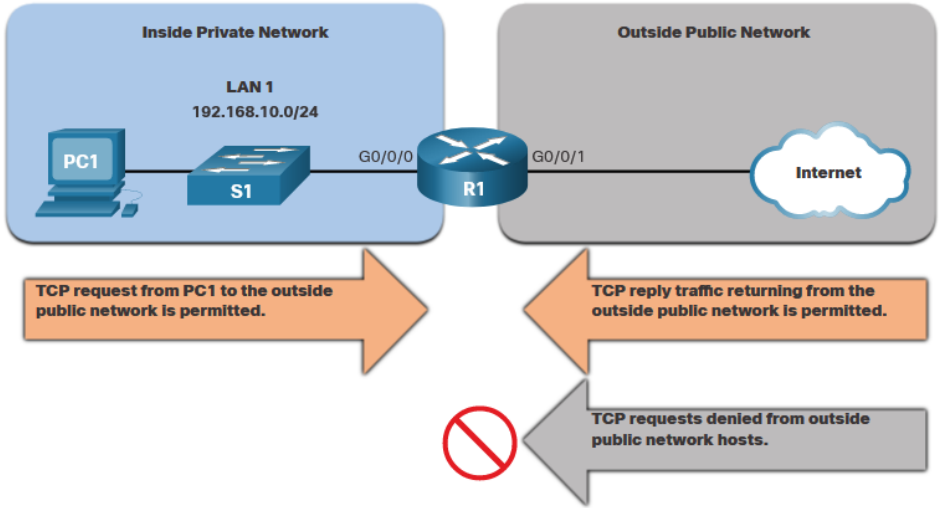
Extended ACLs can be applied in various locations. However, they are commonly applied close to the source. Here ACL 110 is applied inbound on the R1 G0/0/0 interface.

```
R1(config)# access-list 110 permit tcp 192.168.10.0 0.0.0.255 any eq www
R1(config)# access-list 110 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)# interface g0/0/0
R1(config-if)# ip access-group 110 in
R1(config-if)# exit
R1(config)#
```

TCP Established Extended ACL

TCP can also perform basic stateful firewall services using the TCP **established** keyword.

- The **established** keyword enables inside traffic to exit the inside private network and permits the returning reply traffic to enter the inside private network.
- TCP traffic generated by an outside host and attempting to communicate with an inside host is denied.



TCP Established Extended ACL (Cont.)

- ACL 120 is configured to only permit returning web traffic to the inside hosts. The ACL is then applied outbound on the R1 G0/0/0 interface.
- The **show access-lists** command shows that inside hosts are accessing the secure web resources from the internet.

Note: A match occurs if the returning TCP segment has the ACK or reset (RST) flag bits set, indicating that the packet belongs to an existing connection.

```
R1(config)# access-list 120 permit tcp any 192.168.10.0 0.0.0.255 established
R1(config)# interface g0/0/0
R1(config-if)# ip access-group 120 out
R1(config-if)# end
R1# show access-lists
Extended IP access list 110
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443 (657 matches)
Extended IP access list 120
    10 permit tcp any 192.168.10.0 0.0.0.255 established (1166 matches)
R1#
```

Named Extended IPv4 ACL Syntax

Naming an ACL makes it easier to understand its function. To create a named extended ACL, use the **ip access-list extended** configuration command.

In the example, a named extended ACL called NO-FTP-ACCESS is created and the prompt changed to named extended ACL configuration mode. ACE statements are entered in the named extended ACL sub configuration mode.

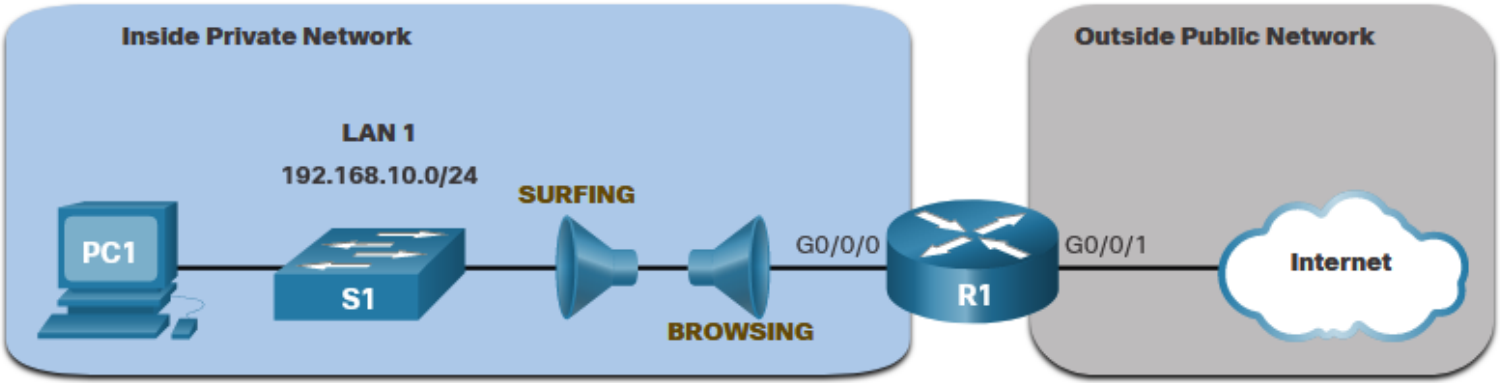
```
Router(config)# ip access-list extended access-list-name
```

```
R1(config)# ip access-list extended NO-FTP-ACCESS
R1(config-ext-nacl)#
```

Named Extended IPv4 ACL Example

The topology below is used to demonstrate configuring and applying two named extended IPv4 ACLs to an interface:

- **SURFING** - This will permit inside HTTP and HTTPS traffic to exit to the internet.
- **BROWSING** - This will only permit returning web traffic to the inside hosts while all other traffic exiting the R1 G0/0/0 interface is implicitly denied.



Named Extended IPv4 ACL Example (Cont.)

- The SURFING ACL permits HTTP and HTTPS traffic from inside users to exit the G0/0/1 interface connected to the internet. Web traffic returning from the internet is permitted back into the inside private network by the BROWSING ACL.
- The SURFING ACL is applied inbound and the BROWSING ACL is applied outbound on the R1 G0/0/0 interface.

```
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# Remark Permits inside HTTP and HTTPS traffic
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config-ext-nacl)# exit
R1(config)#
R1(config)# ip access-list extended BROWSING
R1(config-ext-nacl)# Remark Only permit returning HTTP and HTTPS traffic
R1(config-ext-nacl)# permit tcp any 192.168.10.0 0.0.0.255 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0/0
R1(config-if)# ip access-group SURFING in
R1(config-if)# ip access-group BROWSING out
R1(config-if)# end
R1# show access-lists
Extended IP access list SURFING
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443 (124 matches)
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established (369 matches)
R1#
```

Named Extended IPv4 ACL Example (Cont.)

The show access-lists command is used to verify the ACL statistics. Notice that the permit secure HTTPS counters (i.e., eq 443) in the SURFING ACL and the return established counters in the BROWSING ACL have increased.

```
R1# show access-lists
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
    10 permit tcp 19.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1#
```

Edit Extended ACLs

An extended ACL can be edited using a text editor when many changes are required. Or, if the edit applies to one or two ACEs, then sequence numbers can be used.

Example:

- The ACE sequence number 10 in the SURFING ACL has an incorrect source IP networks address.

```
R1# show access-lists
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
    10 permit tcp 19.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1#
```

Edit Extended ACLs (Cont.)

- To correct this error the original statement is removed with the **no** *sequence_#* command and the corrected statement is added replacing the original statement.
- The **show access-lists** command output verifies the configuration change.

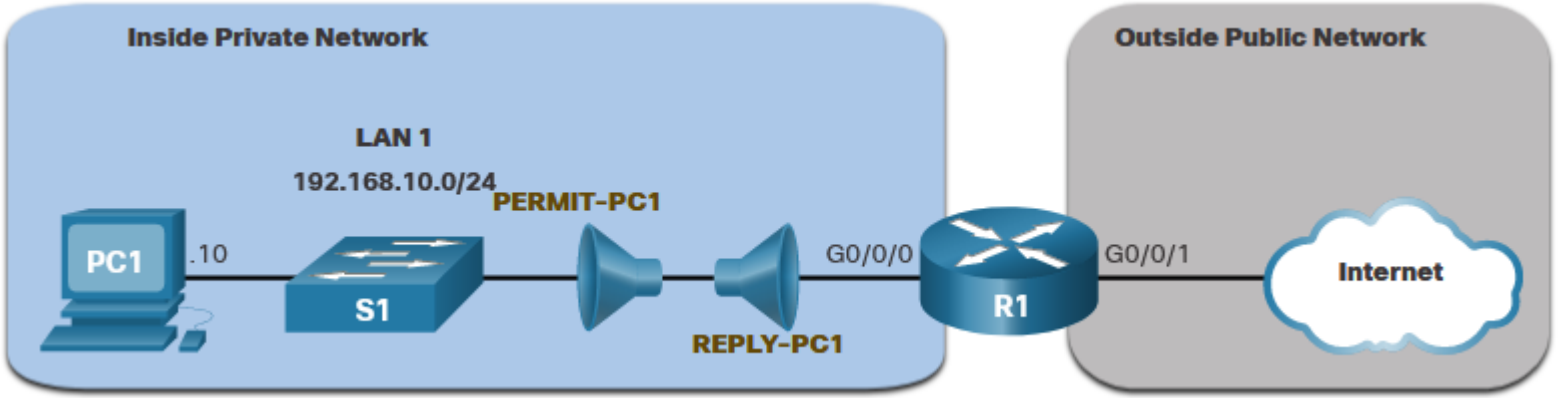
```
R1# configure terminal
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# no 10
R1(config-ext-nacl)# 10 permit tcp 192.168.10.0 0.0.0.255 any eq www
R1(config-ext-nacl)# end
```

```
R1# show access-lists
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established
Extended IP access list SURFING
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1#
```

Another Extended IPv4 ACL Example

Two named extended ACLs will be created:

- **PERMIT-PC1** - This will only permit PC1 TCP access to the internet and deny all other hosts in the private network.
- **REPLY-PC1** - This will only permit specified returning TCP traffic to PC1 implicitly deny all other traffic.



Another Extended IPv4 ACL Example (Cont.)

- The **PERMIT-PC1** ACL permits PC1 (192.168.10.10) TCP access to the FTP, SSH, Telnet, DNS , HTTP, and HTTPS traffic.
- The **REPLY-PC1** ACL will permit return traffic to PC1.
- The **PERMIT-PC1** ACL is applied inbound and the **REPLY-PC1** ACL applied outbound on the R1 G0/0/0 interface.

```
R1(config)# ip access-list extended PERMIT-PC1
R1(config-ext-nacl)# Remark Permit PC1 TCP access to internet
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 20
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 21
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 22
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 23
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 53
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 80
R1(config-ext-nacl)# permit tcp host 192.168.10.10 any eq 443
R1(config-ext-nacl)# deny ip 192.168.10.0 0.0.0.255 any
R1(config-ext-nacl)# exit
R1(config)#
R1(config)# ip access-list extended REPLY-PC1
R1(config-ext-nacl)# Remark Only permit returning traffic to PC1
R1(config-ext-nacl)# permit tcp any host 192.168.10.10 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0/0
R1(config-if)# ip access-group PERMIT-PC1 in
R1(config-if)# ip access-group REPLY-PC1 out
R1(config-if)# end
R1#
```

Verify Extended ACLs

The **show ip interface** command is used to verify the ACL on the interface and the direction in which it was applied.

```
R1# show ip interface g0/0/0
GigabitEthernet0/0/0 is up, line protocol is up (connected)
  Internet address is 192.168.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is REPLY-PC1
  Inbound  access list is PERMIT-PC1
  Proxy ARP is enabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is disabled
  IP fast switching on the same interface is disabled
  IP Flow switching is disabled
  IP Fast switching turbo vector
  IP multicast fast switching is disabled
  IP multicast distributed fast switching is disabled
  Router Discovery is disabled
R1#
R1# show ip interface g0/0/0 | include access list
Outgoing access list is REPLY-PC1
Inbound access list is PERMIT-PC1
R1#
```

Verify Extended ACLs (Cont.)

The **show access-lists** command can be used to confirm that the ACLs work as expected. The command displays statistic counters that increase whenever an ACE is matched.

Note: Traffic must be generated to verify the operation of the ACL.

```
R1# show access-lists
Extended IP access list PERMIT-PC1
10 permit tcp host 192.168.10.10 any eq 20
20 permit tcp host 192.168.10.10 any eq ftp
30 permit tcp host 192.168.10.10 any eq 22
40 permit tcp host 192.168.10.10 any eq telnet
50 permit tcp host 192.168.10.10 any eq domain
60 permit tcp host 192.168.10.10 any eq www
70 permit tcp host 192.168.10.10 any eq 443
80 deny ip 192.168.10.0 0.0.0.255 any
Extended IP access list REPLY-PC1
10 permit tcp any host 192.168.10.10 established
R1#
```

Verify Extended ACLs (Cont.)

The **show running-config** command can be used to validate what was configured. The command also displays configured remarks.

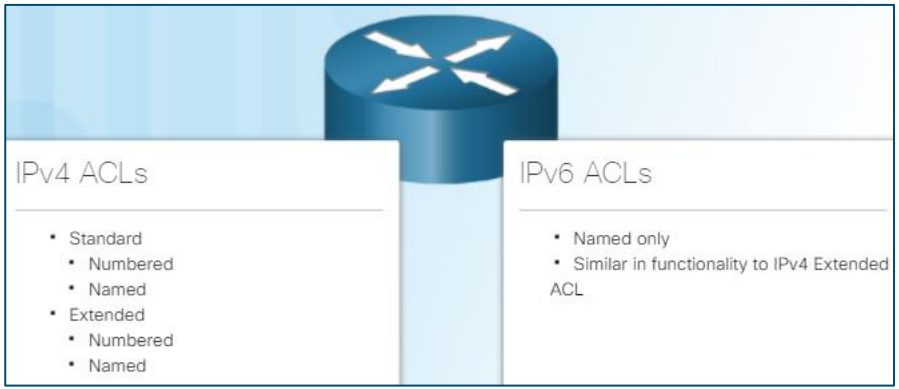
```
R1# show running-config | begin ip access-list
ip access-list extended PERMIT-PC1
remark Permit PC1 TCP access to internet
permit tcp host 192.168.10.10 any eq 20
permit tcp host 192.168.10.10 any eq ftp
permit tcp host 192.168.10.10 any eq 22
permit tcp host 192.168.10.10 any eq telnet
permit tcp host 192.168.10.10 any eq domain
permit tcp host 192.168.10.10 any eq www
permit tcp host 192.168.10.10 any eq 443
deny ip 192.168.10.0 0.0.0.255 any
ip access-list extended REPLY-PC1
remark Only permit returning traffic to PC1
permit tcp any host 192.168.10.10 established
!
```

4.2 IPv6 ACLs

IPv6 ACL Creation

- IPv6 ACLs are similar to IPv4 ACLs in both operation and configuration.

In IPv4 there are two types of ACLs, standard and extended and both types of ACLs can be either numbered or named ACLs.



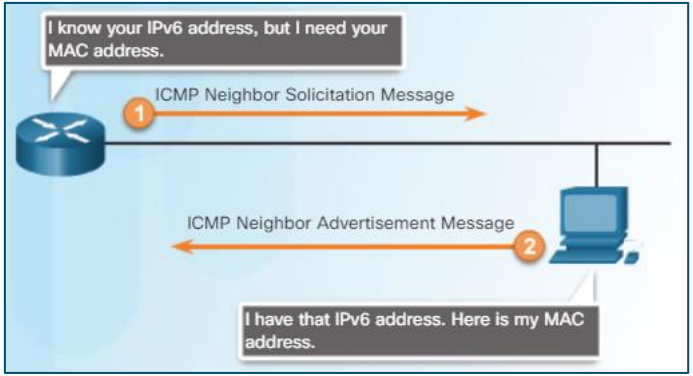
With IPv6, there is only one type of ACL, which is equivalent to an IPv4 extended named ACL and there are no numbered ACLs in IPv6.

- **Note:**
  - An IPv4 ACL and an IPv6 ACL cannot share the same name.

IPv6 ACL Creation

- There are three significant differences between IPv4 and IPv6 ACLs:
  - The command used to apply an IPv6 ACL to an interface is **ipv6 traffic-filter** command.
  - IPv6 ACLs do not use wildcard masks but instead specifies the prefix-length to indicate how much of an IPv6 source or destination address should be matched.
  - An IPv6 ACL adds two implicit permit statements at the end of each IPv6 access list.
    - **permit icmp any any nd-na**
    - **permit icmp any any nd-ns**
    - **deny ipv6 any any statement**

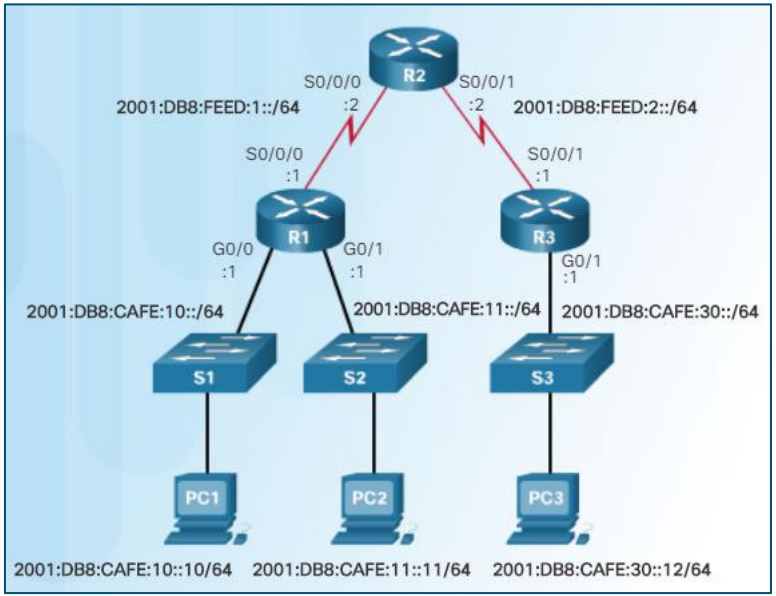
- These two additional statements allow IPv6 ICMP Neighbor Discovery (ND) and Neighbor Solicitation (NS) messages to accomplish the same thing as IPv4 ARP.



Configuring IPv6 ACLs

- The following is the sample topology that will be used to demonstrate IPv6 ACLs.
  - All interfaces are configured and active.

```
R1# show ipv6 interface brief
GigabitEthernet0/0      [up/up]
    FE80::FE99:47FF:FE75:C3E0
    2001:DB8:CAFE:10::1
GigabitEthernet0/1      [up/up]
    FE80::FE99:47FF:FE75:C3E1
    2001:DB8:CAFE:11::1
Serial0/0/0             [up/up]
    FE80::FE99:47FF:FE75:C3E0
    2001:DB8:FEED:1::1
 <output omitted>
R1#
```




```
R2# show ipv6 interface brief
Serial0/0/0             [up/up]
    FE80::FE99:47FF:FE71:78A0
    2001:DB8:FEED:1::2
Serial0/0/1             [up/up]
    FE80::FE99:47FF:FE71:78A0
    2001:DB8:FEED:2::2
 <output omitted>
R2#
```

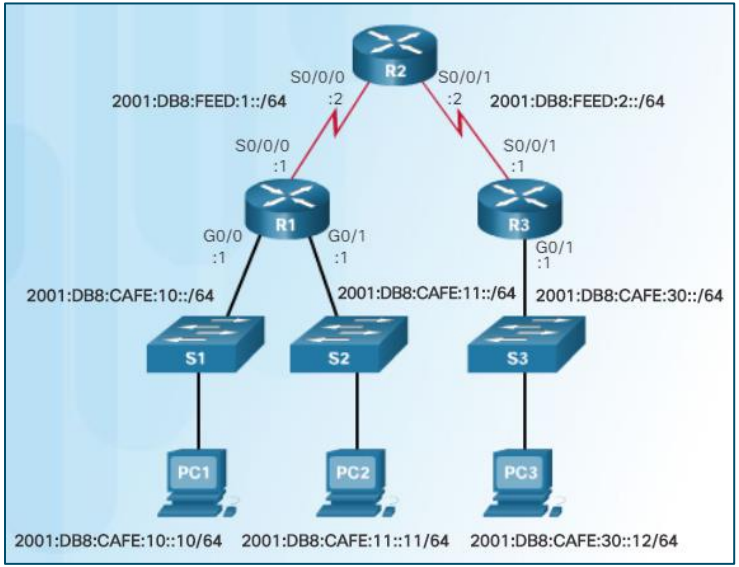
```
R3# show ipv6 interface brief
GigabitEthernet0/0      [up/up]
    FE80::FE99:47FF:FE71:7A20
    2001:DB8:CAFE:30::1
Serial0/0/1             [up/up]
    FE80::FE99:47FF:FE71:7A20
    2001:DB8:FEED:2::1
R3#
```

Configuring IPv6 ACLs

- In IPv6 there are only named ACLs and the configuration is similar to IPv4 extended named ACLs.

```
R1(config)# ipv6 access-list access-list-name
R1(config-ipv6-acl)# deny | permit protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address}
[operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address} [operator
[port-number]]
```

```
R1(config)# ipv6 access-list NO-R3-LAN-ACCESS
R1(config-ipv6-acl)# deny ipv6 2001:db8:cafe:30::/64 any
R1(config-ipv6-acl)# permit ipv6 any any
R1(config-ipv6-acl)# end
R1#
```



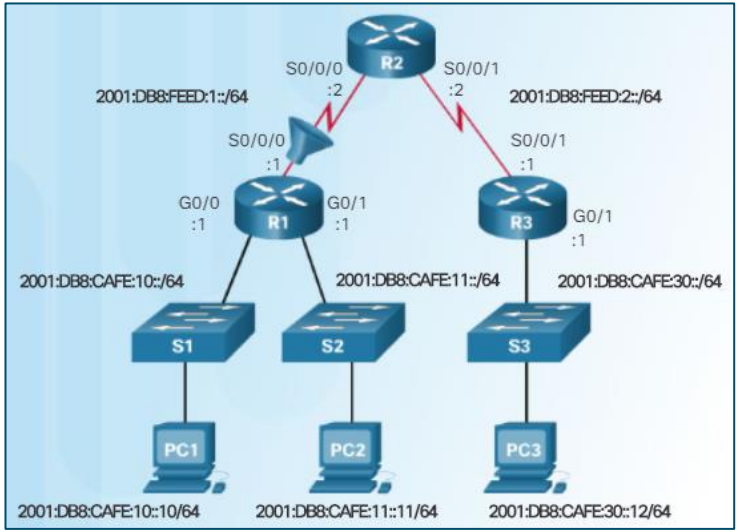
- In this example:
  - The 1st statement names the IPv6 ACL **NO-R3-LAN-ACCESS**.
  - The 2nd statement denies all IPv6 packets from the 2001:DB8:CAFE:30::/64 destined for any IPv6 network.
  - The 3rd statement allows all other IPv6 packets.

Configuring IPv6 ACLs

- After an IPv6 ACL is configured, it is linked to an interface using the following interface command:
  - **ipv6 traffic-filter** *access-list-name* {**in** | **out**}

The command applies the NO-R3-LAN-ACCESS IPv6 ACL inbound to the S0/0/0 interface of R1.

```
R1(config)# interface s0/0/0
R1(config-if)# ipv6 traffic-filter NO-R3-LAN-ACCESS in
```



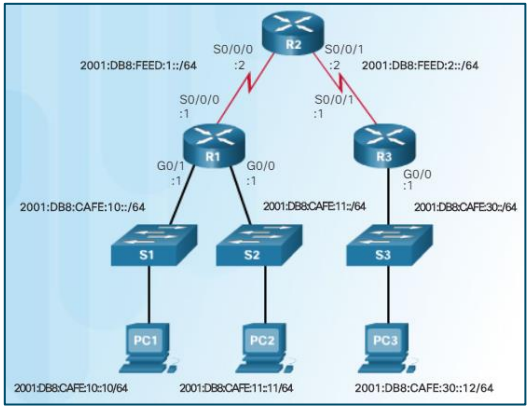
- To remove an IPv6 ACL, enter the **no ipv6 traffic-filter** command on the interface, and then enter the global **no ipv6 access-list** command to remove the access list.

- Note that IPv4 and IPv6 both use the **access-class** command to apply an access list to VTY ports.

Configuring IPv6 ACLs

- In this example, an IPv6 ACL permits R3 LAN users limited access to the LANs on R1.

1. These ACES allow access from any device to the web server (2001:DB8:CAFE:10::10).
2. All other devices are denied access to the 2001:DB8:CAFE:10::/64 network.
3. PC3 (2001:DB8:CAFE:30::12) is permitted Telnet access to PC2 (2001:DB8:CAFE:11::11).
4. All others are denied Telnet access to PC2.
5. All other IPv6 traffic is permitted to all other destinations.
6. The IPv6 access list is applied inbound on G0/0 so only the 2001:DB8:CAFE:30::/64 network is affected.




```
R3(config)# ipv6 access-list RESTRICTED-ACCESS
R3(config-ipv6-acl)# remark Permit access only HTTP and HTTPS to Network 10
R3(config-ipv6-acl)# permit tcp any host 2001:db8:cafe:10::10 eq 80
R3(config-ipv6-acl)# permit tcp any host 2001:db8:cafe:10::10 eq 443
R3(config-ipv6-acl)# remark Deny all other traffic to Network 10
R3(config-ipv6-acl)# deny ipv6 any 2001:db8:cafe:10::/64
R3(config-ipv6-acl)# remark Permit PC3 telnet access to PC2
R3(config-ipv6-acl)# permit tcp host 2001:DB8:CAFE:30::12 host 2001:DB8:CAFE:11::11 eq 23
R3(config-ipv6-acl)# remark Deny telnet access to PC2 for all other devices
R3(config-ipv6-acl)# deny tcp any host 2001:db8:cafe:11::11 eq 23
R3(config-ipv6-acl)# remark Permit access to everything else
R3(config-ipv6-acl)# permit ipv6 any any
R3(config-ipv6-acl)# exit
R3(config)# interface g0/0
R3(config-if)# ipv6 traffic-filter RESTRICTED-ACCESS in
R3(config-if)#
```

1
2
3
4
5
6

Configuring IPv6 ACLs

- The commands used to verify an IPv6 access list are similar to those used for IPv4 ACLs.
- Use the **show ipv6 interface** command to see which ACL and direction is configured on an interface.

```
R3# show ipv6 interface g0/0
GigabitEthernet0/0 is up, line protocol is up
 Global unicast address(es):
  2001:DB8:CAFE:30::1, subnet is 2001:DB8:CAFE:30::/64
 Input features: Access List
 Inbound access list RESTRICTED-ACCESS
<output omitted>
```

- Use the **show access-lists** command displays all configured IPv4 and IPv6 access lists
  - Notice that IPv6 ACL sequence numbers are displayed at the end of the ACE.

```
R3# show access-lists
IPv6 access list RESTRICTED-ACCESS
  permit tcp any host 2001:DB8:CAFE:10::10 eq www sequence 20
  permit tcp any host 2001:DB8:CAFE:10::10 eq 443 sequence 30
  deny ipv6 any 2001:DB8:CAFE:10::/64 sequence 50
  permit tcp host 2001:DB8:CAFE:30::12 host 2001:DB8:CAFE:11::11 eq
  telnet sequence 70
  deny tcp any host 2001:DB8:CAFE:11::11 eq telnet sequence 90
  permit ipv6 any any sequence 110
R3#
```

- The **show running-config** command displays all of the ACEs and remark statements.


CISCO