IPv6 ACLs

Adapted from a slide deck courtesy of Bob Vachon

Contents

Overview	slide 3
IPv4 vs IPv6	slide 4-6
Syntax & Config	slides 7-8
Examples 1-4	slides 9-18

New Terminology

ipv6 traffic-filter
ipv6 access-class
permit icmp any any nd-na
permit icmp any any nd-ns

IPv6 ACLs

- IPv6 attacks are becoming more common therefore it is necessary to develop and implement a strategy to mitigate attacks against IPv6 infrastructures and protocols.
  - This mitigation strategy includes filtering at the edge using IPv6 ACLs.

- IPv6 ACLs are similar to IPv4 ACLs.
  - They allow filtering on source and destination addresses, source and destination ports, and protocol type.

- However:
  - There are no standard ACLs.
  - ACLs must be configured with a name.
  - They are applied to an interface with the **`ipv6 traffic-filter`** command.

Differences Between IPv4 and IPv6 ALCs

- **Applying an IPv6 ACL**
  - IPv4 uses the `ip access-group` interface command.
  - IPv6 uses the `ipv6 traffic-filter` interface command.
  - Use the `ipv6 access-class` command to apply an IPv6 ACL to VTY interfaces.

- **No Wildcard Masks**
  - IPv4 uses a wildcard mask
  - IPv6 uses the prefix-length to indicate how much of an IPv6 source or destination address should be matched (equivalent to the /xx CIDR mask in IPv4).

- **Additional Default Statements**
  - IPv4 ACLs have an implicit deny ACE
  - IPv6 has two implicit **permit** statements at the end of each IPv6 access list followed by an implicit **deny** ACE.

IPv6 ACL Implicit Entries

- IP ACLS include 3 implicit commands at the end of each ACL.
  - These statements will not be displayed in the configuration output.
  - 2 IPv6 neighbor discovery ACE (i.e, ND provides services similar to what ARP does for IPv4)
  - 1 implicit deny all ACE

- These 3 IP ACLS ACES are:
  - `permit icmp any any nd-na` **Permits IPv6 ND**
  - `permit icmp any any nd-ns` **Permits IPv6 ND**
  - `deny ipv6 any any`

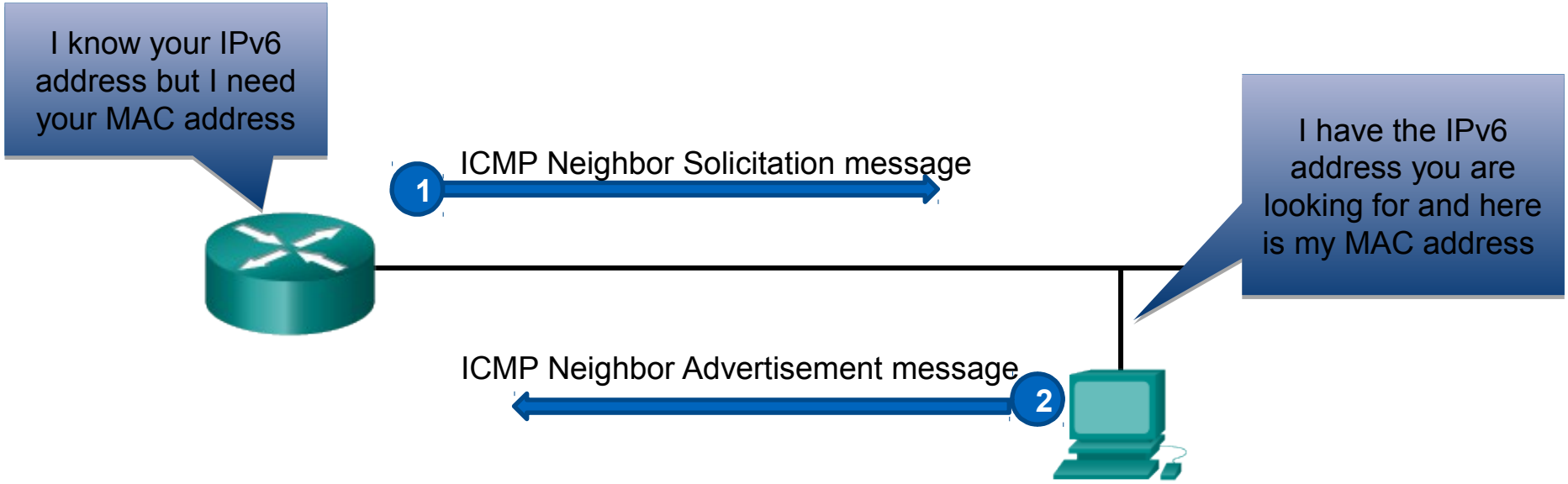
Caution:

- Manually entering **deny ipv6 any any** would deny the implicit permit ND packets.
- A best practice is to **manually enter all 3 implicit commands**.

Comparing IPv4 and IPv6 ACLs

- IPv6 includes two implicit permit icmpV6 statements
  - `permit icmp any any nd-na`
  - `permit icmp any any nd-ns`

- These two messages are used for Neighbor Discovery messages and provide the equivalent as the ARP IPv4 feature.
  - IPv6 ACLs need to implicitly permit ND packets to be sent and received on an interface.



IPv6 ACLs

```
R1(config)# ipv6 access-list ACL-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]]
```

Parameter	Description
deny | **permit**	• Specifies whether to deny or permit the packet.
protocol	• Enter the name or number of an Internet protocol, or an integer representing an IPv6 protocol number.
source-ipv6-prefix/prefix-length *destination-ipv6-address/prefix-length*	• The source or destination IPv6 network or class of networks for which to set deny or permit conditions
any	• Enter **any** as an abbreviation for the IPv6 prefix ::/0. • This matches all addresses.
host	• For **host** *source-ipv6-address or destination-ipv6-address*, enter the source or destination IPv6 host address for which to set deny or permit conditions
operator	• (Optional) An operand that compares the source or destination ports of the specified protocol. • Operands are **lt** (less than), **gt** (greater than), **eq** (equal), **neq** (not equal), and **range**.
port-number	• (Optional) A decimal number or the name of a TCP or UDP port for filtering TCP or UDP, respectively.

IPv6 ACL Configuration

- Create an IPv6 ACL.

```
Router(config)#
ipv6 access-list ACL-NAME
```

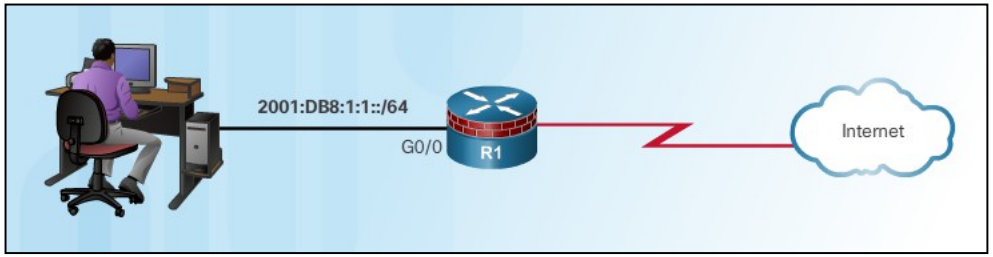
```
Router(config-ipv6-acl)#
{permit | deny} protocol [source-ipv6-prefix/prefix-length]
[operator operand] [destination-ipv6-prefix/prefix length]
[operator operand]
```

- Apply an IPv6 ACL to an interface.

```
Router(config-if)#
ipv6 traffic-filter ACL-NAME {in | out}
```

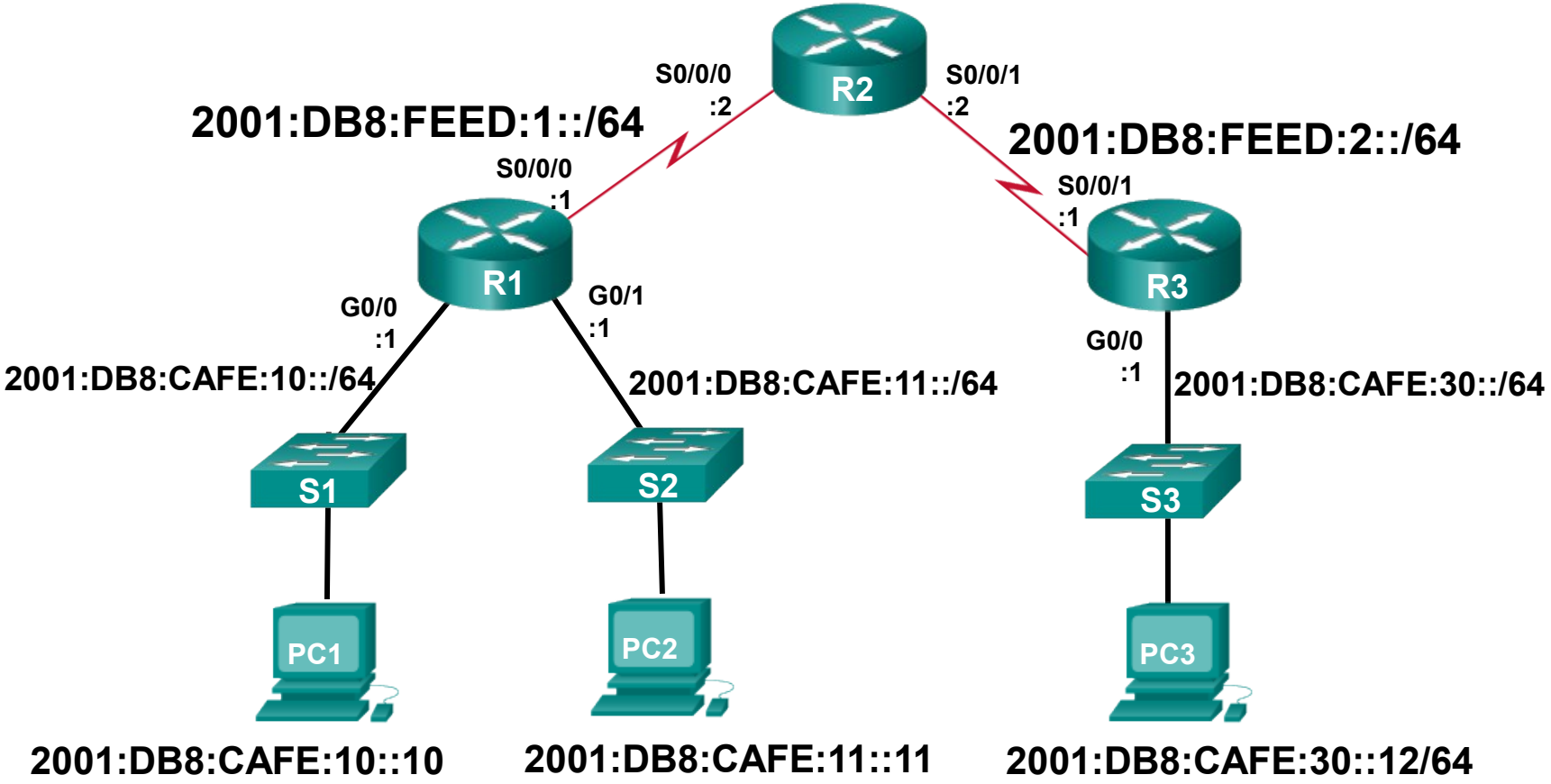
IPv6 ACL Example #1

- Create an IPv6 ACL named **LAN_ONLY** that:
  - Permits inside addresses to exit
  - Permit IPv6 Neighbor discovery
  - Deny all other and log
  - Applied incoming to G0/0



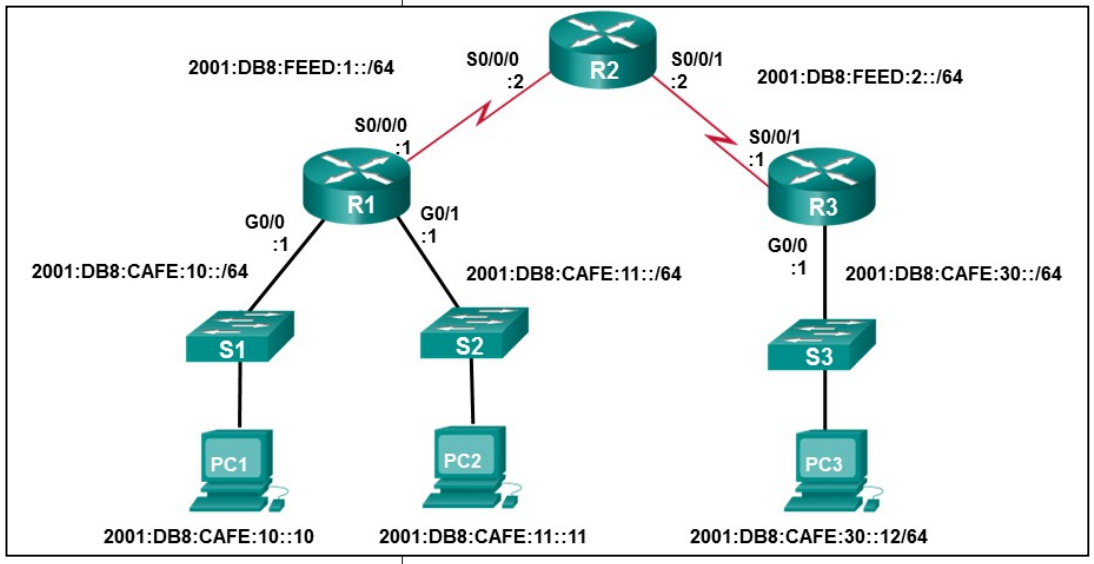

```
R1(config)# ipv6 access-list LAN_ONLY
R1(config-ipv6-acl)# permit ipv6 2001:db8:1:1::/64 any
R1(config-ipv6-acl)# permit icmp any any nd-na
R1(config-ipv6-acl)# permit icmp any any nd-ns
R1(config-ipv6-acl)# deny ipv6 any any log
R1(config-ipv6-acl)# exit
R1(config)# interface G0/0
R1(config-if)# ipv6 traffic-filter LAN_ONLY in
R1(config-if)# end
R1#
R1# show ipv6 access-list
IPv6 access list LAN_ONLY
    permit ipv6 2001:DB8:1:1::/64 any sequence 10
    permit icmp any any nd-na sequence 20
    permit icmp any any nd-ns sequence 30
    deny ipv6 any any sequence 40
```

IPv6 ACL Topology – Examples #2-4



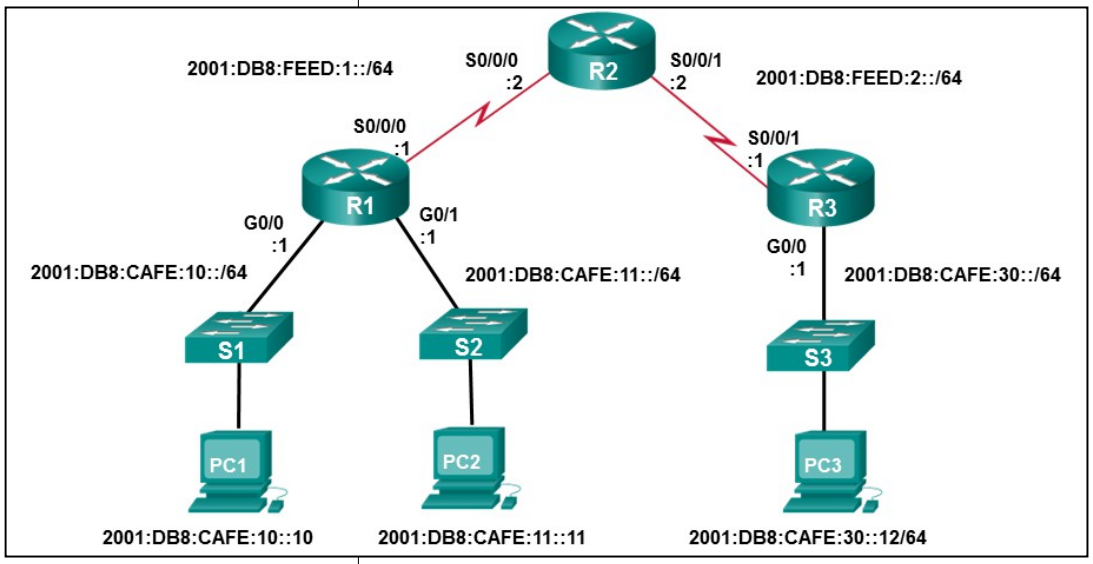
R1 Configuration

```
R1(config)# interface g0/0
R1(config-if)# ipv6 address 2001:db8:cafe:10::1/64
R1(config-if)# exit
R1(config)# interface g0/1
R1(config-if)# ipv6 address 2001:db8:cafe:11::1/64
R1(config-if)# exit
R1(config)# interface s0/0/0
R1(config-if)# ipv6 address 2001:db8:feed:1::1/64
R1(config-if)# end
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
<some output omitted for brevity>
R1#
```



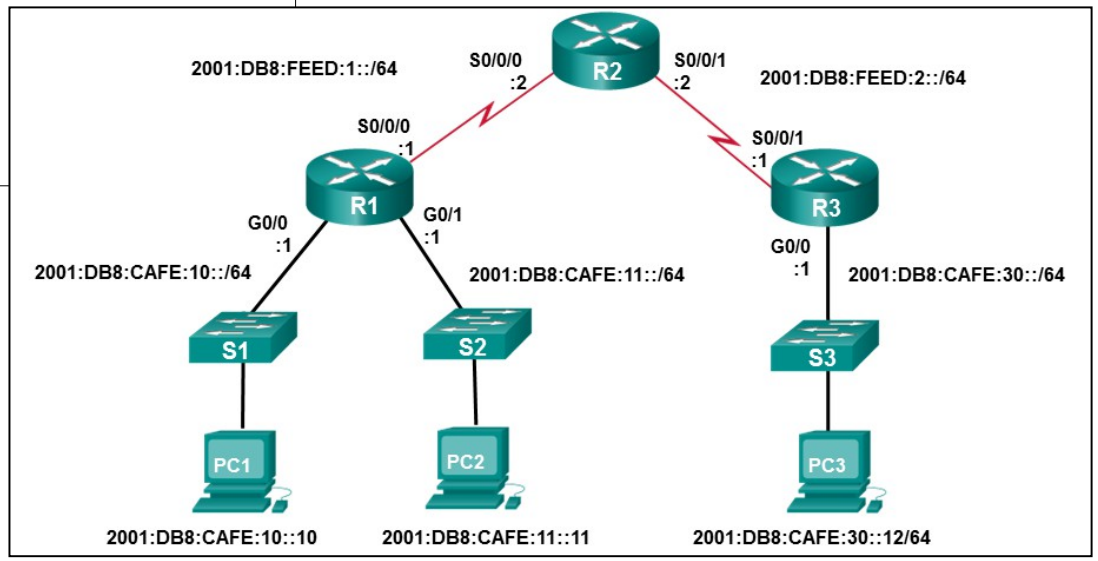
R2 Configuration

```
R2(config)# interface s0/0/0
R2(config-if)# ipv6 address
2001:db8:feed:1::2/64
R2(config-if)# exit
R2(config)# interface s0/0/1
R2(config-if)# ipv6 address
2001:db8:feed:2::2/64
R2(config-if)# end
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
<some output omitted for brevity>
R2#
```



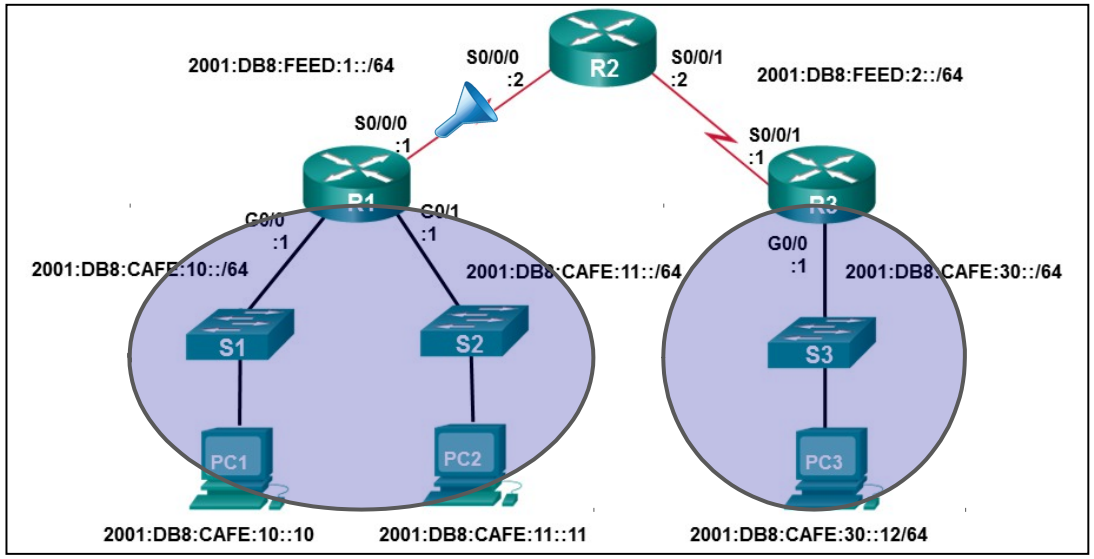
R3 Configuration

```
R3(config)# interface s0/0/1
R3(config-if)# ipv6 address
2001:db8:feed:2::1/64
R3(config-if)# exit
R3(config)# interface g0/0
R3(config-if)# ipv6 address
2001:db8:cafe:30::1/64
R3(config-if)# end
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#
```



IPv6 ACL Example #2

- Create an ACL called NO-R3-LAN-ACCESS.
- Deny all IPv6 packets from the 2001:DB8:CAFE:30::/64 destined for the R1 LANs.
- Allow all other IPv6 packets.
- Apply it to the R1 serial interface.

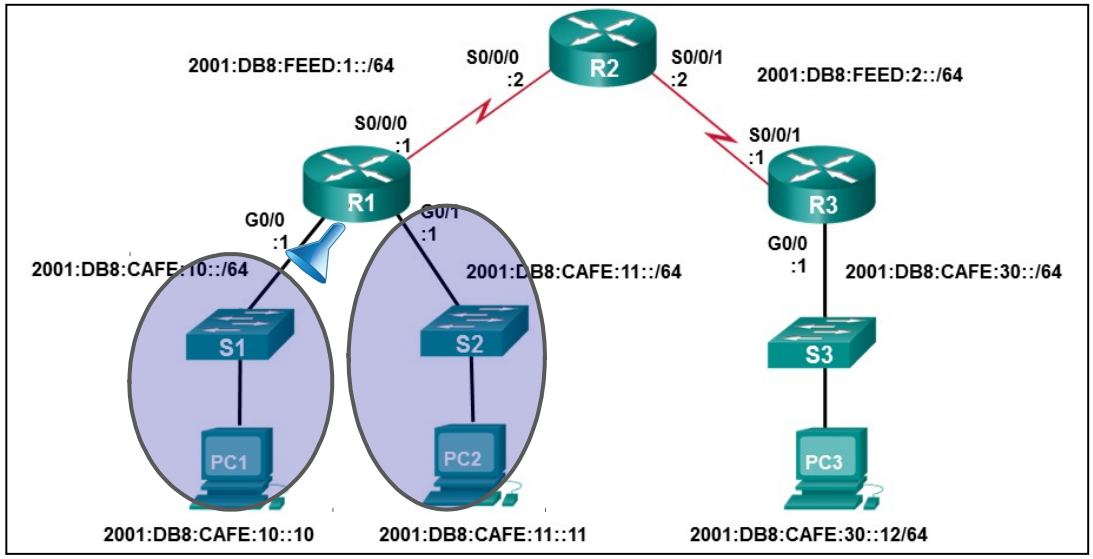



```
R1(config)# ipv6 access-list NO-R3-LAN-ACCESS
R1(config-ipv6-acl)# deny ipv6 2001:db8:cafe:30::/64 any
R1(config-ipv6-acl)# permit ipv6 any any
R1(config-ipv6-acl)# exit
R1(config)#
R1(config)# interface s0/0/0
R1(config-if)# ipv6 traffic-filter NO-R3-LAN-ACCESS in
```

Note: Append the `log` keyword to an ACE to generate logging messages.

IPv6 ACL Example #3

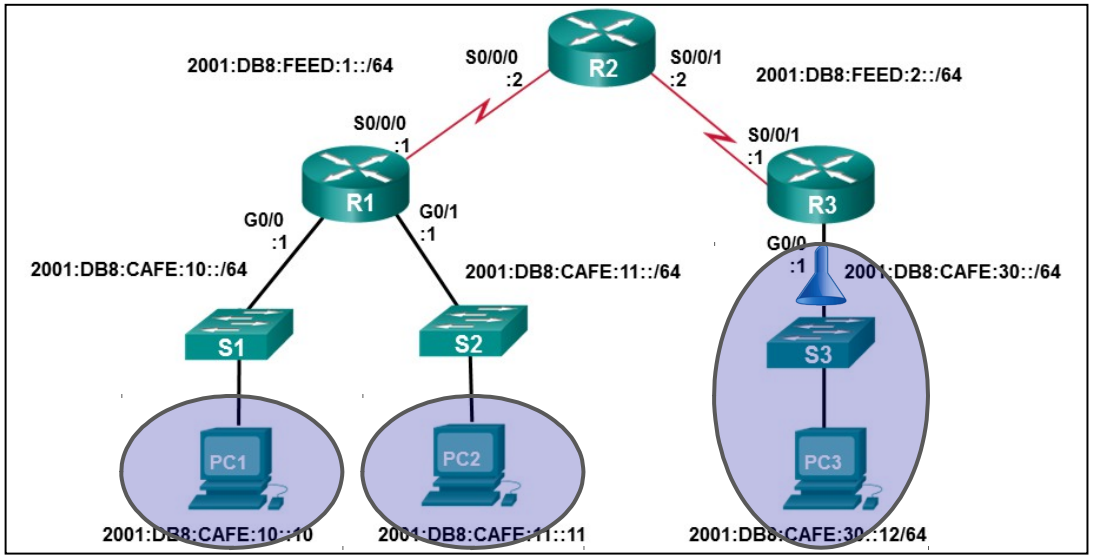
- Configure an incoming IPv6 access list on the R1 G0/0 interface that denies FTP traffic to 2001:DB8:CAFE:11::/64.
  - Both FTP data (port 20) and control (port 21) need to be blocked.
- Because the filter is applied inbound on the G0/0 interface on R1 only traffic from the 2001:DB8:CAFE:10::/64 network will be denied.




```
R1(config)# ipv6 access-list NO-FTP-TO-11
R1(config-ipv6-acl)# deny tcp any
2001:db8:cafe:11::/64 eq ftp
R1(config-ipv6-acl)# deny tcp any
2001:db8:cafe:11::/64 eq ftp-data
R1(config-ipv6-acl)# permit ipv6 any any
R1(config-ipv6-acl)# exit
R1(config)# interface g0/0
R1(config-if)# ipv6 traffic-filter NO-FTP-TO-11 in
```

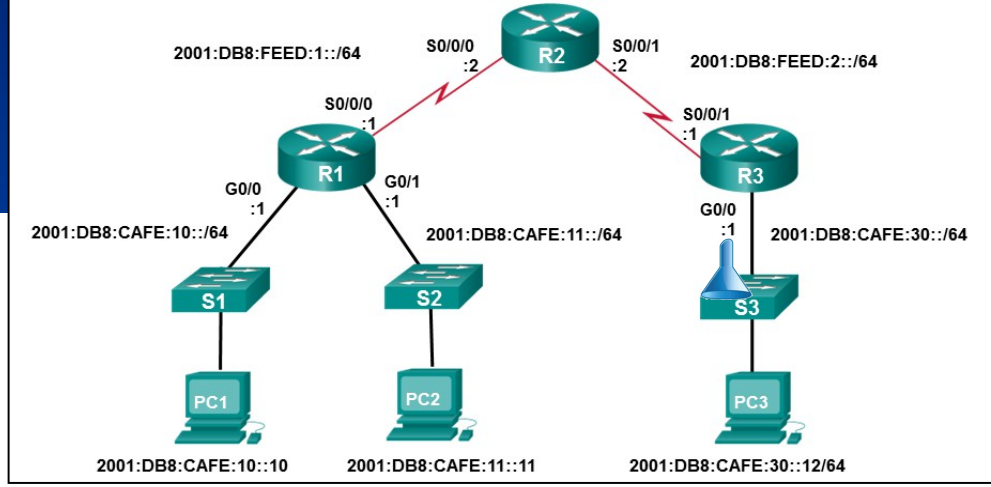
IPv6 ACL Example #4

- Configure an incoming IPv6 access list on the R3 G0/0 interface that:
  - Allows HTTP/HTTPS access from any device to host 2001:DB8:CAFE:10::10.
  - All other devices are denied access to PC1.
  - Allow PC3 Telnet access to PC2.
  - All other devices are denied Telnet access to PC2.
  - All other IPv6 traffic is permitted to all other destinations.
  - Apply the ACL inbound to the R3 G0/0.



IPv6 ACL Example #4

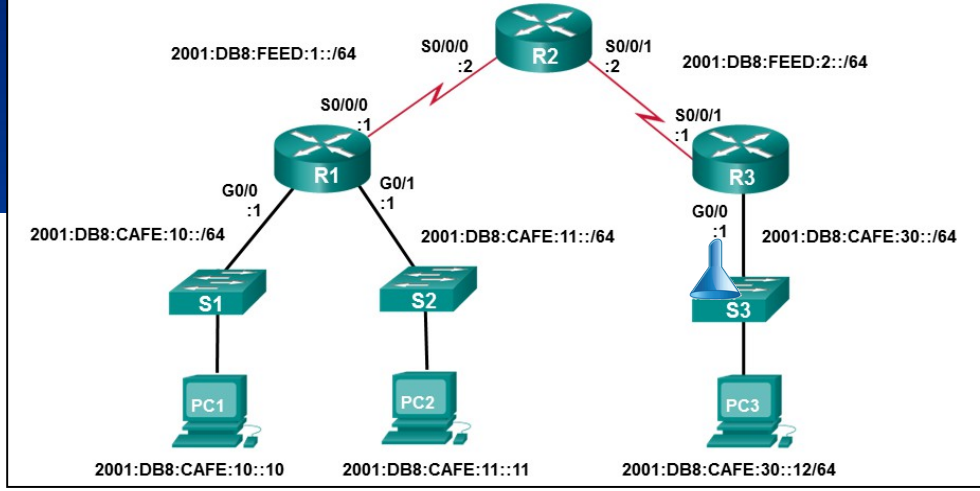
- Configure an incoming IPv6 access list on the R3 G0/0 interface that:
  1. Allows HTTP/HTTPS access from any device to host 2001:DB8:CAFE:10::10.
  2. All other devices are denied access to PC1.
  3. Allow PC3 Telnet access to PC2.
  4. All other devices are denied Telnet access to PC2.
  5. All other IPv6 traffic is permitted to all other destinations.
  6. Apply the ACL inbound to the R3 G0/0.




```
R3(config)# ipv6 access-list RESTRICTED-ACCESS
R3(config-ipv6-acl)# remark Permit access only HTTP and HTTPS to PC1
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 PC1
R3(config-ipv6-acl)# deny ipv6 any 2001:db8:cafe:10::10

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)#
R3(config)# interface g0/0
R3(config-if)# ipv6 traffic-filter RESTRICTED-ACCESS in
```

Verify Example #4

- Configure an incoming IPv6 access list on the R3 G0/0 interface that:
  1. Allows HTTP/HTTPS access from any device to host 2001:DB8:CAFE:10::10.
  2. All other devices are denied access to PC1.
  3. Allow PC3 Telnet access to PC2.
  4. All other devices are denied Telnet access to PC2.
  5. All other IPv6 traffic is permitted to all other destinations.
  6. Apply the ACL inbound to the R3 G0/0.




```
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>

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 23 sequence 70
  deny tcp any host 2001:DB8:CAFE:11::11 eq telnet sequence 90
  permit ipv6 any any sequence 110
```