Setup

- Configure all routers and hosts with the IP addresses above.
- Configure RIP on both routers so that there is full reachability on all networks.
- Verify that all routers and hosts can ping all interfaces.
- Configure SanJose1 and SanJose2 with vty passwords to allow telnets.
- Note: Verify Access List results using the command: `show access-list`
Standard Access Lists

Task 1: Configuring a Standard ACL

Create an standard ACL that will deny all packets from HostB from reaching the 172.16.10.0/24 network.

The pings from HostB to 172.16.10.1 or 172.16.10.20 should fail. However, the pings from SanJose2 should succeed.

SanJose1(config)# access-list 1 deny 172.30.10.20
SanJose1(config)# access-list 1 permit any

How else could we have written the first access list statement?

SanJose1(config)# access-list 1 deny 172.30.10.20 0.0.0.0
or
SanJose1(config)# access-list 1 deny host 172.30.10.20

Standard ACLs are placed as close to the destination as possible. If we put the ACL on SanJose1’s Ethernet 0 interface, it will deny packets for all devices on 172.16.10.0/24 except to 172.16.10.1/24, SanJose1’s Ethernet 0 interface. So we will apply the access list to the serial 0 interface. The down side to this is that this will also deny packets from 172.16.10.0/24 destined for SanJose1’s serial interface 192.168.4.5.

SanJose1(config)# int s 0
SanJose1(config-if)# ip access-group 1 in

HostB

```
C:\>ping 172.16.10.5
Pinging 172.16.10.5 with 32 bytes of data:
Reply from 192.168.4.5: Destination net unreachable.
Reply from 192.168.4.5: Destination net unreachable.
Reply from 192.168.4.5: Destination net unreachable.
Reply from 192.168.4.5: Destination net unreachable.
Ping statistics for 172.16.10.5:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
     Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 172.16.10.1
Pinging 172.16.10.1 with 32 bytes of data:
Reply from 192.168.4.5: Destination net unreachable.
Reply from 192.168.4.5: Destination net unreachable.
Reply from 192.168.4.5: Destination net unreachable.
Reply from 192.168.4.5: Destination net unreachable.
Ping statistics for 172.16.10.1:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
     Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```
Can HostB ping the SanJose1’s Serial 0 interfaces?  **No.**

**HostB**

```
C:\>ping 192.168.4.5
Pinging 192.168.4.5 with 32 bytes of data:
Reply from 192.168.4.5: Destination net unreachable.
Reply from 192.168.4.5: Destination net unreachable.
Reply from 192.168.4.5: Destination net unreachable.
Reply from 192.168.4.5: Destination net unreachable.
```

Can SanJose2 ping HostB?  **Yes, because the source IP address of the packet is 192.168.4.6.**

```
SanJose2# ping 172.16.10.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/32 ms
SanJose2#
```

**Remove all access-list and the access-group statements**

```
SanJose1(config)# no access-list 1
SanJose1(config)# inter s 0
SanJose1(config-if)# no ip access-group 1 in
SanJose1(config-if)#
```
Task 2 – Putting a Standard ACL to close to the source.

What if we placed this same access list but on SanJose1?

```
SanJose2(config)#access-list 1 deny 172.30.10.20
SanJose2(config)#access-list 1 permit any
SanJose2(config)#inter e 0
SanJose2(config-if)#ip access-group 1 in
SanJose2(config-if)#

This will deny HostB from accessing any devices beyond its own LAN including 172.16.10.0/24 and 192.168.4.0/30.
```

Can SanJose2 still ping the SanJose1 and HostA? Yes

```
SanJose2#ping 192.168.4.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
SanJose2#ping 172.16.10.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
SanJose2#
```

Remove all access-list and the access-group statements
Task 3 – Deny an entire network

Deny all devices on 172.30.10.0/24 from reaching 172.16.10.0/24 or the 192.168.4.5 serial interface on SanJose1.

SanJose1(config)#access-list 1 deny 172.30.10.0 0.0.0.255
SanJose1(config)#access-list 1 permit any
SanJose1(config)#

SanJose1(config)#interface 0
SanJose1(config-if)#ip access-group 1 in
SanJose1(config-if)#

This will deny all packets with the source address from the 172.30.10.0/24 network from entering 172.16.10.0/24 or from accessing the 192.168.4.5 serial interface on SanJose1. A better solution would be an extended access list.

HostB

C:\>ping 192.168.4.5
Ping 192.168.4.5 with 32 bytes of data:
Reply from 192.168.4.5: Destination net unreachable.
Reply from 192.168.4.5: Destination net unreachable.
Reply from 192.168.4.5: Destination net unreachable.
Ping statistics for 192.168.4.5:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
     Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\>ping 172.16.10.1
Ping 172.16.10.1 with 32 bytes of data:
Reply from 172.16.10.1: Destination net unreachable.
Reply from 172.16.10.1: Destination net unreachable.
Reply from 172.16.10.1: Destination net unreachable.
Reply from 172.16.10.1: Destination net unreachable.
Ping statistics for 172.16.10.1:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
     Minimum = 0ms, Maximum = 0ms, Average = 0ms
Why does this work?

SanJose2# ping 172.16.10.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
SanJose2#

The pings are originating from SanJose2’s serial interface with an IP address 192.168.4.6, so they are not denied by the access list. However, if we use an extended ping, changing the source IP address to SanJose2’s Ethernet 0 interface, the pings will be denied.

SanJose2# ping
Protocol [ip]:
Target IP address: 172.16.10.5
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.30.10.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.5, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
SanJose2#

Remove all access-list and the access-group statements
Extended Access Lists

Task 1 – Deny all telnets
First verify that you can telnet from both the HostB and SanJose2 to SanJose1.

**HostB**

Deny all telnets sessions coming in on SanJose1’s serial 0 interface. This will deny telnet sessions from both SanJose2 and HostB. Permit all other traffic.

```plaintext
SanJose1(config)#access-list 101 deny tcp any any eq telnet
SanJose1(config)#access-list 101 permit ip any any
SanJose1(config)#
SanJose1(config)#inter s 0
SanJose1(config-if)#ip access-group 101 in
SanJose1(config-if)#
```

Verify that the ACL is denying telnet sessions.

```plaintext
SanJose2#telnet 192.168.4.5
Trying 192.168.4.5 ...
% Destination unreachable; gateway or host down
SanJose2#
```

**HostB**

We will not modify this ACL. It is best just to remove all access-list and the access-group statements.
Add a new access list and access group statements so that it will only deny telnets from HostB and not SanJose2 via Serial 0:

```plaintext
SanJose1(config)#access-list 101 deny tcp 172.30.10.20 0.0.0.0 any eq 23
SanJose1(config)#access-list 101 permit ip any any
SanJose1(config)#
SanJose1(config)#inter s 0
SanJose1(config-if)#ip access-group 101 in
SanJose1(config-if)#
```

Verify that the ACL is denying telnet sessions only from HostB, but not from SanJose2.

**HostB**

```
Microsoft Telnet> open 192.168.4.5
Connecting To 192.168.4.5...Could not open connection to the host, on port 23: Connect Failed
Microsoft Telnet> 
```

```plaintext
SanJose2#telnet 192.168.4.5
Trying 192.168.4.5 ... Open
User Access Verification
Password:
SanJose1>
```

Remove all access-list and the access-group statements.
Task 2 – Deny telnet based on source and destination IP addresses

Deny telnets from HostB to SanJose1’s serial 0 interface, but permit HostB to telnet to SanJose1’s Ethernet 0 interface. Permit all other traffic.

SanJose1(config)#access-list 101 deny tcp host 172.30.10.20 host 192.168.4.5 eq telnet
SanJose1(config)#access-list 101 permit ip any any

SanJose1(config)#int s 0
SanJose1(config-if)#ip access-group 101 in
SanJose1(config-if)#

Verify that HostB cannot telnet to 192.168.4.5 but can telnet to 172.16.0.1

HostB

Microsoft Telnet> open 192.168.4.5
Connecting To 192.168.4.5...Could not open connection to the host, on port 23: Connect failed
Microsoft Telnet> _

HostB

Microsoft Telnet> open 172.16.10.1

Telnet 172.16.10.1

User Access Verification
Password: SanJose1

Remove all access-list and the access-group statements
Named Access Lists

Task 1 – Create a Named Access List with multiple conditions

Create a Named Access List that will:
- Deny ICMP pings from 172.30.10.0/24 network to SanJose1 interfaces and the 172.16.10.0/24 network.
- Permit telnets from 172.30.10.0/24 network to 172.16.10.1, but deny all other telnets from 172.30.10.0/24.
- Permit all other traffic

SanJose1(config)#ip access-list extended Restrict-172.30.10.0/24-Access
SanJose1(config-ext-nacl)#deny icmp 172.30.10.0 0.0.0.255 any echo
SanJose1(config-ext-nacl)#permit tcp 172.30.10.0 0.0.0.255 host 172.16.10.1 eq telnet
SanJose1(config-ext-nacl)#deny tcp 172.30.10.0 0.0.0.255 any eq telnet
SanJose1(config-ext-nacl)#permit ip any any

Does the order of these access list statements matter? Yes. If deny tcp 172.30.10.0 0.0.0.255 any eq telnet came before permit tcp 172.30.10.0 0.0.0.255 host 172.16.10.1 eq telnet all telnets from 172.30.10.0/24 would be denied, before any can be permitted.

SanJose1(config)#int s 0
SanJose1(config-if)#ip access-group Restrict-172.30.10.0/24-Access in
SanJose1(config-if)#
Verify that 172.30.10.0/24 devices, ie.HostB, cannot ping SanJose1 or 172.16.10.0/24.

SanJose2# ping 192.168.4.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
SanJose2# ping 172.16.10.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
SanJose2# ping 172.16.10.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/41/88 ms
SanJose2#

Verify that SanJose2 can still ping all devices.

Remember, this works because the source IP address is not 172.30.10.0/24 but on the 192.168.4.0/30 network.
Try an extended ping from SanJose1. Notice that this fails because the source IP address is part of the 172.30.10.0/24 network.

```
SanJose2# ping
Protocol [ip]: Target IP address: 192.168.4.5
Repeat count [5]: Datagram size [100]:
Timeout in seconds [2]: Extended commands [n]: y
Source address or interface: 172.30.10.1
Type of service [0]: Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]: Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.5, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
SanJose2#
```

Verify that HostB can telnet to 172.16.10.1 but cannot telnet to 192.168.4.5.
Verify that SanJose2 can telnet to all devices.

SanJose2# telnet 192.168.4.5
Trying 192.168.4.5 ... Open

User Access Verification
Password:
SanJose1> exit

[Connection to 192.168.4.5 closed by foreign host]
SanJose2# telnet 172.16.10.1
Trying 172.16.10.1 ... Open

User Access Verification
Password:
SanJose1> exit

[Connection to 172.16.10.1 closed by foreign host]
SanJose2#

Verify that all other traffic is permitted. From HostB show that TCP port 80 traffic is permitted.
Verify Access Lists results.

<table>
<thead>
<tr>
<th>SanJose1# show access-list</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended IP access list Restrict-172.30.10.0/24-Access</td>
</tr>
<tr>
<td>deny icmp 172.30.10.0 0.0.0.255 any echo (32 matches)</td>
</tr>
<tr>
<td>permit tcp 172.30.10.0 0.0.0.255 host 172.16.10.1 eq telnet (42 matches)</td>
</tr>
<tr>
<td>deny tcp 172.30.10.0 0.0.0.255 any eq telnet (6 matches)</td>
</tr>
<tr>
<td>permit ip any any (205 matches)</td>
</tr>
<tr>
<td>SanJose1#</td>
</tr>
</tbody>
</table>