28-1 ACL Configuration - Answer Key

In this lab you will configure standard and extended Access Control Lists.

The routers and PCs have been configured with their network addressing settings, and R2 has a static route for the internal 10.0.1.0/24 and 10.0.2.0/24 networks.

Numbered Standard ACL

1) Verify that all PCs have connectivity to each other, to R1 and to R2.

From PC1, ping PC2, PC3, R1 and R2.

```
C:\>ping 10.0.1.11

Pinging 10.0.1.11 with 32 bytes of data:

Reply from 10.0.1.11: bytes=32 time=1ms TTL=128
Reply from 10.0.1.11: bytes=32 time<1ms TTL=128
Reply from 10.0.1.11: bytes=32 time<1ms TTL=128
Reply from 10.0.1.11: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.1.11:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 10.0.2.10

Pinging 10.0.2.10 with 32 bytes of data:

Request timed out.
Reply from 10.0.2.10: bytes=32 time<1ms TTL=127
Reply from 10.0.2.10: bytes=32 time<1ms TTL=127
Reply from 10.0.2.10: bytes=32 time<1ms TTL=127

Ping statistics for 10.0.2.10:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\>ping 10.0.1.1

Pinging 10.0.1.1 with 32 bytes of data:

Reply from 10.0.1.1: bytes=32 time<1ms TTL=255
Reply from 10.0.1.1: bytes=32 time<1ms TTL=255
Reply from 10.0.1.1: bytes=32 time<1ms TTL=255
Reply from 10.0.1.1: bytes=32 time<1ms TTL=255

Ping statistics for 10.0.1.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

We have already verified connectivity between the PCs in both subnets. Ping R1 and R2 from PC3.

```
C:\>ping 10.0.2.1

Pinging 10.0.2.1 with 32 bytes of data:

Reply from 10.0.2.1: bytes=32 time=1ms TTL=255
Reply from 10.0.2.1: bytes=32 time<1ms TTL=255
Reply from 10.0.2.1: bytes=32 time=1ms TTL=255
Reply from 10.0.2.1: bytes=32 time<1ms TTL=255

Ping statistics for 10.0.2.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 10.0.0.2
```

```
Pinging 10.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time=1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

2) Configure and apply a numbered standard ACL on R1 which denies traffic from all hosts in the 10.0.2.0/24 subnet to R2.

   The PCs in the 10.0.1.0/24 and 10.0.2.0/24 subnets must maintain connectivity to each other.

   The PCs in the 10.0.1.0/24 subnet must maintain connectivity to R2.

The task specifies that a numbered standard ACL be used on R1. This checks the source address only. This prevents us from configuring an ACL inbound on the F0/1 interface which denies traffic from the 10.0.2.0/24 subnet to R2 but permits it to the 10.0.1.0/24 network – that would require an extended ACL.

The only way the task can be completed is by applying the ACL outbound on the F0/0 interface facing R2.

Configure a numbered standard ACL that denies traffic from 10.0.2.0/24. Do not forget to permit from 10.0.1.0/24 as the implicit deny any at the end of the ACL would block the traffic otherwise.

```
R1(config)#access-list 1 deny 10.0.2.0 0.0.0.255
R1(config)#access-list 1 permit 10.0.1.0 0.0.0.255
```

Do not forget to apply the ACL to the interface.

```
R1(config)#interface f0/0
R1(config-if)#ip access-group 1 out
```

3) Test that traffic is secured exactly as required.

   Verify PC1 and PC2 can ping R2.
   PC3 cannot ping R2.
   PC3 can ping PC1 and PC2.

PC1 and PC2 should be able to ping R2.

```
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

PC3 should not be able to ping R2.

```
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.2.1: Destination host unreachable.
Reply from 10.0.2.1: Destination host unreachable.
Reply from 10.0.2.1: Destination host unreachable.
Reply from 10.0.2.1: Destination host unreachable.

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss)
```

PC3 should be able to ping PC1 and PC2.

```
C:\>ping 10.0.1.10

Pinging 10.0.1.10 with 32 bytes of data:

Reply from 10.0.1.10: bytes=32 time<1ms TTL=127
Reply from 10.0.1.10: bytes=32 time<1ms TTL=127
Reply from 10.0.1.10: bytes=32 time<1ms TTL=127
Reply from 10.0.1.10: bytes=32 time=1ms TTL=127

Ping statistics for 10.0.1.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 10.0.1.11

Pinging 10.0.1.11 with 32 bytes of data:

Reply from 10.0.1.11: bytes=32 time<1ms TTL=127
Reply from 10.0.1.11: bytes=32 time<1ms TTL=127
Reply from 10.0.1.11: bytes=32 time=5ms TTL=127
Reply from 10.0.1.11: bytes=32 time=1ms TTL=127

Ping statistics for 10.0.1.11:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 5ms, Average = 1ms
```

Numbered Extended ACL

4) Configure and apply a numbered extended ACL on R1 which permits Telnet access from PC1 to R2. Telnet to R2 must be denied for all other PCs in the network.

   All other connectivity must be maintained.

   Do not change the existing ACL.

   Telnet access has already been enabled on R2. The password is 'Flackbox'

All traffic from the 10.0.2.0/24 subnet to R2 is already denied by the numbered standard ACL we configured.

We need to create an ACL which will allow Telnet traffic to R2 at 10.0.0.2 from PC1 at 10.0.1.10 but deny it from all other hosts in the 10.0.1.0/24 subnet.

The implicit deny all at the end of the ACL would block Telnet traffic from the other hosts to R2, but it would also block all other traffic from the 10.0.1.0/24 subnet including other applications to R2 and traffic to the 10.0.2.0/24 subnet. We need to explicitly block the Telnet traffic but allow other traffic.

```
R1(config)#access-list 100 permit tcp host 10.0.1.10 host
10.0.0.2 eq telnet
R1(config)#access-list 100 deny tcp 10.0.1.0 0.0.0.255 host
10.0.0.2 eq telnet
R1(config)#access-list 100 permit ip any any
```

The access list should be applied as close to the source as possible, so apply it inbound on interface F1/0. We already have an outbound ACL on F0/0 so we could not apply it there anyway.

```
R1(config)#interface f1/0
R1(config-if)#ip access-group 100 in
```

5) Test that traffic is secured exactly as required. Use the command ‘telnet 10.0.0.2’ from the PCs to test and the password ‘Flackbox’. Type ‘exit’ to leave the Telnet session.

   Verify that PC1 can ping and Telnet to R2.
   PC2 can ping R2 but not Telnet to it.
   PC3 cannot ping or Telnet to R2.
   The PCs can all ping each other.

PC1 should be able to Telnet to R2.

```
PC1#telnet 10.0.0.2
Trying 10.0.0.2 ... Open

User Access Verification
Password: Flackbox
R2>
```

PC2 should be able to ping but not Telnet to R2.

```
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>telnet 10.0.0.2
Trying 10.0.0.2 ...
% Connection timed out; remote host not responding
```

PC3 should not be able to ping or Telnet to R2.

```
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.2.1: Destination host unreachable.
Reply from 10.0.2.1: Destination host unreachable.
Reply from 10.0.2.1: Destination host unreachable.
Reply from 10.0.2.1: Destination host unreachable.

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>telnet 10.0.0.2
Trying 10.0.0.2 ...
% Connection timed out; remote host not responding
```

PC3 should be able to ping PC1 and PC2.

```
C:\>ping 10.0.1.10

Pinging 10.0.1.10 with 32 bytes of data:

Reply from 10.0.1.10: bytes=32 time<1ms TTL=127
Reply from 10.0.1.10: bytes=32 time<1ms TTL=127
Reply from 10.0.1.10: bytes=32 time<1ms TTL=127
Reply from 10.0.1.10: bytes=32 time<1ms TTL=127

Ping statistics for 10.0.1.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 10.0.1.11

Pinging 10.0.1.11 with 32 bytes of data:

Reply from 10.0.1.11: bytes=32 time<1ms TTL=127
Reply from 10.0.1.11: bytes=32 time<1ms TTL=127
Reply from 10.0.1.11: bytes=32 time=1ms TTL=127
Reply from 10.0.1.11: bytes=32 time<1ms TTL=127

Ping statistics for 10.0.1.11:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

6) How many Telnet packets were permitted by the ACL?

Enter ‘show access-list’ to check the hit counts. Note that your values may be different.

```
R1#show access-lists 100
Extended IP access list 100
 permit tcp host 10.0.1.10 host 10.0.0.2 eq telnet (23 match(es))
 deny tcp 10.0.1.0 0.0.0.255 host 10.0.0.2 eq telnet (12 match(es))
 permit ip any any (12 match(es))
```

Named Extended ACL

7) Remove the numbered extended ACL you just configured from the interface. Do not delete the ACL.

```
R1(config)#int f1/0
R1(config-if)#no ip access-group 100 in
```

8) Configure and apply a named extended ACL on R1 as follows:

   Permit Telnet from PC1 to R2. Telnet to R2 must be denied for all other PCs in the network.

   Permit ping from PC2 to R2. Ping to R2 must be denied for all other PCs in the network.

   All other connectivity must be maintained.

All traffic from the 10.0.2.0/24 subnet to R2 is already denied by the numbered standard ACL we configured.

We do need to configure an ACL to secure traffic from the 10.0.1.0/24 subnet.

```
R1(config)#ip access-list extended F1/0_in
R1(config-ext-nacl)#permit tcp host 10.0.1.10 host 10.0.0.2
eq telnet
R1(config-ext-nacl)#deny tcp 10.0.1.0 0.0.0.255 host
10.0.0.2 eq telnet
R1(config-ext-nacl)#permit icmp host 10.0.1.11 host
10.0.0.2 echo
R1(config-ext-nacl)#deny icmp 10.0.1.0 0.0.0.255 host
10.0.0.2 echo
R1(config-ext-nacl)#permit ip any any
```

Don’t forget to apply the ACL to the interface.

```
R1(config)#int f1/0
R1(config-if)#ip access-group F1/0_in in
```

9) Test that traffic is secured exactly as required.

   Verify that PC1 cannot ping R2 but can Telnet to it.
   PC2 can ping R2 but cannot Telnet to it.
   PC3 cannot ping or Telnet to R2.
   The PCs can all ping each other.

PC1 cannot ping R2 but can Telnet to it.

```
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.1.1: Destination host unreachable.
Reply from 10.0.1.1: Destination host unreachable.
Reply from 10.0.1.1: Destination host unreachable.
Reply from 10.0.1.1: Destination host unreachable.

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>telnet 10.0.0.2
Trying 10.0.0.2 ...Open

User Access Verification

Password:
R2>
```

PC2 can ping R2 but cannot Telnet to it.

```
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254
Reply from 10.0.0.2: bytes=32 time<1ms TTL=254

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>telnet 10.0.0.2
Trying 10.0.0.2 ...
% Connection timed out; remote host not responding
```

PC3 cannot ping or Telnet to R2.

```
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.2.1: Destination host unreachable.
Reply from 10.0.2.1: Destination host unreachable.
Reply from 10.0.2.1: Destination host unreachable.
Reply from 10.0.2.1: Destination host unreachable.

Ping statistics for 10.0.0.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>telnet 10.0.0.2
Trying 10.0.0.2 ...
% Connection timed out; remote host not responding
```

PC3 should be able to ping PC1 and PC2.

```
C:\>ping 10.0.1.10

Pinging 10.0.1.10 with 32 bytes of data:

Reply from 10.0.1.10: bytes=32 time<1ms TTL=127
Reply from 10.0.1.10: bytes=32 time=1ms TTL=127
Reply from 10.0.1.10: bytes=32 time<1ms TTL=127
Reply from 10.0.1.10: bytes=32 time<1ms TTL=127

Ping statistics for 10.0.1.10:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 10.0.1.11

Pinging 10.0.1.11 with 32 bytes of data:

Reply from 10.0.1.11: bytes=32 time=1ms TTL=127
Reply from 10.0.1.11: bytes=32 time<1ms TTL=127
Reply from 10.0.1.11: bytes=32 time<1ms TTL=127
Reply from 10.0.1.11: bytes=32 time<1ms TTL=127

Ping statistics for 10.0.1.11:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
```