Complex Networks Using Static Routes
Introducing the Topology

- 172.16.0.0/24
- 172.16.1.0/24
- 192.168.1.0/24
- 192.168.2.0/24
- 192.168.3.0/24
- 172.16.1.0/24
Examine Router Interfaces

R1# `show ip route`

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, O - ODR
P - periodic downloaded static route
Gateway of last resort is not set
R1#

- `show ip route` command is used to display the routing table.
- Initially, the routing table is empty if no interfaces have been configured.

- **Note**: Static routes and dynamic routes cannot be added to the routing table until the appropriate local interfaces, also known as the exit interfaces, have been configured on the router. (later)
ip route Command

- The command for configuring a static route is **ip route**.
- The complete syntax for configuring a static route is:

```
ip route prefix mask {ip-address | interface-type interface-number [ip-address]} [dhcp] [distance] [name next-hop-name] [permanent | track number] [tag tag]
```
ip route Command

- Simpler version of the syntax:

```
Router(config)# ip route network-address subnet-mask
{ip-address | exit-interface}
```

- `network-address`: Destination network address of the remote network
- `subnet-mask`: Subnet mask of the remote network

- One or both of the following parameters must also be used:
  - `ip-address`: Next-hop router’s IP address. (Does not have to be next-hop.)
  - `exit-interface`: Outgoing or exit interface
Verifying Static Routes

R1# show running-config
<output omitted>
!
ip route 172.16.1.0 255.255.255.0 172.16.2.2
ip route 192.168.1.0 255.255.255.0 172.16.2.2
ip route 192.168.2.0 255.255.255.0 172.16.2.2
!
<output omitted>
R1# copy running-config startup-config

Now is a good time to save the configuration to NVRAM with the copy running-config startup-config command.
Alex Zinin’ s Routing Table Principles

I know about my remote networks but it is not my responsibility if R2 and R3 know about their remote networks.

Principle 1: Every router makes its decision alone, based on the information it has in its own routing table.

- R1 makes forwarding decisions based solely on the information in the routing table.
- R1 does not consult the routing tables in any other routers.
- Making each router aware of remote networks is the responsibility of the network administrator.
Just because I know how to get to R3’s LAN, 192.168.2.0/24 and I send that packet to R2, doesn’t mean R2 knows how to get there.

Principle 2: The fact that one router has certain information in its routing table does not mean that other routers have the same information.
And if the packet for R3’s LAN reaches 192.168.2.0/24, I don’t know if R3 has a route back to 172.16.3.0/24 for any return traffic.

Principle 3: Routing information about a path from one network to another does not provide routing information about the reverse, or return, path.
R1# show ip route
<output omitted>
    172.16.0.0/24 is subnetted, 3 subnets
S  172.16.1.0 [1/0] via 172.16.2.2
C  172.16.2.0 is directly connected, Serial0/0/0
C  172.16.3.0 is directly connected, FastEthernet0/0
S  192.168.1.0/24 [1/0] via 172.16.2.2
S  192.168.2.0/24 [1/0] via 172.16.2.2

R2# show ip route
<output omitted>
    172.16.0.0/24 is subnetted, 3 subnets
C  172.16.1.0 is directly connected, FastEthernet0/0
C  172.16.2.0 is directly connected, Serial0/0/0
S  172.16.3.0 [1/0] via 172.16.2.1
C  192.168.1.0/24 is directly connected, Serial0/0/1
S  192.168.2.0/24 [1/0] via 192.168.1.1

R3# show ip route
<output omitted>
    172.16.0.0/24 is subnetted, 3 subnets
S  172.16.1.0 [1/0] via 192.168.1.2
S  172.16.2.0 [1/0] via 192.168.1.2
S  172.16.3.0 [1/0] via 192.168.1.2
C  192.168.1.0/24 is directly connected, Serial0/0/1
C  192.168.2.0/24 is directly connected, FastEthernet0/0
Modifying Static Routes (already did this)

There is no way to modify an existing static route.

The static route must be deleted and a new one configured.
Summary and Default Static Routes

- Summary Static Routes
- Default Static Routes
Summarizing Routes to Reduce the Size of the Routing Table

- **Summary route** is a single route that can be used to represent multiple routes.
  - Generally a set of contiguous networks (but do not have to be)
  - Have the same exit interface or next-hop IP address.
- Example:
  - 10.0.0.0/16, 10.1.0.0/16, 10.2.0.0/16, 10.3.0.0/16, 10.4.0.0/16, 10.5.0.0/16, all the way through 10.255.0.0/16
  - Can be represented by a single network address: 10.0.0.0/8.
- Creates smaller routing tables
  - More efficient routing table lookup process more efficient.
- A single static route can be used to represent dozens, hundreds, or even thousands of routes.
  - As of Feb 2010, there are more than 300,000 routes in the Internet core routers.
  - Most of these are summarized routes.
A default route is used to represent all routes with zero or no bits matching.

When there are no routes that have a more specific match, the default route will be a match. (more later)

A default static route is a route that will match all packets.
Configuring a Default Static Route

Router(config)# **ip route 0.0.0.0 0.0.0.0 [exit-interface | ip-address]**

- 0.0.0.0 0.0.0.0 network address and mask is called a **quad-zero route**.
Configuring a Default Static Route

- ip route 172.16.1.0 255.255.255.0 serial 0/0/0
- ip route 192.168.1.0 255.255.255.0 serial 0/0/0
- ip route 192.168.2.0 255.255.255.0 serial 0/0/0

R1 is a stub router and is connected only to R2.
R1 doesn’t need specific routing information to reach R3 networks.
Currently R1 has three static routes.
All three static routes have a(n):
  - Exit interface Serial 0/0/0
  - Next-hop Router R2
Configuring a Default Static Route

| R1(config)# no ip route 172.16.1.0 255.255.255.0 serial 0/0/0 |
| R1(config)# no ip route 192.168.1.0 255.255.255.0 serial 0/0/0 |
| R1(config)# no ip route 192.168.2.0 255.255.255.0 serial 0/0/0 |
| R1(config)# ip route 0.0.0.0 0.0.0.0 serial 0/0/0 |

- Delete individual static routes.
- Configure single default static route.
What will the routing table look like now?

R1# `show ip route`  

**Before**

```
172.16.0.0/24 is subnetted, 3 subnets
S   172.16.1.0 is directly connected, Serial0/0/0
C   172.16.2.0 is directly connected, Serial0/0/0
C   172.16.3.0 is directly connected, FastEthernet0/0
S   192.168.1.0/24 is directly connected, Serial0/0/0
S   192.168.2.0/24 is directly connected, Serial0/0/0
```

R1# `show ip route`  

**After**

```
<some codes omitted>
* - candidate default, U - per-user static route, o - ODR

Gateway of last resort is 0.0.0.0 to network 0.0.0.0
172.16.0.0/24 is subnetted, 2 subnets
C   172.16.2.0 is directly connected, Serial0/0/0
C   172.16.3.0 is directly connected, FastEthernet0/0
S*  0.0.0.0/0 is directly connected, Serial0/0/0
```
Verifying a Default Static Route

R1# show ip route
<some codes omitted>
* - candidate default, U - per-user static route, o - ODR

Gateway of last resort is 0.0.0.0 to network 0.0.0.0
  172.16.0.0/24 is subnetted, 2 subnets
  C     172.16.2.0 is directly connected, Serial0/0/0
  C     172.16.3.0 is directly connected, FastEthernet0/0
S*    0.0.0.0/0 is directly connected, Serial0/0/0

- A /0 mask indicates that zero or no bits are needed to match.
- As long as a more specific match doesn’t exist, the default static route will match all packets.
- Very common
- No need to store hundreds or thousands of routes to different networks.
Troubleshooting a Missing Route

- When end-to-end connectivity is a problem, begin by making sure that you can ping your own interface and other devices on your own directly connected networks.
- When this has been verified, begin testing connectivity to remote networks and from other devices.
- Networks are subject to many different forces that can cause their status to change quite often:
  - Interface failure
  - Dropped connection by a service provider
  - Oversaturation of links
  - Incorrect configuration entered by an administrator.
- Tools:
  - ping
  - traceroute
  - show ip route
  - show ip interface brief
Troubleshooting Static Routes

- Download: cis82-static-routes-troubleshooting.pkt
- Determine why the PCs cannot ping each other and fix the problem.
Troubleshooting Techniques

- Make sure all interfaces are “up” and “up”
- What command will help you determine this?
  - `R1# show ip interface brief`
- What are some of the reasons an interface may be down?
  - Didn’t issue the “no shutdown” command
  - Missing “clock rate” command on Serial DCE interface.
  - Missing or incorrect cable
  - No connection to other end device:
    - Ethernet: No switch or hub attached
    - Serial: Other end not connected or configured
    - Both ends of serial interface must be configured correctly to be “up”
Troubleshooting Techniques

- Can you ping addresses on directly connected networks (next-hop addresses)?
- Can you ping other interfaces on the next-hop router?
- Can you ping the next-hop interface?
- If unsuccessful:
  - Make sure each router on the path to that network have the remote network in its routing table.
  - Make sure each router on the path to that network have a return route back to this router.
Access Control Lists
Routers provide basic traffic filtering capabilities, such as blocking Internet traffic, with access control lists (ACLs).

An ACL is a sequential list of permit or deny statements that apply to addresses or upper-layer protocols.

This module will introduce standard and extended ACLs as a means to control network traffic, and how ACLs are used as part of a security solution.
What is an ACL?

- An ACL is a router configuration script that controls whether a router permits or denies packets to pass based on criteria found in the packet header.
- ACLs are among the most commonly used objects in Cisco IOS software.
- ACLs are also used for selecting types of traffic to be analyzed, forwarded, or processed in other ways.
- As each packet comes through an interface that has an ACL associated with it, the ACL is checked from top to bottom, one line at a time, looking for a criteria that matches the incoming packet.
- The ACL enforces one or more corporate security policies by applying a permit or deny rule to determine the fate of the packet. ACLs can be configured to control access to a network, subnet or host.
ACL Guidelines

- Use ACLs in firewall routers positioned between your internal network and an external network such as the Internet.

- Use ACLs on a router positioned between two parts of your network to control traffic entering or exiting a specific part of your internal network.

- Configure ACLs on border routers - routers situated at the edges of your networks. This provides a very basic buffer from the outside network, or between a less controlled area of your own network and a more sensitive area of your network.

- Configure ACLs for each network protocol configured on the border router interfaces. You can configure ACLs on an interface to filter inbound traffic, outbound traffic, or both.
What is an ACL?
ACL Traffic Filter Rules

With two interfaces and three protocols running, this router could have a total of 12 separate ACLs applied.

The three Ps for using ACLs

You can only have one ACL per protocol, per interface, and per direction:
- One ACL per protocol (e.g., IP or IPX)
- One ACL per interface (e.g., FastEthernet0/0)
- One ACL per direction (i.e., IN or OUT)
Types of ACLs

There are two types of ACLs:

**Standard**: permit or deny packets based on source IP

**Extended**: filter packets based on multiple criteria like source and destination IP, source and destination ports, and more.
Standard ACL Example

Standard ACLs filter IP packets based on the source IP address only!

access-list 10 permit 192.168.30.0 0.0.0.255
Extended ACLs

Extended ACLs are used more often than standard ACLs because they provide a greater range of control.

Like standard ACLs, extended ACLs check the source packet addresses, but they also check the destination address, protocols and port numbers (or services).

This gives a greater range of criteria on which to base the ACL.

For example, an extended ACL can simultaneously allow e-mail traffic from a network to a specific destination while denying file transfers and web browsing.
## Extended ACL Example

### Using port numbers

```plaintext
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 23
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 21
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq 22
```

### Using keywords

```plaintext
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq telnet
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp
access-list 114 permit tcp 192.168.20.0 0.0.0.255 any eq ftp-data
```
Numbering and Naming ACLs

**Numbered ACL:**
You assign a number based on which protocol you want filtered:
- (1 to 99) and (1300 to 1999): Standard IP ACL
- (100 to 199) and (2000 to 2699): Extended IP ACL

**Named ACL:**
You assign a name by providing the name of the ACL:
- Names can contain alphanumeric characters.
- It is suggested that the name be written in CAPITAL LETTERS.
- Names cannot contain spaces or punctuation and must begin with a letter.
- You can add or delete entries within the ACL.
## Command Syntax

### Standard ACL `access-list` Command Syntax

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>access-list-number</code></td>
<td>Number of an ACL. This is a decimal number from 1 to 99, or 1300 to 1999 (for standard ACL).</td>
</tr>
<tr>
<td><code>deny</code></td>
<td>Denies access if the conditions are matched.</td>
</tr>
<tr>
<td><code>permit</code></td>
<td>Permits access if the conditions are matched.</td>
</tr>
<tr>
<td><code>remark</code></td>
<td>Add a remark about entries in an IP access list to make the list easier to understand and scan.</td>
</tr>
<tr>
<td><code>source</code></td>
<td>Number of the network or host from which the packet is being sent. There are two ways to specify the <code>source</code>:</td>
</tr>
<tr>
<td></td>
<td>• Use a 32-bit quantity in four-part, dotted-decimal format.</td>
</tr>
<tr>
<td></td>
<td>• Use the keyword <code>any</code> as an abbreviation for a <code>source</code> and <code>source-wildcard</code> of 0.0.0.0 255.255.255.55.</td>
</tr>
<tr>
<td><code>source-wildcard</code></td>
<td>(Optional) Wildcard bits to be applied to the source. There are two ways to specify the <code>source-wildcard</code>:</td>
</tr>
<tr>
<td></td>
<td>• Use a 32-bit quantity in four-part, dotted-decimal format. Place ones in the bit positions you want to ignore.</td>
</tr>
<tr>
<td></td>
<td>• Use the keyword <code>any</code> as an abbreviation for a <code>source</code> and <code>source-wildcard</code> of 0.0.0.0 255.255.255.55.</td>
</tr>
<tr>
<td><code>log</code></td>
<td>(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the <code>logging console</code> command.)</td>
</tr>
</tbody>
</table>

The message includes the ACL number, whether the packet was permitted or denied, the source address, and the number of packets. The message is generated for the first packet that matches, and then at five-minute intervals, including the number of packets permitted or denied in the prior five-minute interval.
**Wildcard Mask**

- A wildcard mask is a string of binary digits telling a router which part of an IP address to look at.

- A **wildcard mask** is a mask of bits that indicates which parts of an IP address can assume any value.
  - Used by routing protocols to define the networks or subnets to be advertised
  - Used by Access Control Lists (ACLs) to define addresses or address ranges
- It allows to filter out individual addresses, subnets, or ranges of addresses (also allows to filter on odd and even numbers).
### Wildcard Mask Examples

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Decimal</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Address</td>
<td>192.168.1.1</td>
<td>11000000.10101000.00000001 .00000001</td>
</tr>
<tr>
<td>Wildcard Mask</td>
<td>0.0.0.0.</td>
<td>00000000.00000000.00000000.00000000</td>
</tr>
<tr>
<td>Result</td>
<td>192.168.1.1</td>
<td>11000000.10101000.00000001 .00000001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Decimal</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Address</td>
<td>192.168.1.1</td>
<td>11000000.10101000.00000001 .00000001</td>
</tr>
<tr>
<td>Wildcard Mask</td>
<td>255.255.255.255</td>
<td>11111111.11111111.11111111.11111111</td>
</tr>
<tr>
<td>Result</td>
<td>0.0.0.0</td>
<td>00000000.00000000.00000000.00000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Decimal</th>
<th>Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Address</td>
<td>192.168.1.1</td>
<td>11000000.10101000.00000001 .00000001</td>
</tr>
<tr>
<td>Wildcard Mask</td>
<td>0.0.0.255</td>
<td>00000000.00000000.00000000.00000000.11111111</td>
</tr>
<tr>
<td>Result</td>
<td>192.168.1.0</td>
<td>11000000.10101000.00000001 .00000000</td>
</tr>
</tbody>
</table>
The Any and Host Keywords

**Example 1:**
```
R1(config)#access-list 1 permit 0.0.0.0 255.255.255.255
R1(config)#access-list 1 permit any
```

**Example 2:**
```
R1(config)#access-list 1 permit 192.168.10.10 0.0.0.0
R1(config)#access-list 1 permit host 192.168.10.10
```
ACL Placement – Standard ACL

Close to the destination

Where to Place ACLs

Standard ACL 99 to block any traffic from R1
Entering Criteria Statements

Use Notepad or another text editor to create and edit ACLs
Applying ACL

R1(config)# interface S0/0/0
R1(config-if)# ip access-group 103 out
R1(config-if)# ip access-group 104 in
Monitoring and Verifying

```
R1# show access-lists {access-list-number|name}

R1# show access-lists
Standard IP access list SALES
   10 deny 10.1.1.0 0.0.0.255
   20 permit 10.3.3.1
   30 permit 10.4.4.1
   40 permit 10.5.5.1
Extended IP access list ENG
   10 permit tcp host 192.168.10.2 any eq telnet (25 matches)
   20 permit tcp host 192.168.10.2 any eq ftp
   30 permit tcp host 192.168.10.2 any eq ftp-data
```
Examples
Deny a Specific Subnet

R1(config)#access-list 1 deny 192.168.10.10 0.0.0.255
R1(config)#access-list 1 permit 192.168.0.0 0.0.255.255
R1(config)#interface s0/0/0
R1(config-if)#ip access-group 1 out
Deny Telnet

Extended ACL to Deny Only Telnet from Subnet

R1(config)#access-list 101 deny tcp 192.168.11.0 0.0.0.255 any eq 23
R1(config)#access-list 101 permit ip any any

R1(config)#interface Fa0/1
R1(config-if)#ip access-group 101 in
Control VTY Access

Access-class to VTY line
Placing ACLs

The general rule:

- **Standard ACLs** do not specify destination addresses, so they should be placed as close to the destination as possible.

- Put the **extended ACLs** as close as possible to the source of the traffic denied.
Placing ACLs

If the ACLs are placed in the proper location, not only can traffic be filtered, but it can make the whole network more efficient.

If traffic is going to be filtered, the ACL should be placed where it has the greatest impact on increasing efficiency.
Placing ACLs – Standard Example

Standard ACLs do not specify destination addresses, so they should be placed as close to the destination as possible.

If a **standard** ACL is put too close to the source, it will not only deny the intended traffic, but all other traffic to all other networks.
Placing ACLs – Standard Example

Better to use extended access lists, and place them close to the source, as this traffic will travel all the way to RouterD before being denied.

Source
10.0.0.0/8

Destination
172.16.0.0/16

RouterD

interface fastethernet 0/0
  access-group 10 in
  access-list 10 deny 10.0.0.0 0.255.255.255
  access-list 10 permit any
Placing ACLs – Extended Example

- `deny telnet`  
- `deny ftp`  
- `permit any`  

**Source**: 10.0.0.0/8  

**Destination**: 172.16.0.0/16

- Policy is to deny telnet or FTP Router A LAN to Router D LAN.
- All other traffic must be permitted.
- Several approaches can accomplish this policy.
- The recommended approach uses an extended ACL specifying both source and destination addresses.
Placing ACLs – Extended Example

- **deny telnet**
- **deny ftp**
- **permit any**

**Source**
10.0.0.0/8

**Destination**
172.16.0.0/16

```plaintext
interface fastethernet 0/1
    access-group 101 in
access-list 101 deny tcp any 172.16.0.0 0.0.255.255 eq telnet
access-list 101 deny tcp any 172.16.0.0 0.0.255.255 eq ftp
access-list 101 permit ip any any
```

- Place this extended ACL in Router A.
- Then, packets do not cross Router A's Ethernet, do not cross the serial interfaces of Routers B and C, and do not enter Router D.
- Traffic with different source and destination addresses will still be permitted.
Placing ACLs – Extended Example

- deny telnet
- deny ftp
- permit any

Source 10.0.0.0/8
RouterA

Destination 172.16.0.0/16

interface fastethernet 0/1
  access-group 101 in
access-list 101 deny tcp any 172.16.0.0 0.0.255.255 eq telnet
access-list 101 deny tcp any 172.16.0.0 0.0.255.255 eq ftp
access-list 101 permit ip any any

β If the **permit ip any any** is not used, then no traffic is permitted.

β Be sure to **permit ip** and not just tcp or all udp traffic will be denied.
Troubleshooting

Error 1:
Host 192.168.10.10 has no connectivity with 192.168.30.12

# show access-lists 10
10 deny tcp 192.168.10.0 0.0.0.255 any
20 permit tcp 192.168.10.0 0.0.0.255 any eq telnet
30 permit ip any any

192.168.10.10
192.168.30.12
ACL Error Example 2

```
# show access-lists 120
Extended IP access list 120
  10 deny tcp 192.168.10.0 0.0.255.255 any eq telnet
  20 deny tcp 192.168.1.0 0.0.0.255 host 10.100.100.1 eq smtp
  30 permit tcp any any
```

Error 2:
The 192.168.10.0 /24 network cannot use TFTP to connect to the 192.168.30.0 /24.
ACL Error Example 3

```
# show access-lists 130
Extended IP access list 130
  10 deny tcp any eq telnet any
  20 deny tcp 192.168.1.0 0.0.0.255 host 192.168.30.0 eq smtp
  30 permit ip any any
```

Error 3:
192.168.10.0 /24 network can use Telnet to connect to 192.168.30.0 /24, but this connection should not be allowed.
ACL Error Example 4

Error 4:
Host 192.168.10.10 can use Telnet to connect to 192.168.30.12, but this connection should not be allowed.
# ACL Best Practices

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base your ACLs on the security policy of the organization.</td>
<td>This will ensure you implement organizational security guidelines.</td>
</tr>
<tr>
<td>Prepare a description of what you want your ACLs to do.</td>
<td>This will help you avoid inadvertently creating potential access problems.</td>
</tr>
<tr>
<td>Use a text editor to create, edit and save ACLs.</td>
<td>This will help you create a library of reusable ACLs.</td>
</tr>
<tr>
<td>Test your ACLs on a development network before implementing them on a production network.</td>
<td>This will help you avoid costly errors.</td>
</tr>
</tbody>
</table>
Summary

An Access List (ACL) is:
A series of permit and deny statements that are used to filter traffic

Standard ACL
- Identified by numbers 1 - 99 and 1300 - 1999
- Filter traffic based on source IP address

Extended ACL
- Identified by number 100 -199 & 2000 - 2699
- Filter traffic based on
  • Source IP address
  • Destination IP address
  • Protocol
  • Port number
Summary

Implementing ACLs
- 1st create the ACL
- 2nd place the ACL on an interface
  - Standard ACL are placed nearest the destination
  - Extended ACL are placed nearest the source

Use the following commands for verifying & troubleshooting an ACL
- Show access-list
- Show interfaces
- Show run
DMZ
What is a DMZ?

- DMZ stands for demilitarized zone.
- In addition to the standard inside (private) and outside (public) zones, firewalls can have one or more DMZ zones.
- Although it usually involves a firewall, a firewall is not required to create a DMZ. It can be created between two routers using ACL’s.
What is a DMZ?

- A DMZ is a network that is neither inside nor outside the firewall.
- This network can be accessed from inside and outside the firewall, but security rules (ACL’s) will stop traffic in the DMZ from connecting to devices on the inside. A DMZ is less secure than the inside network, but more secure than the outside network.
- Servers in a DMZ should be secured as if they were on the Internet. Firewall rules should permit only the required services to each server.
Simple DMZ topology
DMZ Features

The main benefit of a DMZ is isolation. If a DMZ server gets attacked and is compromised, the attacker will not have access to the users on the inside network. However, the attacker will have access to the other servers in the DMZ as they are on the same network.

The servers can be protected from this by implementing other Cisco security features such as private VLAN’s, port ACL’s, and multiple DMZ’s.
Objective

Create static routes to allow all networks to communicate. Please note the IP addresses in the table below.

Background / Scenario

At the start of this exercise, the laptops can only communicate with devices on the local network. After adding static routes, the laptops can communicate across the routers to the other networks.

Addressing Table

<table>
<thead>
<tr>
<th>Device</th>
<th>Interface</th>
<th>IP Address</th>
<th>Subnet Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUTER-1</td>
<td>Gi0/0</td>
<td>1.1.1.1</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>ROUTER-1</td>
<td>Gi0/2</td>
<td>10.1.1.1</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>ROUTER-2</td>
<td>Gi0/0</td>
<td>2.2.2.1</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>ROUTER-2</td>
<td>Gi0/1</td>
<td>10.1.1.2</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>ROUTER-2</td>
<td>Gi0/2</td>
<td>10.2.2.1</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>ROUTER-3</td>
<td>Gi0/0</td>
<td>3.3.3.1</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>ROUTER-3</td>
<td>Gi0/2</td>
<td>10.2.2.2</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>LAPTOP-1</td>
<td>1.1.1.100</td>
<td>255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>LAPTOP-2</td>
<td>2.2.2.100</td>
<td>255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>LAPTOP-3</td>
<td>3.3.3.100</td>
<td>255.255.255.0</td>
<td></td>
</tr>
</tbody>
</table>

Exercise

Step 1 - Test network connectivity

a. From each laptop, try to ping the other laptops. The pings will fail.

PC>ping 2.2.2.100

Pinging 2.2.2.100 with 32 bytes of data:

Reply from 1.1.1.1: Destination host unreachable.
Reply from 1.1.1.1: Destination host unreachable.
Reply from 1.1.1.1: Destination host unreachable.
Reply from 1.1.1.1: Destination host unreachable.

Ping statistics for 2.2.2.100:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
PC>ping 3.3.3.100
Pinging 3.3.3.100 with 32 bytes of data:
Reply from 1.1.1.1: Destination host unreachable.
Reply from 1.1.1.1: Destination host unreachable.
Reply from 1.1.1.1: Destination host unreachable.
Reply from 1.1.1.1: Destination host unreachable.
Ping statistics for 3.3.3.100:
  Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

b. Check routes on each router using the command "show ip route". Repeat this step for each router.

ROUTER-1>enable
ROUTER-1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
     D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
     i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
     * - candidate default, U - per-user static route, o - ODR
     P - periodic downloaded static route

Gateway of last resort is not set

1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
 C     1.1.1.0/24 is directly connected, GigabitEthernet0/0
 L     1.1.1.1/32 is directly connected, GigabitEthernet0/0

10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
 C     10.1.1.0/24 is directly connected, GigabitEthernet0/2
 L     10.1.1.1/32 is directly connected, GigabitEthernet0/2

Step 2 - Configure static routes in each router, providing route information to the remote networks.

a. On router 1, add a static route for both remote networks.

ROUTER-1(config)#ip route 2.2.2.0 255.255.255.0 10.1.1.2
ROUTER-1(config)#ip route 3.3.3.0 255.255.255.0 10.1.1.2

b. On router 1, verify new static routes.

ROUTER-1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
     D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
     N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
c. On router 2, add a static route for both remote networks.

```
ROUTER-2>enable
ROUTER-2#config t
Enter configuration commands, one per line. End with CNTL/Z.
ROUTER-2(config)#ip route 1.1.1.0 255.255.255.0 10.1.1.1
ROUTER-2(config)#ip route 3.3.3.0 255.255.255.0 10.2.2.2
ROUTER-2(config)#end
```

d. On router 2, verify new static routes.

```
ROUTER-2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

         1.0.0.0/24 is variably subnetted, 2 subnets, 2 masks
             C     1.1.1.0/24 is directly connected, GigabitEthernet0/0
             L     1.1.1.1/32 is directly connected, GigabitEthernet0/0
             2.0.0.0/24 is subnetted, 1 subnets
                     S     2.2.2.0/24 [1/0] via 10.1.1.2
             3.0.0.0/24 is subnetted, 1 subnets
                     S     3.3.3.0/24 [1/0] via 10.1.1.2
             10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
                     C     10.1.1.0/24 is directly connected, GigabitEthernet0/2
                     L     10.1.1.1/32 is directly connected, GigabitEthernet0/2
                     10.2.2.0/24 is directly connected, GigabitEthernet0/2
```
e. On router 3, add a static route for both remote networks.

```
ROUTER-3>enable
ROUTER-3#config t
Enter configuration commands, one per line. End with CNTL/Z.
ROUTER-3(config)#ip route 1.1.1.0 255.255.255.0 10.2.2.1
ROUTER-3(config)#ip route 2.2.2.0 255.255.255.0 10.2.2.1
ROUTER-3(config)#end
```

f. On router 3, verify new static routes.

```
ROUTER-3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

1.0.0.0/24 is subnetted, 1 subnets
 S    1.1.1.0/24 [1/0] via 10.2.2.1

2.0.0.0/24 is subnetted, 1 subnets
 S    2.2.2.0/24 [1/0] via 10.2.2.1

3.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
 C   3.3.3.0/24 is directly connected, GigabitEthernet0/0
 L   3.3.3.1/32 is directly connected, GigabitEthernet0/0
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
 C  10.2.2.0/24 is directly connected, GigabitEthernet0/1
 L  10.2.2.2/32 is directly connected, GigabitEthernet0/1
```

**Step 3 - Test network connectivity**

a. From each laptop, ping the other laptops. The pings will be successful.

```
From LAPTOP-1:

PC>ping 3.3.3.100

Pinging 3.3.3.100 with 32 bytes of data:

Reply from 3.3.3.100: bytes=32 time=0ms TTL=125
Reply from 3.3.3.100: bytes=32 time=0ms TTL=125
Reply from 3.3.3.100: bytes=32 time=0ms TTL=125
Reply from 3.3.3.100: bytes=32 time=0ms TTL=125

Ping statistics for 3.3.3.100:

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

b. From each laptop, trace the route to each remote laptop

PC> **tracert 3.3.3.100**

Tracing route to 3.3.3.100 over a maximum of 30 hops:

```
  1  0 ms   0 ms   0 ms  1.1.1.1
  2  0 ms   0 ms   0 ms  10.1.1.2
  3  0 ms   0 ms   0 ms  10.2.2.2
  4  0 ms   0 ms   0 ms  3.3.3.100
```

Trace complete.
Objectives

Part 1: Configure, Apply and Verify an Extended Numbered ACL
Part 2: Configure, Apply and Verify an Extended Named ACL

Background / Scenario

Two employees need access to services provided by the server. PC1 only needs FTP access while PC2 only needs web access. Both computers are able to ping the server, but not each other.

Addressing Table

<table>
<thead>
<tr>
<th>Device</th>
<th>Interface</th>
<th>IP Address</th>
<th>Subnet Mask</th>
<th>Default Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>G0/0</td>
<td>172.22.34.65</td>
<td>255.255.255.224</td>
<td>N/A</td>
</tr>
<tr>
<td>R1</td>
<td>G0/1</td>
<td>172.22.34.97</td>
<td>255.255.255.240</td>
<td>N/A</td>
</tr>
<tr>
<td>R1</td>
<td>G0/2</td>
<td>172.22.34.1</td>
<td>255.255.255.192</td>
<td>N/A</td>
</tr>
<tr>
<td>Server</td>
<td>NIC</td>
<td>172.22.34.62</td>
<td>255.255.255.192</td>
<td>172.22.34.1</td>
</tr>
<tr>
<td>PC1</td>
<td>NIC</td>
<td>172.22.34.66</td>
<td>255.255.255.224</td>
<td>172.22.34.65</td>
</tr>
<tr>
<td>PC2</td>
<td>NIC</td>
<td>172.22.34.98</td>
<td>255.255.255.240</td>
<td>172.22.34.97</td>
</tr>
</tbody>
</table>

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Step 1: Configure an ACL to permit FTP and ICMP.

a. From global configuration mode on R1, enter the following command to determine the first valid number for an extended access list.

```
R1(config)# access-list ?
<1-99> IP standard access list
<100-199> IP extended access list
```

b. Add 100 to the command, followed by a question mark.

```
R1(config)# access-list 100 ?
deny Specify packets to reject
permit Specify packets to forward
remark Access list entry comment
```

c. To permit FTP traffic, enter permit, followed by a question mark.

```
R1(config)# access-list 100 permit ?
ahp Authentication Header Protocol
eigrp Cisco's EIGRP routing protocol
esp Encapsulation Security Payload
gre Cisco's GRE tunneling
icmp Internet Control Message Protocol
ip Any Internet Protocol
ospf OSPF routing protocol
```
Packet Tracer - Configuring Extended ACLs

tcp Transmission Control Protocol
udp User Datagram Protocol

d. This ACL permits FTP and ICMP. ICMP is listed above, but FTP is not, because FTP uses TCP. So you enter TCP. Enter tcp to further refine the ACL help.

R1(config)# access-list 100 permit tcp ?
   A.B.C.D Source address
   any Any source host
   host A single source host

e. Notice that we could filter just for PC1 by using the host keyword or we could allow any host. In this case, any device is allowed that has an address belonging to the 172.22.34.64/27 network. Enter the network address, followed by a question mark.

R1(config)# access-list 100 permit tcp 172.22.34.64 ?
   A.B.C.D Source wildcard bits

f. Calculate the wildcard mask determining the binary opposite of a subnet mask.

   11111111.11111111.11111111.1111111111111111 = 255.255.255.224
   00000000.00000000.00000000.00000000.00011111 = 0.0.0.31

g. Enter the wildcard mask, followed by a question mark.

R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 ?
   A.B.C.D Destination address
   any Any destination host
   eq Match only packets on a given port number
   gt Match only packets with a greater port number
   host A single destination host
   lt Match only packets with a lower port number
   neq Match only packets not on a given port number
   range Match only packets in the range of port numbers

h. Configure the destination address. In this scenario, we are filtering traffic for a single destination, the server. Enter the host keyword followed by the server’s IP address.

R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 ?
   dscp Match packets with given dscp value
   eq Match only packets on a given port number
   established established
   gt Match only packets with a greater port number
   lt Match only packets with a lower port number
   neq Match only packets not on a given port number
   precedence Match packets with given precedence value
   range Match only packets in the range of port numbers
   <cr>

i. Notice that one of the options is <cr> (carriage return). In other words, you can press Enter and the statement would permit all TCP traffic. However, we are only permitting FTP traffic; therefore, enter the eq keyword, followed by a question mark to display the available options. Then, enter ftp and press Enter.
Packet Tracer - Configuring Extended ACLs

R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 eq ?
   <0-65535> Port number
   ftp File Transfer Protocol (21)
   pop3 Post Office Protocol v3 (110)
   smtp Simple Mail Transport Protocol (25)
   telnet Telnet (23)
   www World Wide Web (HTTP, 80)
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 eq ftp

j. Create a second access list statement to permit ICMP (ping, etc.) traffic from PC1 to Server. Note that the access list number remains the same and a specific type of ICMP traffic does not need to be specified.
R1(config)# access-list 100 permit icmp 172.22.34.64 0.0.0.31 host 172.22.34.62

k. All other traffic is denied, by default.

Step 2: Apply the ACL on the correct interface to filter traffic.

From R1’s perspective, the traffic that ACL 100 applies to is inbound from the network connected to Gigabit Ethernet 0/0 interface. Enter interface configuration mode and apply the ACL.
R1(config)# interface gigabitEthernet 0/0
R1(config-if)# ip access-group 100 in

Step 3: Verify the ACL implementation.

a. Ping from PC1 to Server. If the pings are unsuccessful, verify the IP addresses before continuing.
b. FTP from PC1 to Server. The username and password are both cisco.
   PC> ftp 172.22.34.62
c. Exit the FTP service of the Server.
   ftp> quit
d. Ping from PC1 to PC2. The destination host should be unreachable, because the traffic was not explicitly permitted.

Part 2: Configure, Apply and Verify an Extended Named ACL

Step 1: Configure an ACL to permit HTTP access and ICMP.

a. Named ACLs start with the ip keyword. From global configuration mode of R1, enter the following command, followed by a question mark.
   R1(config)# ip access-list ?
   extended Extended Access List
   standard Standard Access List

b. You can configure named standard and extended ACLs. This access list filters both source and destination IP addresses; therefore, it must be extended. Enter HTTP_ONLY as the name. (For Packet Tracer scoring, the name is case-sensitive.)
   R1(config)# ip access-list extended HTTP_ONLY
c. The prompt changes. You are now in extended named ACL configuration mode. All devices on the PC2 LAN need TCP access. Enter the network address, followed by a question mark.

R1(config-ext-nacl)# permit tcp 172.22.34.96 ?
A.B.C.D Source wildcard bits

d. An alternative way to calculate a wildcard is to subtract the subnet mask from 255.255.255.255.

\[
\begin{align*}
255.255.255.255 & - 255.255.255.240 \\
\hline
\end{align*}
\]

= 0. 0. 0. 15

R1(config-ext-nacl)# permit tcp 172.22.34.96 0.0.0.15 ?

e. Finish the statement by specifying the server address as you did in Part 1 and filtering www traffic.

R1(config-ext-nacl)# permit tcp 172.22.34.96 0.0.0.15 host 172.22.34.62 eq www

f. Create a second access list statement to permit ICMP (ping, etc.) traffic from PC2 to Server. Note: The prompt remains the same and a specific type of ICMP traffic does not need to be specified.

R1(config-ext-nacl)# permit icmp 172.22.34.96 0.0.0.15 host 172.22.34.62

R1(config-ext-nacl)#

g. All other traffic is denied, by default. Exit out of extended named ACL configuration mode.

Step 2: Apply the ACL on the correct interface to filter traffic.

From R1’s perspective, the traffic that access list HTTP_ONLY applies to is inbound from the network connected to Gigabit Ethernet 0/1 interface. Enter the interface configuration mode and apply the ACL.

R1(config)# interface gigabitEthernet 0/1
R1(config-if)# ip access-group HTTP_ONLY in

Step 3: Verify the ACL implementation.

a. Ping from PC2 to Server. If the pings unsuccessful, verify the IP addresses before continuing.

b. FTP from PC2 to Server. The connection should fail.

c. Open the web browser on PC2 and enter the IP address of Server as the URL. The connection should be successful.
Packet Tracer – DMZ Configuration

Objective
Configure, apply and verify ACL’s to protect the DMZ and Inside networks. Please note the ACL numbers in the table below.

Background / Scenario
We need to allow HTTP and HTTPS traffic to the DMZ webserver for Internet users. The DMZ server needs to connect to the FTP server on the Inside network for data backups.

In this exercise, you need to restrict the DMZ server to only allow HTTP/HTTPS traffic from the Internet and only allow FTP and ICMP to the inside network. If the webserver comes under attack and gets compromised, the hackers will not have unrestricted access into the private network. This provides some protection for the web server and also for the private network.

Addressing Table

<table>
<thead>
<tr>
<th>Device</th>
<th>Interface</th>
<th>ACL Number</th>
<th>IP Address</th>
<th>Subnet Mask</th>
<th>Default Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>G0/0</td>
<td>100</td>
<td>172.16.1.2</td>
<td>255.255.255.252</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>G0/1</td>
<td>101</td>
<td>192.168.1.1</td>
<td>255.255.255.0</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>G0/2</td>
<td>102</td>
<td>10.1.1.1</td>
<td>255.255.255.0</td>
<td>N/A</td>
</tr>
<tr>
<td>Server – DMZ</td>
<td>NIC</td>
<td></td>
<td>192.168.1.2</td>
<td>255.255.255.0</td>
<td>192.168.1.1</td>
</tr>
<tr>
<td>PC1</td>
<td>NIC</td>
<td></td>
<td>10.1.1.2</td>
<td>255.255.255.0</td>
<td>10.1.1.1</td>
</tr>
<tr>
<td>Server – Inside</td>
<td>NIC</td>
<td></td>
<td>10.1.1.100</td>
<td>255.255.255.0</td>
<td>10.1.1.1</td>
</tr>
</tbody>
</table>

Exercise

Step 1: Configure an ACL to permit HTTP and HTTPS from the Internet to the DMZ server. No traffic should be allowed to the Inside network.

a. From global configuration mode on R1, create ACL 100 to allow only HTTP and HTTPS to the DMZ server.
   
   R1(config)# access-list 100 permit tcp any host 192.168.1.2 eq www
   
   R1(config)# access-list 100 permit tcp any host 192.168.1.2 eq 443

Step 2: Apply the ACL to the Outside interface.

a. From global configuration mode on R1, enter interface configuration mode.
   
   R1(config)# interface GigabitEthernet0/0

b. From interface configuration mode, apply ACL 100 to traffic entering the interface.
   
   R1(config-if)# ip access-group 100 in
Step 3: Configure an ACL for the DMZ network. This ACL should permit HTTP and HTTPS reply traffic to the Internet and also permit FTP and ICMP traffic to the internal network.

a. From global configuration mode on R1, create ACL 101 to allow only HTTP and HTTPS reply traffic to the Internet.

```
R1(config)# access-list 101 permit tcp host 192.168.1.2 eq www any
R1(config)# access-list 101 permit tcp host 192.168.1.2 eq 443 any
```

b. From global configuration mode on R1, append ACL 101 to allow FTP to the inside FTP server.

```
R1(config)# access-list 101 permit tcp host 192.168.1.2 host 10.1.1.100 eq ftp
R1(config)# access-list 101 permit tcp host 192.168.1.2 eq ftp host 10.1.1.100
```

c. From global configuration mode on R1, append ACL 101 to permit ICMP traffic to the inside network and deny all other traffic to the inside network.

```
R1(config)# access-list 101 permit icmp host 192.168.1.2 10.1.1.0 0.0.0.255
R1(config)# access-list 101 deny ip host 192.168.1.2 10.1.1.0 0.0.0.255
R1(config)# access-list 101 permit ip any any
```

Step 4: Apply the ACL to the DMZ interface.

a. From global configuration mode on R1, enter interface configuration mode.

```
R1(config)# interface GigabitEthernet0/1
```

b. From interface configuration mode, apply ACL 101 to traffic entering the interface.

```
R1(config-if)# ip access-group 101 in
```

Step 5: Configure an ACL for the Inside network. This ACL should permit FTP and ICMP traffic for the DMZ and deny all other traffic to the DMZ. All traffic should be permitted from the Inside to the Internet.

a. From global configuration mode on R1, create ACL 102 to allow FTP to the DMZ server.

```
R1(config)# access-list 102 permit tcp host host 10.1.1.100 192.168.1.2 eq ftp
R1(config)# access-list 102 permit tcp host 10.1.1.100 eq ftp host 192.168.1.2
```

b. From global configuration mode on R1, append ACL 102 to allow ICMP form the inside network to the DMZ server.

```
R1(config)# access-list 102 permit icmp 10.1.1.0 0.0.0.255 host 192.168.1.2
R1(config)# access-list 102 deny ip 10.1.1.0 0.0.0.255 host 192.168.1.2
R1(config)# access-list 102 permit ip any any
```

Step 6: Apply the ACL to the Inside interface.
a. From global configuration mode on R1, enter interface configuration mode.
   
   ```
   R1(config)# interface GigabitEthernet0/2
   ```

b. From interface configuration mode, apply ACL 102 to traffic entering the interface.
   
   ```
   R1(config-if)# ip access-group 102 in
   ```

---

**Step 7: Verify the ACL implementation.**

a. Ping from PC1 to the DMZ Server and in the reverse direction.

b. Ping from the inside FTP server to the DMZ server and in the reverse direction.

c. FTP from PC1 to the DMZ Server. The username/password is cisco/cisco.

d. FTP from DMZ server to the inside server. The username/password is cisco/cisco.

e. Open the web browser from the Internet PC. Browse to http://192.18.1.2.