NETWOK SEGMENTATION PROJECT – VLAN AND ACL

- VLAN Theory
- ACL Theory
- Applying ACL to Vlan Interface
- Project

VLAN THEORY

VLAN is a group of devices on one or more LAN that are configured to communicate as if they are attached to the same wire, when in fact they are in different LAN segments physically. Because Vlan is based on logic instead of physical connection, they are extremely flexible.
A Broadcast Domain is the set of all devices that will receive broadcast frames originating from any device within the set. Layer 2 switches create Broadcast Domain based on the configuration of the switch. Each of the Broadcast Domain is like a distinct virtual bridge within a switch hence devices in a particular VLAN are said to be in their own Broadcast Domain.
Traffic cannot pass directly to another VLAN within the switch or between two switches unless configured to do so with a Layer 3 device (Layer 3 switch or Router). VLAN are often associated with IP subnetworks. All devices in a particular IP subnetwork belong to the same VLAN traffic. Traffic between VLAN must be routed. You must assign LAN interface VLAN membership on an interface-by-interface basis.
You can set the following parameters when you create a VLAN in the management domain

- VLAN NUMBER - giving each vlan their number like “VLAN 10”
- VLAN NAME - giving each vlan their name for easy remembrance of their functionality
- VLAN TYPE - whether they are normal vlan (communication through packet exchange), or they are management or protocol
- MAXIMUM TRANSMISSION uNIT (MTU) - for vlan, this is the size of data that can travel through the vlan
- SECURITY ASSOCIATION IDENTIFIER - secure password to make sure only authorize device can use the vlan
- VLAN number to use when translating from one media Vlan type to another

VLAN RANGES
0 and 4095 are for system use only. You cannot see or use these VLANS
1 - Cisco default. You can use this VLAN but you cannot delete it.
2 - 1001 - used Ethernet VLANS, you can create use and delete these VLANS
1002 to 1005 - CISCO default for FDDI AND Token Ring
1006 - 4095 - for extended only.
More about Vlan and their configuration can be accessed here.

ACCESS LIST CONTROL (ACL)

Access Control List is an ordered list of rules used to filter traffic. A rule is referred to as Access Control Entry. Each entry states what's permitted and what's **denied.** When a packet attempts to enter or leave a router, it is tested against each rule in the list from first to last. If the packet matches a rule, its outcome is determined by the condition of the statement. If the first rule the packet match is a permit rule, its permitted and if it is a deny statement, the packet is denied.

Now technically speaking, when a packet is sent out, it must know its source and destination through IP addressing. The router looks at this information to determine if it matches any entry in the ACL. If a Router can’t find any match between the information in an ACL and the information in the packet that is attempting to enter, the packet is denIed **IMPLICITLY**.

IMPLICIT DENY

The last rule in every ACL is an Implicit Deny Statement. Because it’s implicit, you won't see it. Be aware because just because you don't see it doesn't mean its doesn't do anything. This rule is very powerful. Every bit of traffic that doesn’t match a rule in ACL will be denied.

Every ACL has an hidden rule at the end that says “Deny Everything” if a packet doesn’t match any of the specific rules you've written in the ACL, it will be blocked (Denied). It's like a security guard saying “if your name is not on the guest list, you can’t come in”. Because of this Implicit Deny, you must explicitly allow all the traffic you want.

As I have mentioned earlier, ACL filters packets based on information like source/destination IP address, protocol, and even ports. They are used for Security, Policy Enforcement and Traffic Control. There are two types of ACL mainly: Standard and Extended.

Standard ACL

Standard ACLs are limited to controlling traffic based on the source IP information as opposed to the source and destination IP address information. You use standard ACL when you want to block traffic from a specific network or device regardless of the type of traffic. Complete distrust or trust is one of the attributes of Standard ACL because you can't single out a specific host device. The numbers range from 1-99.

Extended ACL

ACL is evolving and hence the introduction of Extended ACL. Extended ACL on the other hand filters traffic based on source and destination IP and port and protocol. This is used when you need precise control, like “Allow web traffic (http) from 192.168.1.1 to 10.0.0.0”. The numbers range from 100-199 and 2000-2699. Extended ACLs are more in use nowadays because there is nothing Standard ACL can do that Extended can't.

Inbound and Outbound Traffic.

We mentioned that packets are tested against a set of entries (Rules) that state either Permit or Deny and this may leave you wondering what the packet is permitted or denied to do. That depends on where the ACL is applied- Inbound or Outbound.

Inbound Traffic

When a traffic is said to In-bound, the traffic arrives at the interface from the public internet trying to get into the internal network. Example is if you apply an ACL to a router interface **FastEthernet0/0** as **Inbound**, it checks the packet as they arrive at the interface.

Outbound Traffic

These on the hand are traffic trying to leave the interface, through the router and going off to the internet. Example is if you apply an ACL to a router interface **FastEthernet0/0** as **Outbound,** it checks the traffic as they leave the interface.

So in the context of security, Inbound ACL prevents malicious traffic from entering the network while Outbound ACL prevents sensitive data from going out.

Only one packet filter can be applied as a filter ACL and you can only have one per direction and protocol.

Syntax to Apply ACL to Interface (IPv4)

```
Interface <Interface>
Ip access-group <ID> <In | out>

Interface <Interface>
Ipv6 traffic-filter <ID> <In | Out>
```

ACL Best Practice

Since Standard ACL only filters traffic based on source address, it is advisable to place the ACL closer to the destination. Extended ACL can be anywhere and still function well but it is best placed closer to the source in order to drop the packet as early as possible.

Top-Down Matching and Order of Operation

ACLs processes packets using a Top-down matching approach. This means that ACL, like I have been saying, is a list of rules (Access Control Entries or ACE) processed in order, from Top to Bottom. The packet checks itself against each rule in sequence until it finds a match. Once it finds a match, the router applies the action <permit or deny> and stops checking further rules. If no match is found, the Implicit Deny at the end is applied.

The order of rules matters a lot because a more specific rule like **Deny 192.168.1.100** should come before general rules like **Permit 192.168.1.0/24**

Numbered ACL

ACL can be identified by a number. For example, 10 for Standard and 100 for Extended. This might be harder to manage in a larger network because numbers aren't descriptive.

Syntax for Numbered ACL
SACL
access-list #ID action <Permit | Deny> source-addr

EACL
access-list #ID action <Permit | Deny> protocol source addr | port destination addr | port

Named ACL

Like I mentioned earlier, ACL is evolving. Like Extended ACL is a better version of Standard ACL, the Named ACL is to better Numbered ACL as well. These ACLs are identified by a definitive name like **BLOCK_HTTP.** Named ACLs have additional features that make them better than Standard ACL by all standards. For example, if your matching order does not tally, the only way to fix in a number ACL is to delete and create a new set of entries but the **Sequence** syntax in the named ACL can fix this among other things named ACL can fix. Some of these are

- Allow you to remove individual lines
- Allow you insert lines at the desired location using the sequence number
- Modified already constructed numbered ACL as long as you modify them using named sequence
- Allow you to renumber (resequence) ACL sequence number
- It can also allow you to configure IPv6 ACLs. Numbered ACL can’t do this

Syntax for Named ACL
SACL
Ip access-list Standard <ID or Name>
[sequence] <action> <source>

EACL
Ip access-list Extended <ID or Name>
[sequence]   <action>   <protocol> <source add | port> <Destination add| port>

Ipv6
Ipv6 access-list   <ID or Name>
[Sequence <#>] <action>   <protocol>   <source>   <Destination>

Wild Card Mask

ACL uses a wild card mask to specify ranges of IP addresses. They are like the opposite of a subnet mask. Subnet mask specifies which part of an IP belongs to the network and which one belongs to the host. For example 255.255.255.0 for a /24 network. Wild card on the other hand specifies which part of an IP to match <0> and which part to ignore <1>.

0 = means to match this perfectly.
1 = ignore this part (any value is okay).
Example - Ip 192.168.1.0 has a wildcard mask of 0.0.0.255

Common Wildcard Mask
0.0.0.0 - matches any single Ip exactly (192.168.1.100)
255.255.255.255 - matches any Ip
0.0.0.255 - match a /24 subnet.

APPLYING ACL TO VLAN INTERFACE

After understanding both concepts, applying ACLs to VLAN interfaces can be considered a crucial network security and control, allowing you to filter traffic within and between VLANs. It provides granular control over which traffic is allowed or denied, improving network security and potentially improving performance.

Key Concepts to Conciser When trying to Apply ACL to Vlan Interface

Vlan Interface
This is a virtual interface on switches that represent a Vlan <Interface Vlan 10>. You apply ACL to the Vlan the same way you apply them to a physical interface. Use Inbound ACL to control the traffic entering the interface and use Outbound ACL to control the traffic leaving the Vlan Interface.

Example

Consider a topology of a switch with 2 Vlan - Vlan 10 (192.168.10.2 | 192.168.10.5) and Vlan 20 (192.168.20.10).
The goal is to allow Vlan 10 PCs to access the Server HTTP service on port 80 but deny any other traffic using the implicit deny (The invisible rule at the end of every ACLs).

The first step would be Configuring the Vlan: Naming and Port Assignment.
The name would stay **Vlan 10** and **Vlan 20.**
The port assignment would take this command

- ➔ Interface fa0/1
- ➔ switchport mode access
- ➔ switchport access Vlan 10

Now a Vlan Interface in the context of a Vlan is to assign a virtual interface that represents all the devices in a Vlan no matter the amount of devices in the Vlan (Like a default gateway). Often takes that first IP in a subnet, at least that's what I use. Consider these bash

- ➔ Interface Vlan 10
- ➔ Ip add 192.168.10.1 255.255.255.0
- ➔ no shutdown <this activates the interface

And to allow inter-vlan routing, assuming this is L3 switch, the command to consider will be

- ➔ Ip routing

The next step after creating the Vlan would be to create an ACL and the scenario we have up there describes an extended ACL. Creating a named extended ACL <VLAN_SERVER>

- ➔ Ip access-list extended VLAN_SERVER
- ➔ permit tcp 192.168.10.0   0.0.0.255 host 192.168.20.10 eq 80
- ➔ Exit

And the final step would be to apply this just created ACL to the Vlan interface. We will consider the Inbound direction, trying to allow the http traffic from the server to enter the Vlan interface.

- ➔ Interface Vlan 10
- ➔ Ip access-group VLAN_SERVER In
- ➔ exit

To confirm if this works, we need to test this by trying to access the server via the browser and this should work. And any other service the icmp echo pinging request should fail due to implicit deny.

Explanation

The ACL allow tcp port 80 (HTTP) traffic from VLAN 10 (192.168.10.0) to the server (192.168.20.10). All other traffic <ICMP,FTP> is blocked by the implicit deny. The ACL then apply the In-bound in the Vlan 10 SVI (Switched Virtual Interface), so it checks the packet as they enter the Vlan.

Troubleshooting ACL

Some of the command to consider when trying to troubleshoot ACL include

- ➔ Show access-list : shows all the ACLs and their rules including their Interface count (how many packets matched each rule).
- ➔ Show running config | include access-list : show ACL in the configuration.
- ➔ Show ip interface <Interface> : show which ACLs are applied to an interface and in which direction.
- ➔ Show Vlan brief : verifies Vlans and port assignment (switch).

Project

Tools

Cisco Packet Tracer for Simulation

Controlling Traffic within a Small Office with Multiple Vlan

Objective

You are a network admin for a growing office. You need to configure a network with 5 Vlans for different departments Including a Server department, and apply a named extended ACL to enforce the following security policies

- Allow HR-PC1 TO access the web server on HTTP (port 80) and File server on FTP (port 21).
- Allow IT-PCs to ping any of the servers.
- Allow Guest-PC1 to access the web server on HTTPs (443) only.
- Deny Guest-PC1 from accessing any other services or servers.
- Allow Management-PCs to access both servers on SSH (port 22) for admin purpose
- Deny all inter-vlan traffic for security.
- Deny any other Vlan from accessing Management Vlan
- Allow Management-PCs to ping any of the servers
- Allow HR-PC1 to ping the File Server
- Deny II-PC2 from pinging the File Server

Topology

- Router (Cisco 2911, Router0)
- L2 Switch (2960-24TT, Switch0)
- PCs - 2 for each departments (HR, IT, Guest, Servers, and Management)
- Servers - 2 Servers (WEB and File Server)

Vlan and Subnets

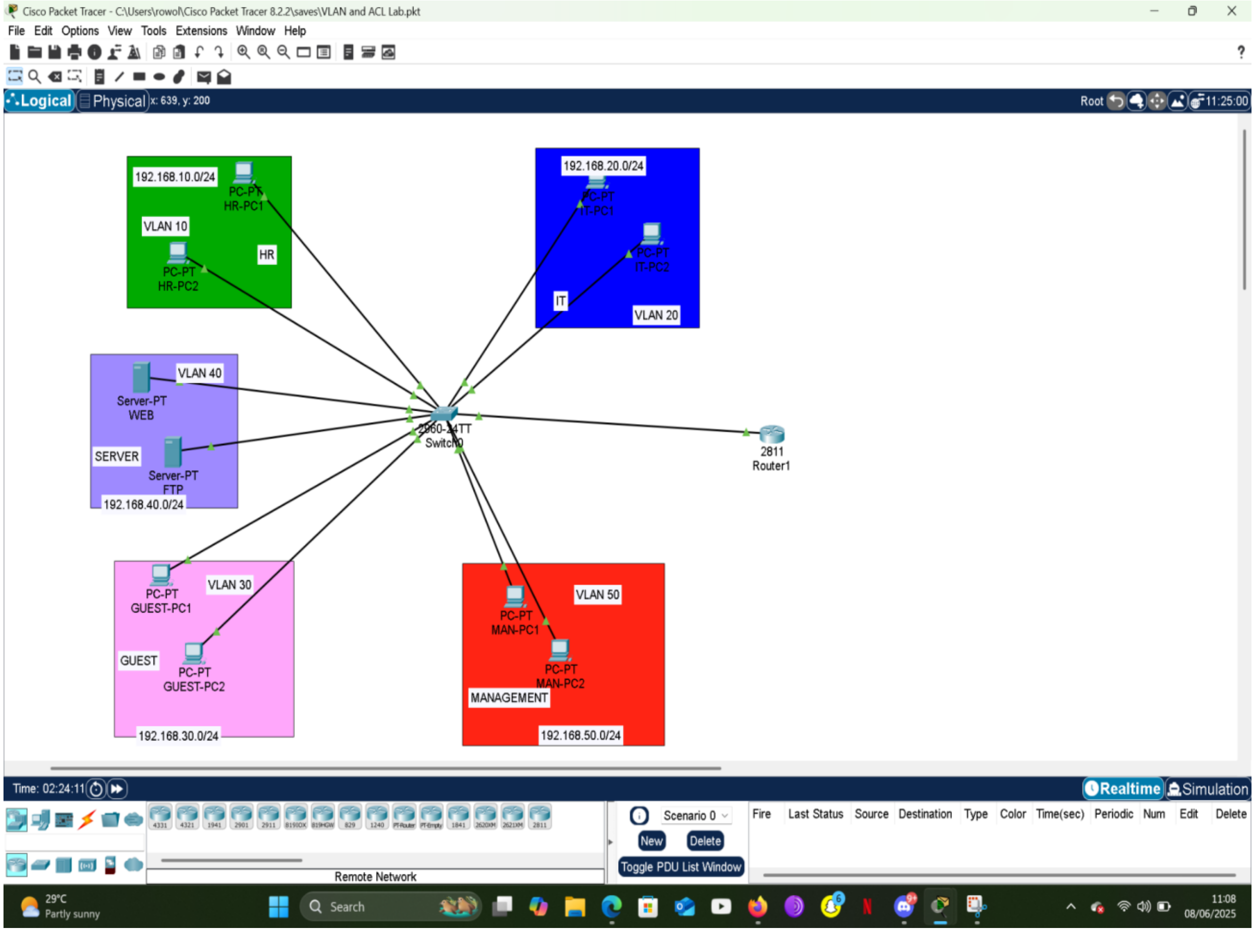
- Vlan 10: HR (192.168.10.0/24)
- Vlan 20: IT (192.168.20.0/24)
- Vlan 30: Guest (192.168.30.0/24)
- Vlan 40: Servers (192.168.40.0/24)
- Vlan 50: Management (192.168.500.0/24)

Vlan Configuration

Topology

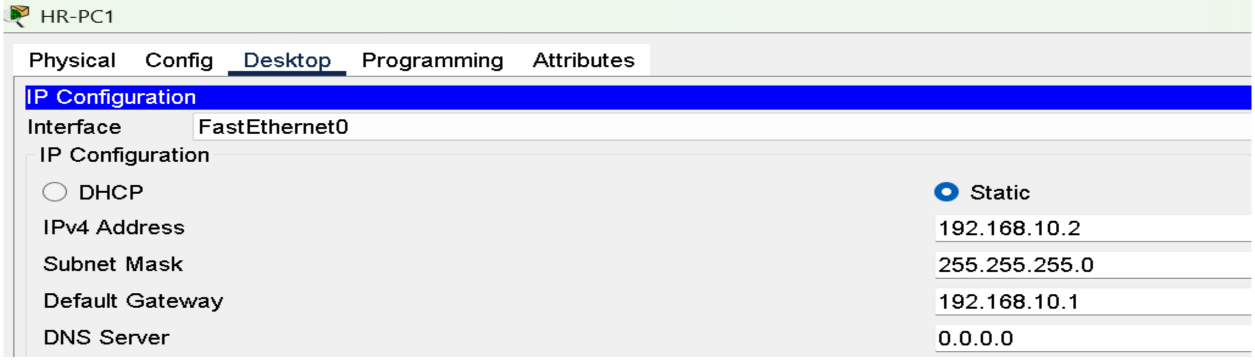
Generally speaking, Topology is the arrangement of devices, connection and network in a computer system, defining how data flows between components.

In the context of our simulation, it refers to the layout of your router, switch, pcs and servers as well as Vlan, Interconnected to manage traffic with ACL.



This is what my topology looks like after laying out all the devices we will use in this simulation.

I went on to assign IP addresses to each device. This is a means devices on networks uses to identify and communicate with each other. It is a must for any device connected to the internet to have these addresses.



HR-PC1

Physical Config Desktop Programming Attributes

IP Configuration

Interface FastEthernet0

IP Configuration

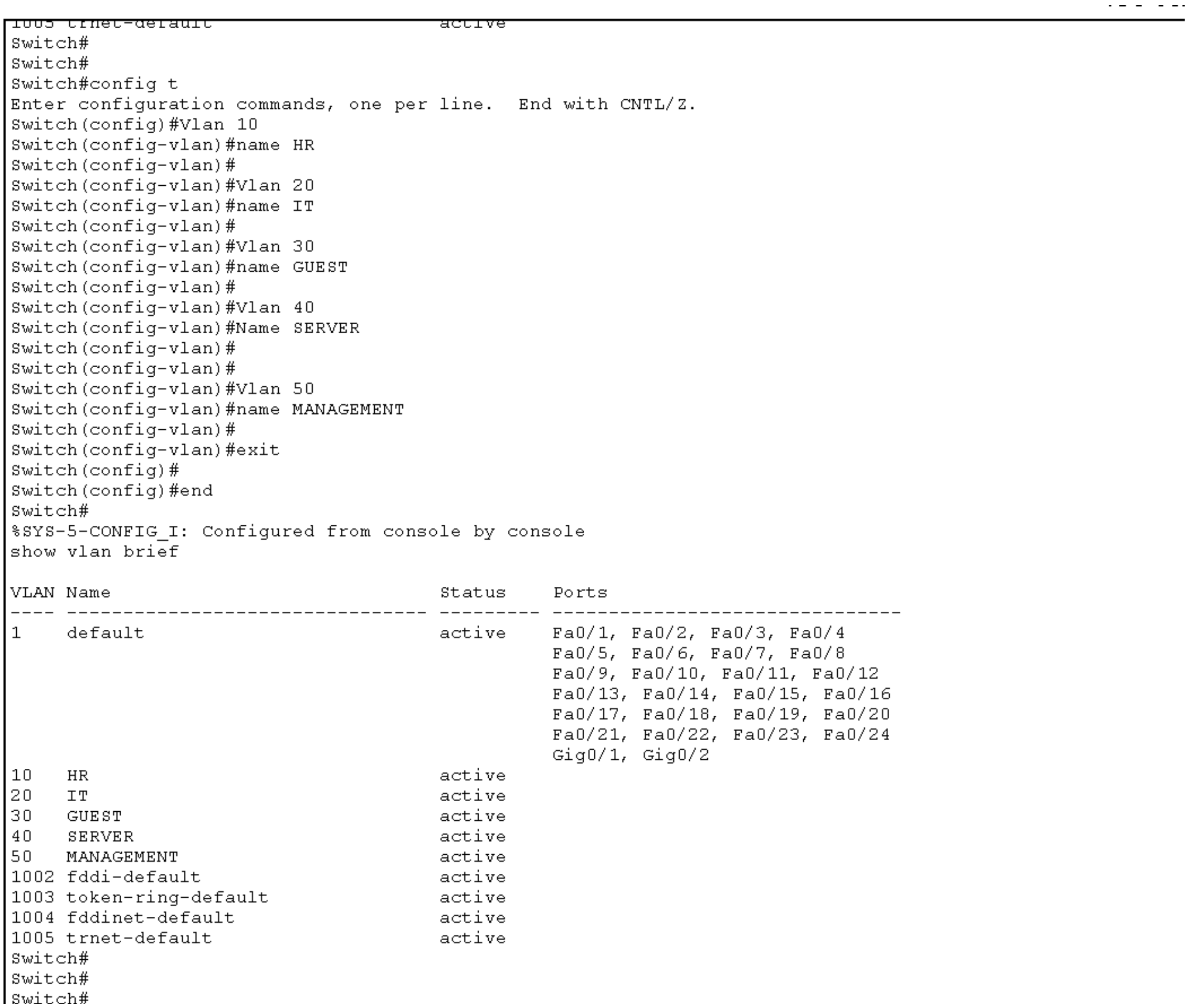
○ DHCP ● Static

IPv4 Address	192.168.10.2
Subnet Mask	255.255.255.0
Default Gateway	192.168.10.1
DNS Server	0.0.0.0

Vlan Configuration

Vlan configuration happens on the switch in the CLI. The first thing is to create each Vlan from 10 to 50 in our context and name them. Consider these commands:

- ➔ Vlan 10
- ➔ name HR
- ➔ Vlan 20
- ➔ name IT
- ➔ Vlan 30
- ➔ name Guest
- ➔ Vlan 40
- ➔ name Server
- ➔ Vlan 50
- ➔ name Management
- ➔ exit



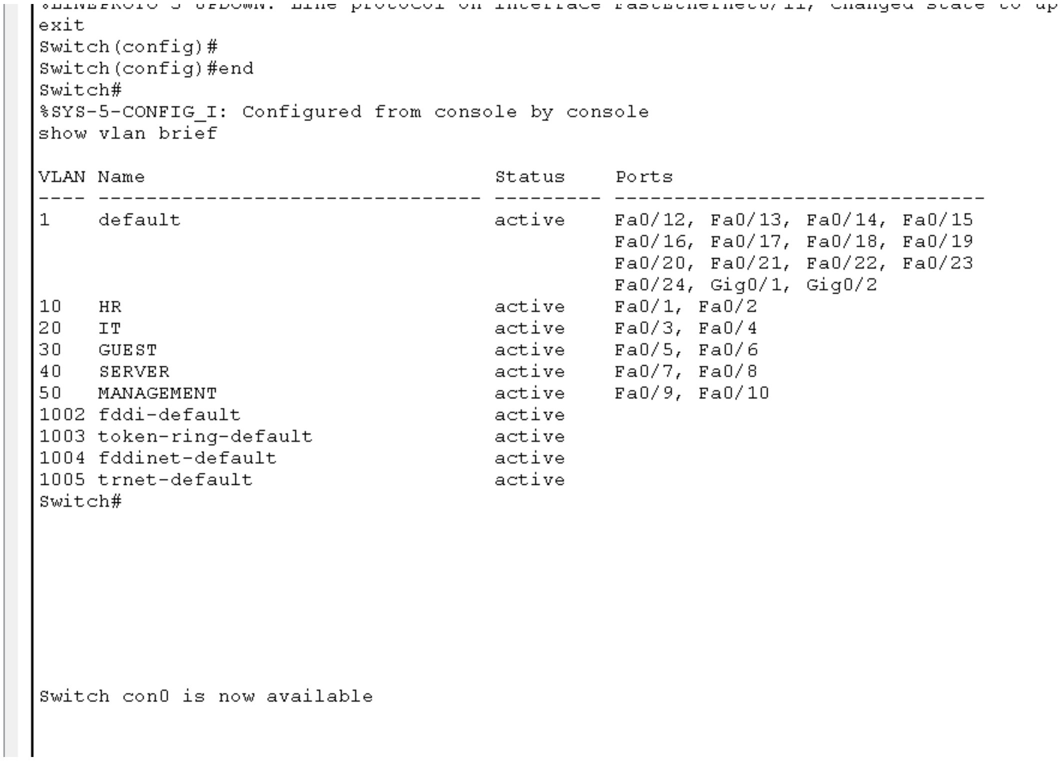
```
1005 trnet-default                    active
Switch#
Switch#
Switch#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)#Vlan 10
Switch(config-vlan)#name HR
Switch(config-vlan)#
Switch(config-vlan)#Vlan 20
Switch(config-vlan)#name IT
Switch(config-vlan)#
Switch(config-vlan)#Vlan 30
Switch(config-vlan)#name GUEST
Switch(config-vlan)#
Switch(config-vlan)#Vlan 40
Switch(config-vlan)#Name SERVER
Switch(config-vlan)#
Switch(config-vlan)#
Switch(config-vlan)#Vlan 50
Switch(config-vlan)#name MANAGEMENT
Switch(config-vlan)#
Switch(config-vlan)#exit
Switch(config)#
Switch(config)#end
Switch#
%SYS-5-CONFIG_I: Configured from console by console
show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                                Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                                Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                                Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                                Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                                Fa0/21, Fa0/22, Fa0/23, Fa0/24
                                                Gig0/1, Gig0/2
10   HR                               active
20   IT                               active
30   GUEST                            active
40   SERVER                           active
50   MANAGEMENT                       active
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
Switch#
Switch#
Switch#
```

The next thing to do under Vlan config is to assign each port to their respective Vlan. Until this is done, the devices are still on the same broadcast domain meaning they have not really been allocated to different Vlan. These commands can be considered when trying to assign ports to Vlan

➔ Interface fa0/1
➔ switchport mode access
➔ switchport access Vlan 10
➔
➔ Interface fa0/2
➔ switchport mode access
➔ switchport access Vlan 10

These commands basically assign the 2 ports which the HR-PCs are connected to Vlan 10. The same process is repeated for the rest of the PCs and Servers. After that, a Trunking config is done to aid inter-vlan routing. This is done with command:

➔ Interface fa0/11 <The router is connected to this port on the switch>
➔ switchport mode Trunk
➔ Exit.



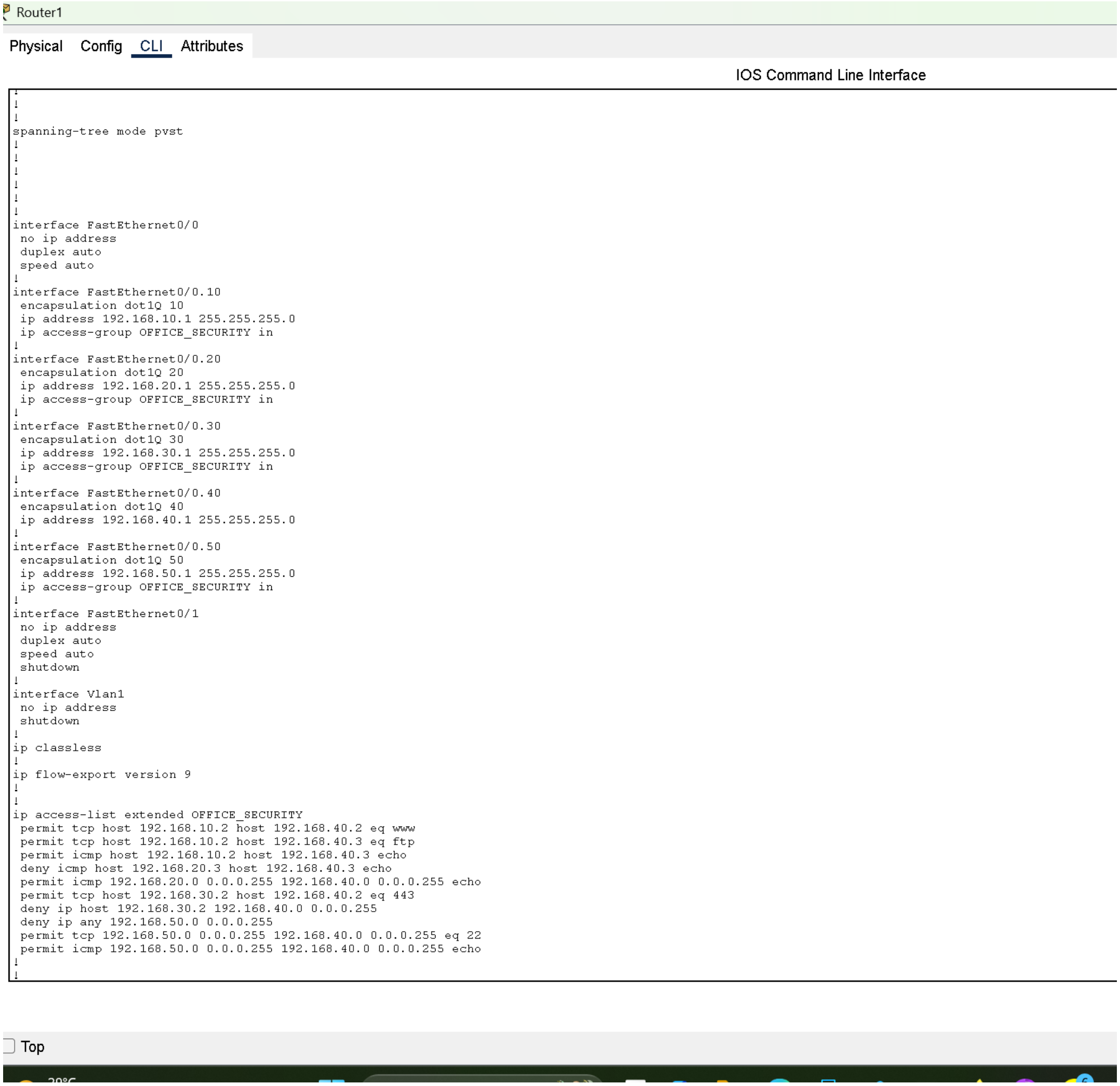
```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to up
exit
Switch(config)#
Switch(config)#end
Switch#
%SYS-5-CONFIG_I: Configured from console by console
show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/12, Fa0/13, Fa0/14, Fa0/15
                                                Fa0/16, Fa0/17, Fa0/18, Fa0/19
                                                Fa0/20, Fa0/21, Fa0/22, Fa0/23
                                                Fa0/24, Gig0/1, Gig0/2
10   HR                               active    Fa0/1, Fa0/2
20   IT                               active    Fa0/3, Fa0/4
30   GUEST                            active    Fa0/5, Fa0/6
40   SERVER                           active    Fa0/7, Fa0/8
50   MANAGEMENT                       active    Fa0/9, Fa0/10
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
Switch#







Switch con0 is now available
```

This shows the following ports were successfully assigned to their respective Vlan.

The next step is to go to the router to finalize the Inter-vlan router we already started on the switch by configuring each sub-interface. This is done on the router because a L3 device is needed for routing traffic between networks, even a virtual one. This is also important as this will serve as the Interfaces where ACLs will be applied (SVI). We proceed to the CLI and use these commands:

- ➔ Interface fa0/0.10 <this command forces us into the sud-interface for Vlan 10>
- ➔ encapsulation dot1q 10 <a standard for trunking>
- ➔ Ip add 192.168.10.1 255.255.255.0 <a form of gateway for Vlan 10, this was added to each device when i was configuring Ip as well>
- ➔ exit

This is repeated for the remaining Vlans as well. It should look like this after you are done:



```
!
!
!
spanning-tree mode pvst
!
!
!
!
!
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/0.10
 encapsulation dot1Q 10
 ip address 192.168.10.1 255.255.255.0
 ip access-group OFFICE_SECURITY in
!
interface FastEthernet0/0.20
 encapsulation dot1Q 20
 ip address 192.168.20.1 255.255.255.0
 ip access-group OFFICE_SECURITY in
!
interface FastEthernet0/0.30
 encapsulation dot1Q 30
 ip address 192.168.30.1 255.255.255.0
 ip access-group OFFICE_SECURITY in
!
interface FastEthernet0/0.40
 encapsulation dot1Q 40
 ip address 192.168.40.1 255.255.255.0
!
interface FastEthernet0/0.50
 encapsulation dot1Q 50
 ip address 192.168.50.1 255.255.255.0
 ip access-group OFFICE_SECURITY in
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
ip classless
!
ip flow-export version 9
!
!
ip access-list extended OFFICE_SECURITY
 permit tcp host 192.168.10.2 host 192.168.40.2 eq www
 permit tcp host 192.168.10.2 host 192.168.40.3 eq ftp
 permit icmp host 192.168.10.2 host 192.168.40.3 echo
 deny icmp host 192.168.20.3 host 192.168.40.3 echo
 permit icmp 192.168.20.0 0.0.0.255 192.168.40.0 0.0.0.255 echo
 permit tcp host 192.168.30.2 host 192.168.40.2 eq 443
 deny ip host 192.168.30.2 192.168.40.0 0.0.0.255
 deny ip any 192.168.50.0 0.0.0.255
 permit tcp 192.168.50.0 0.0.0.255 192.168.40.0 0.0.0.255 eq 22
 permit icmp 192.168.50.0 0.0.0.255 192.168.40.0 0.0.0.255 echo
!
!
```

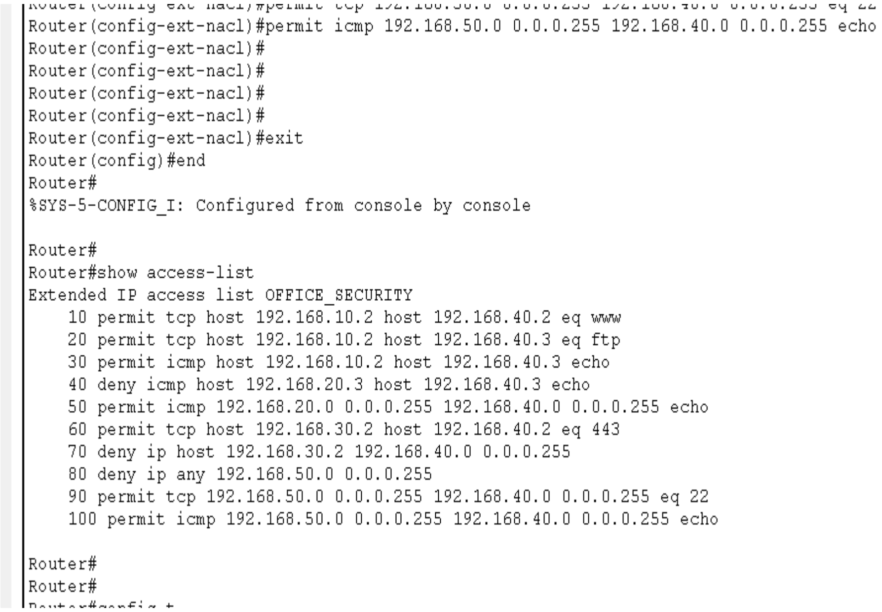
This marks the end of the Vlan config for this lab. The next thing will be to configure the ACL and apply them to our virtual interface,

ACL Configuration

The first of course is to create our Access Control List. These entries will/should take the Top-Down matching order meaning it should be arranged in a way that a more specific rule comes first before a general rule. The actual command for these are:

- Ip access-list extended OFFICE_SECURITY
- permit tcp host 192.168.10.2 host 192.168.40.2 eq 80
- permit tcp host 192.168.10.2 host 192.168.40.3 eq 21
- permit icmp host 192.168.10.2 host 192.168.40.3 echo
- deny icmp host 192.168.20.3 host 192.168.40.3 echo
- permit icmp 192.168.20.0 0.0.0.255 192.168.40.0 0.0.0.255 echo
- permit tcp host 192.168.30.2 host 192.168.40.2 eq 443
- deny ip host 192.168.30.2 192.168.40.0 0.0.0.255
- deny ip any 192.168.50.0 0.0.0.255
- permit tcp 192.168.50.0 0.0.0.255 192.168.40.0 0.0.0.255 eq 22
- permit icmp 192.168.50.0 0.0.0.255 192.168.40.0 0.0.0.255 echo
- exit

They should look like this after using the command **show access-list**

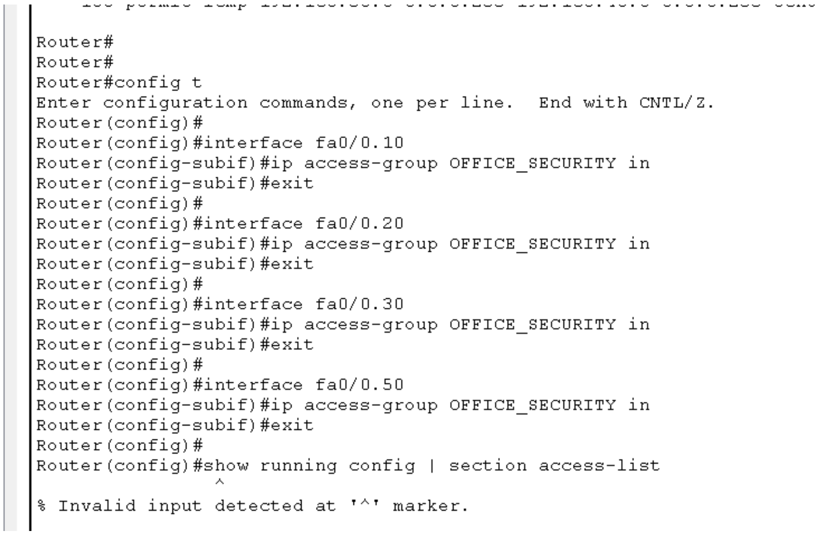


```
Router(config-ext-nacl)#permit tcp 192.168.50.0 0.0.0.255 192.168.40.0 0.0.0.255 eq 22
Router(config-ext-nacl)#permit icmp 192.168.50.0 0.0.0.255 192.168.40.0 0.0.0.255 echo
Router(config-ext-nacl)#
Router(config-ext-nacl)#
Router(config-ext-nacl)#
Router(config-ext-nacl)#
Router(config-ext-nacl)#exit
Router(config)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#
Router#show access-list
Extended IP access list OFFICE_SECURITY
    10 permit tcp host 192.168.10.2 host 192.168.40.2 eq www
    20 permit tcp host 192.168.10.2 host 192.168.40.3 eq ftp
    30 permit icmp host 192.168.10.2 host 192.168.40.3 echo
    40 deny icmp host 192.168.20.3 host 192.168.40.3 echo
    50 permit icmp 192.168.20.0 0.0.0.255 192.168.40.0 0.0.0.255 echo
    60 permit tcp host 192.168.30.2 host 192.168.40.2 eq 443
    70 deny ip host 192.168.30.2 192.168.40.0 0.0.0.255
    80 deny ip any 192.168.50.0 0.0.0.255
    90 permit tcp 192.168.50.0 0.0.0.255 192.168.40.0 0.0.0.255 eq 22
    100 permit icmp 192.168.50.0 0.0.0.255 192.168.40.0 0.0.0.255 echo

Router#
Router#
Router#config t
```

The number **10 - 100** in front of every entry is called the **Sequence Number.** It is only seen in an Extended ACL and it can come in handy when trying to perform tasks like increase a sequence or add rules in between the list without deleting all the list and recreate another one.

The next thing to do would be applying the just created rules to an Interface or interfaces to make sure the rules are enforced. Like I've said earlier, it is applied on the SVI and this is done with these command:

- ➔ Interface fa0/0.10
- ➔ Ip access-group OFFICE_SECURITY In
- ➔ exit
- ➔
- ➔ Interface fa0/0.20
- ➔ Ip access-group OFFICE_SECURITY In
- ➔ exit
- ➔
- ➔ Interface fa0/0.30
- ➔ Ip access-group OFFICE_SECURITY In
- ➔ exit
- ➔
- ➔ Interface fa0/0.50
- ➔ Ip access-group OFFICE_SECURITY In
- ➔ exit



```
Router#
Router#
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#interface fa0/0.10
Router(config-subif)#ip access-group OFFICE_SECURITY in
Router(config-subif)#exit
Router(config)#
Router(config)#interface fa0/0.20
Router(config-subif)#ip access-group OFFICE_SECURITY in
Router(config-subif)#exit
Router(config)#
Router(config)#interface fa0/0.30
Router(config-subif)#ip access-group OFFICE_SECURITY in
Router(config-subif)#exit
Router(config)#
Router(config)#interface fa0/0.50
Router(config-subif)#ip access-group OFFICE_SECURITY in
Router(config-subif)#exit
Router(config)#
Router(config)#show running config | section access-list
                ^
% Invalid input detected at '^' marker.
```

These are applied In-bound on each Vlan sub-interface to filter traffic from HR, IT, Guest and Management. We didn't mention servers since we are controlling traffic at the source so no ACL is applied to Vlan 40 as it is mostly the destination when considering **In-bound** in our context. However the servers will be configured to perform the services they have been tagged with (WEB) and (FILE).

Servers Configuration

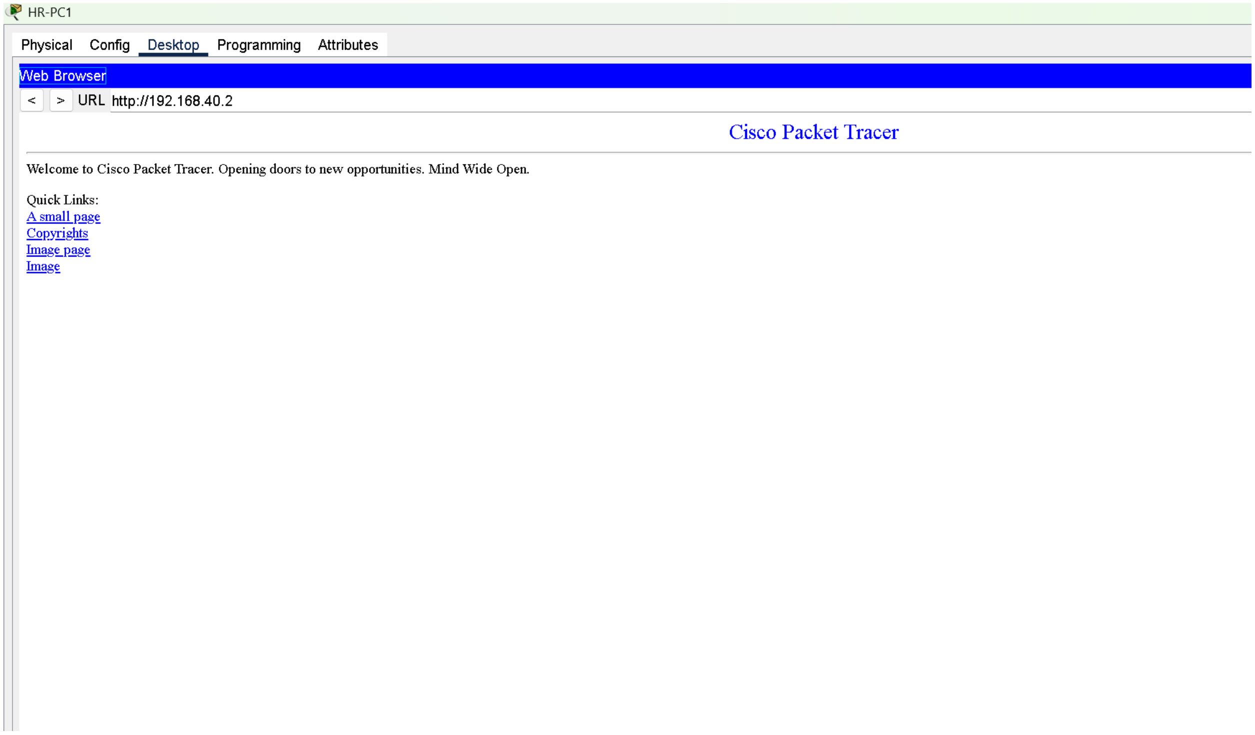
This can be enabled in Cisco Packet Tracer in the **Service** tab on the servers and this has to be done for the sake of testing.

- ➔ For the web server (192.168.40.2), we go to **Service** and enable **HTTP** and **HTTPS.**
- ➔ For File Server (192.168.40.3), we go to **Service** and enable **FTP.**

Testing Logs

To ensure they have been successfully enforced, we can try accessing the permitted or denied service to see if the actions are carried out perfectly.

- From HR-PC1 **192.168.10.2**, access the **web browser** and try loading http://192.168.40.2. This should connect to the web server



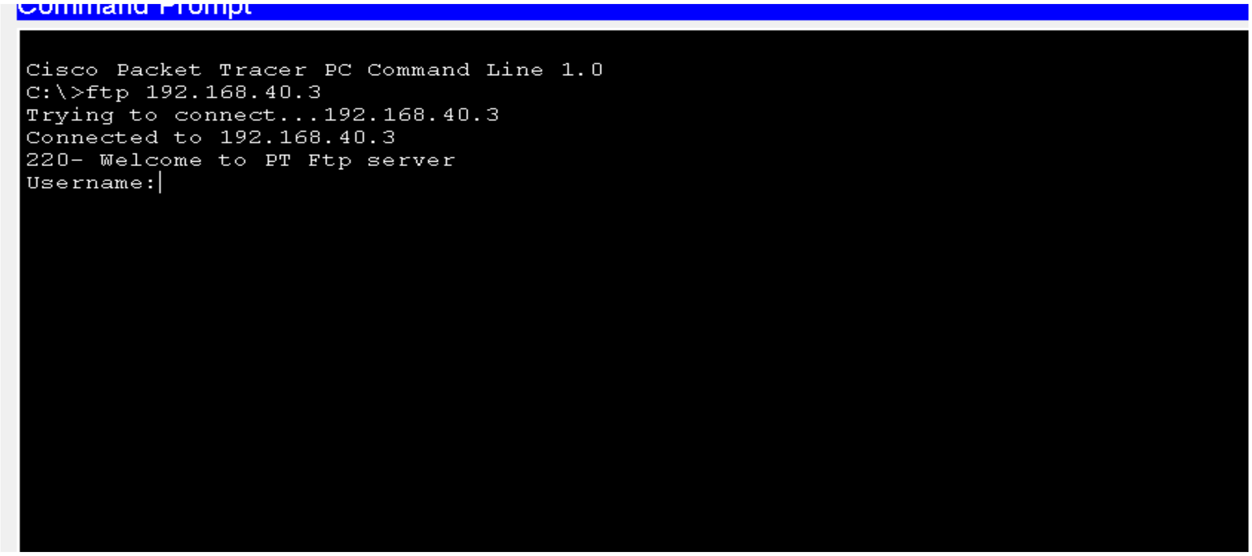
http://192.168.40.2

Cisco Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.

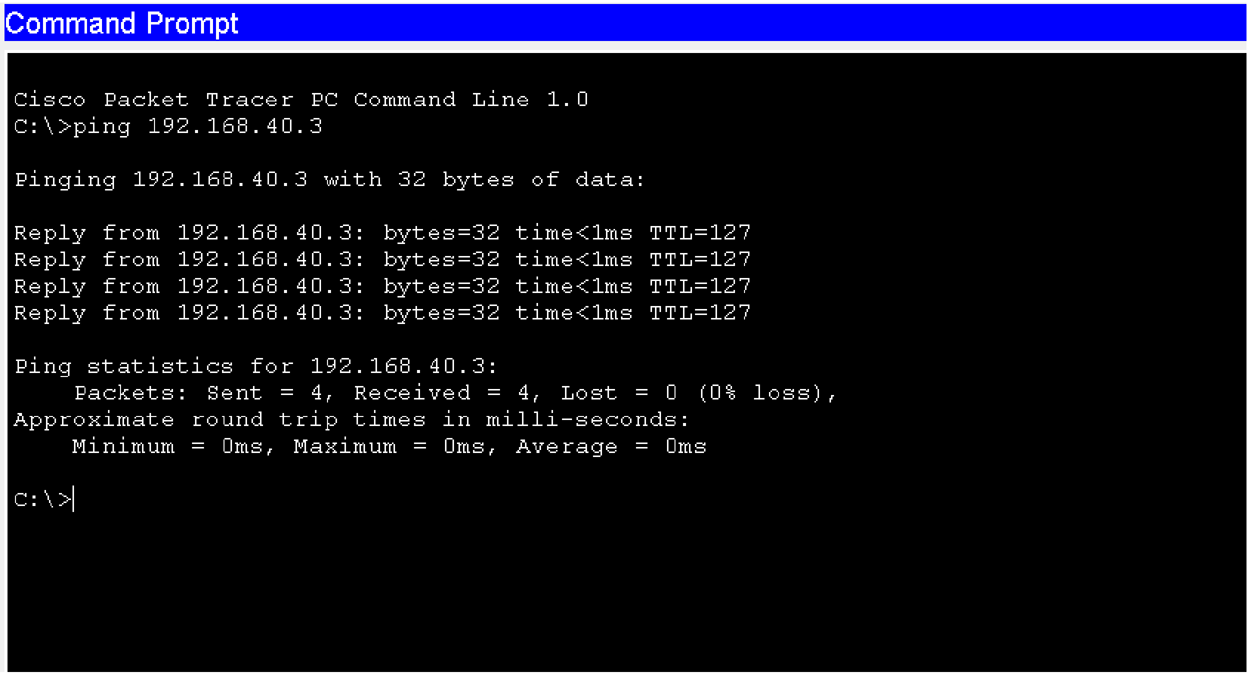
Quick Links:
A small page
Copyrights
Image page
Image

- **Command Prompt** and enter **Ftp 192.168.40.3**. This should connect to the File Server



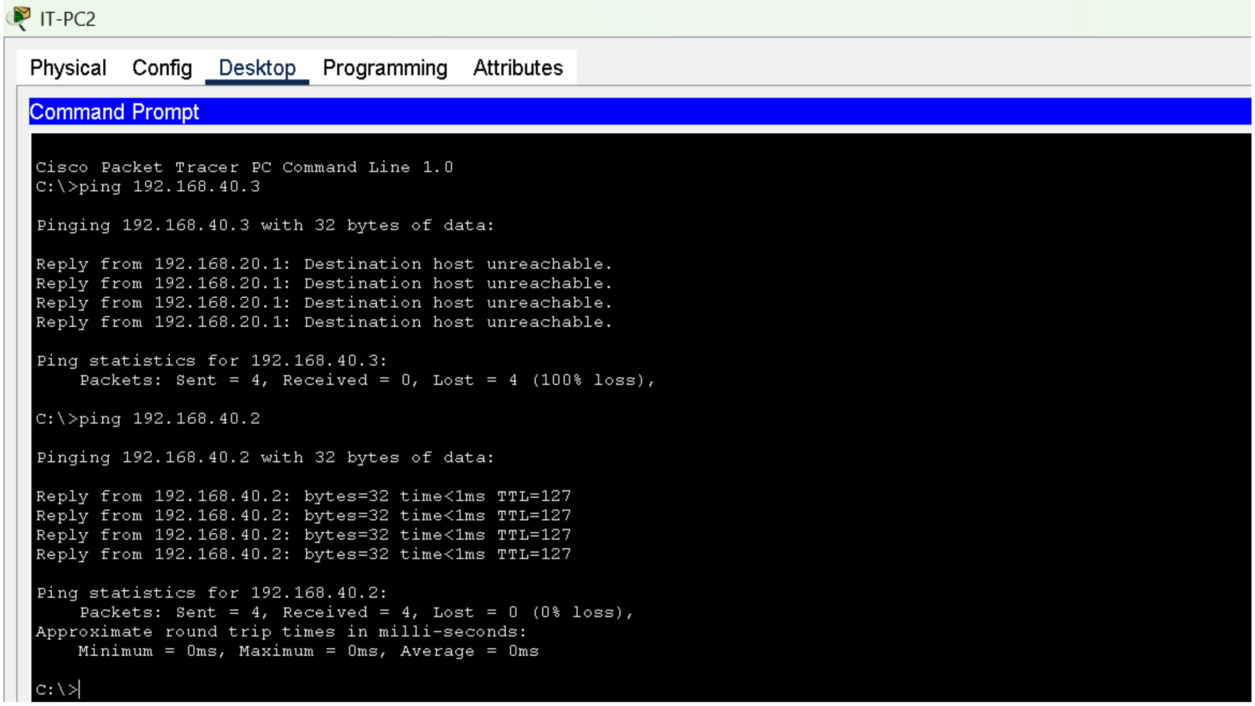
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ftp 192.168.40.3
Trying to connect...192.168.40.3
Connected to 192.168.40.3
220- Welcome to PT Ftp server
Username:
```

- Ping File Server **192.168.40.3**, this should work



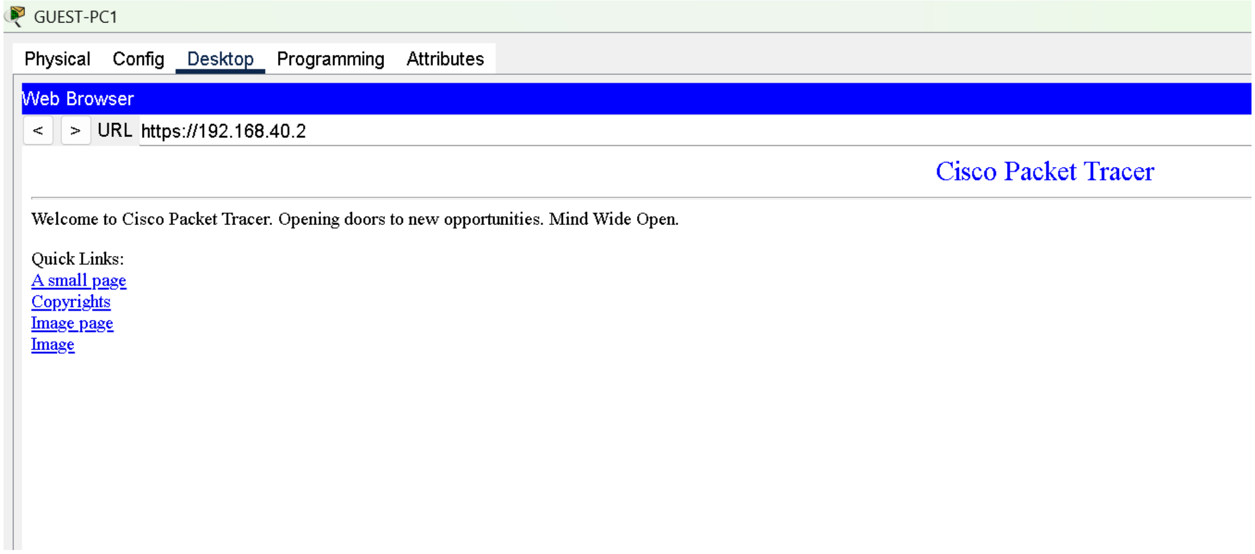
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.40.3

Pinging 192.168.40.3 with 32 bytes of data:

Reply from 192.168.40.3: bytes=32 time<1ms TTL=127
Reply from 192.168.40.3: bytes=32 time<1ms TTL=127
Reply from 192.168.40.3: bytes=32 time<1ms TTL=127
Reply from 192.168.40.3: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.40.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

- Now, from IT-PC2 **192.168.20.3**, pinging the File Server **192.168.40.3** should fail
- While pinging the web server **192.168.40.2** works perfectly fine from any PC in the IT Vlan **192.168.20.0/24**



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.40.3

Pinging 192.168.40.3 with 32 bytes of data:

Reply from 192.168.20.1: Destination host unreachable.
Reply from 192.168.20.1: Destination host unreachable.
Reply from 192.168.20.1: Destination host unreachable.
Reply from 192.168.20.1: Destination host unreachable.

Ping statistics for 192.168.40.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.40.2

Pinging 192.168.40.2 with 32 bytes of data:

Reply from 192.168.40.2: bytes=32 time<1ms TTL=127
Reply from 192.168.40.2: bytes=32 time<1ms TTL=127
Reply from 192.168.40.2: bytes=32 time<1ms TTL=127
Reply from 192.168.40.2: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.40.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

- From Guest-PC1 **192.168.30.2**, access the **web browser** and enter https://192.168.40.2 This should connect just fine.



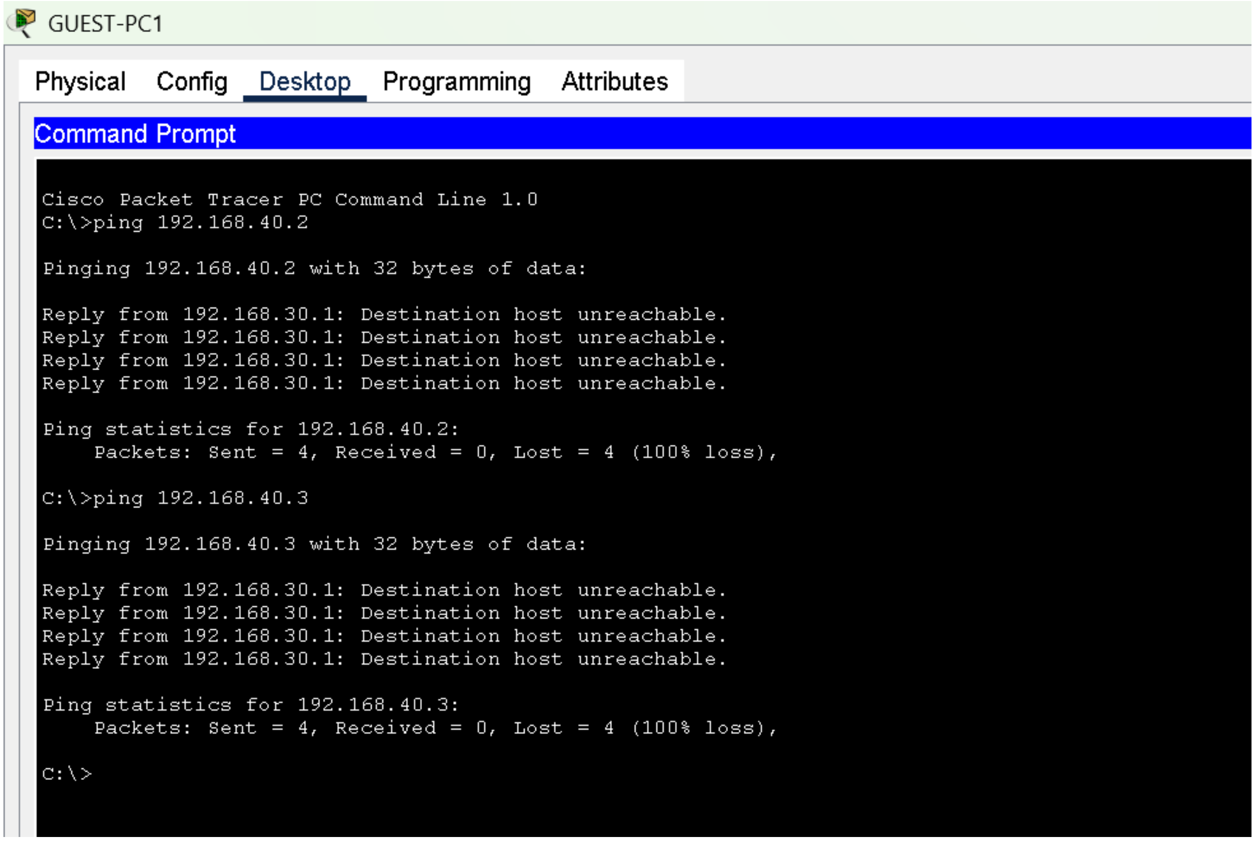
URL https://192.168.40.2

Cisco Packet Tracer

Welcome to Cisco Packet Tracer. Opening doors to new opportunities. Mind Wide Open.

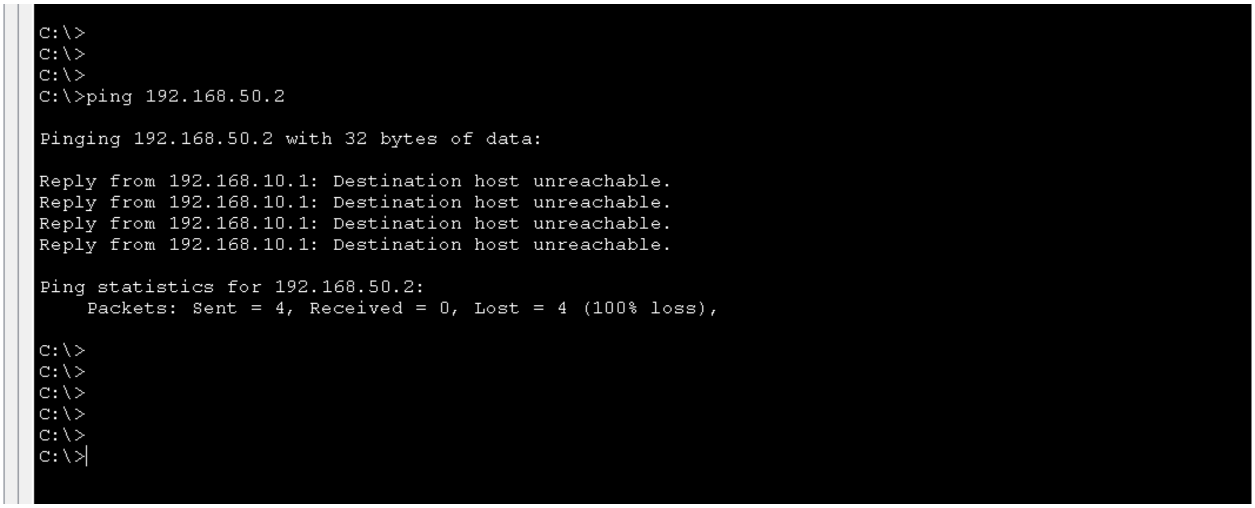
Quick Links:
A small page
Copyrights
Image page
Image

- From Guest-PC1 **192.168.30.2**, any other service (Pinging) should fail



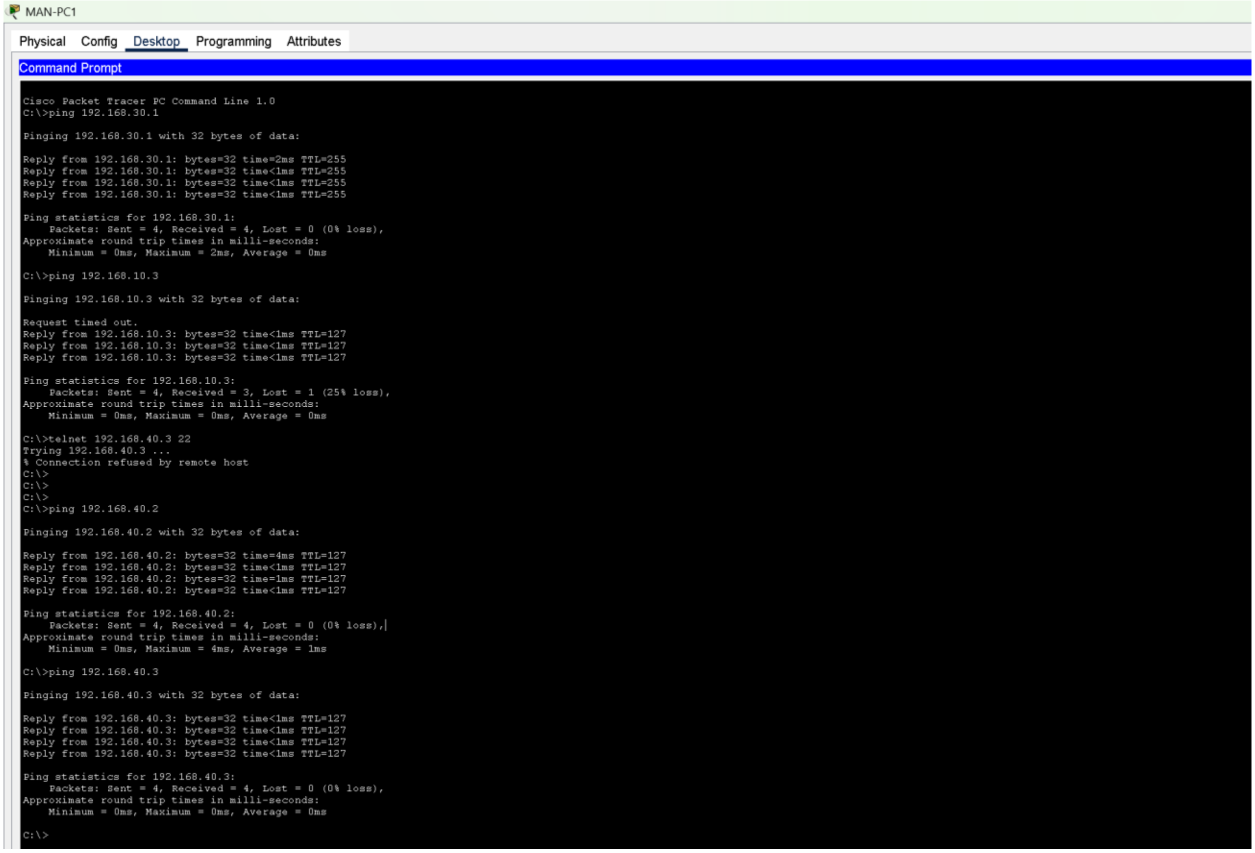
```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.40.2

Pinging 192.168.40.2 with 32 bytes of data:

Reply from 192.168.30.1: Destination host unreachable.
Reply from 192.168.30.1: Destination host unreachable.
Reply from 192.168.30.1: Destination host unreachable.
Reply from 192.168.30.1: Destination host unreachable.

Ping statistics for 192.168.40.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.40.3

Pinging 192.168.40.3 with 32 bytes of data:

Reply from 192.168.30.1: Destination host unreachable.
Reply from 192.168.30.1: Destination host unreachable.
Reply from 192.168.30.1: Destination host unreachable.
Reply from 192.168.30.1: Destination host unreachable.

Ping statistics for 192.168.40.3:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

- Any attempt from any other Vlan to access the Management Vlan **192.168.50.0** should fail



```
C:\>
C:\>
C:\>
C:\>ping 192.168.50.2

Pinging 192.168.50.2 with 32 bytes of data:

Reply from 192.168.10.1: Destination host unreachable.
Reply from 192.168.10.1: Destination host unreachable.
Reply from 192.168.10.1: Destination host unreachable.
Reply from 192.168.10.1: Destination host unreachable.

Ping statistics for 192.168.50.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
```

- Management-PCs **192.168.50.0** on the other hand should be able to access everywhere and anywhere



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.30.1

Pinging 192.168.30.1 with 32 bytes of data:

Reply from 192.168.30.1: bytes=32 time=2ms TTL=255
Reply from 192.168.30.1: bytes=32 time<1ms TTL=255
Reply from 192.168.30.1: bytes=32 time<1ms TTL=255
Reply from 192.168.30.1: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.30.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\>ping 192.168.10.3

Pinging 192.168.10.3 with 32 bytes of data:

Request timed out.
Reply from 192.168.10.3: bytes=32 time<1ms TTL=127
Reply from 192.168.10.3: bytes=32 time<1ms TTL=127
Reply from 192.168.10.3: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.10.3:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>telnet 192.168.40.3 22
Trying 192.168.40.3 ...
% Connection refused by remote host
C:\>
C:\>
C:\>
C:\>ping 192.168.40.2

Pinging 192.168.40.2 with 32 bytes of data:

Reply from 192.168.40.2: bytes=32 time=4ms TTL=127
Reply from 192.168.40.2: bytes=32 time<1ms TTL=127
Reply from 192.168.40.2: bytes=32 time=1ms TTL=127
Reply from 192.168.40.2: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.40.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms

C:\>ping 192.168.40.3

Pinging 192.168.40.3 with 32 bytes of data:

Reply from 192.168.40.3: bytes=32 time<1ms TTL=127
Reply from 192.168.40.3: bytes=32 time<1ms TTL=127
Reply from 192.168.40.3: bytes=32 time<1ms TTL=127
Reply from 192.168.40.3: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.40.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

We were not able to test Management remote access to the servers on port 22 (SSH) or port 23 (Telnet) because the service is not available in Cisco Packet Tracer.