

The University of Akron

IdeaExchange@Uakron

Williams Honors College, Honors Research
Projects

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

Spring 2024

Comprehensive Network Redundancy Implementation and Cybersecurity Hardening Project: Ensuring Resilience and Defending Against DHCP Starvation, STP Man-in-the-Middle, and Brute Force Attacks

Seth Shaheen
sms540@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 [Hardware Systems 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

Shaheen, Seth, "Comprehensive Network Redundancy Implementation and Cybersecurity Hardening Project: Ensuring Resilience and Defending Against DHCP Starvation, STP Man-in-the-Middle, and Brute Force Attacks" (2024). *Williams Honors College, Honors Research Projects*. 1794.
https://ideaexchange.uakron.edu/honors_research_projects/1794

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.

**Comprehensive Network Redundancy Implementation and
Cybersecurity Hardening Project: Ensuring Resilience and
Defending Against DHCP Starvation, STP Man-in-the-Middle, and
Brute Force Attacks**

Seth Shaheen

Department of CIS: Cybersecurity

Honors Research Project

Submitted to

*The Williams Honors College
The University of Akron*

Approved:

Dr. Nadhem Ebrahim
Honors Project Sponsor (signed)

malame 4/10/2024
Honors Project Sponsor (printed) Date

Stanley Smith
Honors Project Reader (signed)

[Signature] 4-11-2024
Honors Project Reader (printed) Date

Dr. Mahmood Safaei
Honors Project Reader (signed)

[Signature] 04/10/24
Honors Project Reader (printed) Date

Accepted:

Janet Kropff
Honors Faculty Advisor (signed)

Janet Kropff 4/10/24
Honors Faculty Advisor (printed) Date

Dr. Timothy O'Neil
Department Chair (signed)

Timothy O'Neil 4/15/2024
Department Chair (printed) Date



The University of Akron
College of Engineering
Department of Computer
Information Systems

CIS Senior Cybersecurity Project to fulfil the requirement of the CIS-Cybersecurity degree.

Comprehensive Network Redundancy Implementation and Cybersecurity Hardening Project: Ensuring Resilience and Defending Against DHCP Starvation, STP Man-in-the-Middle, and Brute Force Attacks

By Seth Shaheen

Student Name :- Seth Shaheen

Course Name: CIS Senior Cybersecurity Project 001

Section Number:- CISS 491-001

Supervisor: Dr. Nadhem Ebrahim

Abstract

I have created a network topology that contains three Cisco routers, three Cisco switches, and three endpoints. The network has been built using the software GNS-3. The endpoints on the topology include one VPC, one Kali Linux VM, and one Ubuntu Server VM. The main purpose of this network topology is to show the skills I have learned in my tenure at The University of Akron. This will be done by hardening this network to ensure that the network is impervious to cyber-attacks. The Kali Linux VM will act as the attacker on the network and conducted three attacks: STP man-in-the-middle, DHCP starvation, and a brute force attack against a SSH connection. The Ubuntu Server VM is configured as a Web Server hosting a website created by me called "SportsBallStore." SSH is configured and secured on the Ubuntu server as well. To obtain the Cisco router and switch images, the Cisco Modeling Labs – Personal license was purchased by me. The routers have been configured with hostnames, encrypted passwords, static IP addresses, OSPF, and DHCP. The three switches have been configured with hostnames, encrypted passwords, static IP addresses, and STP. These protocols have been properly configured and secured on each device. This report will discuss the background, problems, processes, and results of this topology, its configurations, and the attacks in detail.

Table of Contents

List of Tables, Figures, and Abbreviations	5
Acknowledgements.....	7
Dedication	8
Chapter 1: Introduction.....	9
1.1 Overview	9
1.2 Research Objectives	11
1.3 My Contribution in Brief	12
Chapter 2: Background and related work	14
2.1 Introduction	14
2.2 Previous Work	14
2.3 Technology Used	18
2.4 Opportunities for Contribution	19
2.5 Conclusion	20
Chapter 3: Problem statement and Solution	22
3.1 Introduction to the Problem.....	22
3.2 Impact of the Problem	25
3.3 Solution	28
3.4 Methods and Configuration	32
3.5 Conclusion	51
Chapter 4: Results and Analysis	53
4.1 Introduction.....	53
4.2 Results and Analysis.....	53
Chapter 5: Conclusions and Future Work.....	59
5.1 Contributions	59
5.2 Future Work	60
5.3 Conclusion	60

References	62
Appendix	64
Appendix A: General	64
Appendix B: Switch Configuration	66
Appendix C: Router Configuration	82
Appendix D: Ubuntu Server Configuration	92
Appendix E: Attacks	93

List of Tables, Figures and Abbreviations

CIS (Computer Information Systems).....	1
GNS-3 (Graphical Network Simulator-3)	2
VPC (Virtual Personal Computer).....	2
VM (Virtual Machine).....	2
STP (Spanning-Tree Protocol)	2
DHCP (Dynamic Host Configuration Protocol)	2
SSH (Secure Shell)	2
IP (Internet Protocol)	2
OSPF (Open Shortest Path First).....	2
SOC (Security Operations Center)	7
OS (Operating System)	9
USD (Computer Information Systems)	10
ASA (Adaptive Security Appliance)	10
BPDU (Bridge Protocol Data Unit).....	12
MAC (Media Access Control)	12
NSA (National Security Agency).....	15
IoT (Internet of Things)	14
TLS (Transport Layer Security)	15
HTTP (Hypertext Transfer Protocol)	15
FTP (File Transfer Protocol)	15
MFA (Multi-Factor Authentication).....	15
HTTPS (Hypertext Transfer Protocol Secure)	16
SNMP (Simple Network Management Protocol).....	16
SCP (Secure Copy Protocol).....	16
NIST (National Institute of Security and Technology)	16
VLAN (Virtual Local Area Network)	16
ACL (Access Control List).....	17
IBM (International Business Machine)	17

SSL (Secure Socket Layer)	19
VPN (Virtual Private Network)	20
ARP (Address Resolution Protocol).....	20
CDP (Cisco Discovery Protocol).....	20
VTP (VLAN Trunking Protocol).....	20
DTP (Dynamic Trunking Protocol)	20
GB (Gigabytes).....	22
RAM (Random Access Memory)	22
OSI (Open Systems Interconnection)	24
PII (Personal Identifiable Information)	26
MB (Megabytes)	32
GUI (Graphical User Interface)	33
Figure 1: Topology.....	35
VLSM (Variable Length Subnet Mask)	35
Figure 2: VLSM Table (Subnets Only).....	36
VTY (Virtual Teletype)	38
Figure 3: Encrypted Passwords:.....	39
Figure 4: STP Configuration	40
Figure 5: Port Security.....	42
Figure 6: DHCP Snooping	43
LAN (Local Area Network).....	44
Figure 7: OSPF.....	45
DNS (Domain Name System).....	47
Figure 8: DHCP Server.....	47
HTML (Hypertext Markup Language)	48
CSS (Cascading Style Sheets).....	48
Figure 9: STP Man-in-the-Middle.....	54
Figure 10: DHCP Starvation	56
Figure 11: Brute Force.....	57

Acknowledgements

I would like to acknowledge the assistance and education I have received at The University of Akron. This includes the professors I have worked with during my education: Dr. Ebrahim, Professor Smith, Professor Kropff, and Dr. Safaei. Sarah Hoge has been my academic advisor for my four years at The University of Akron and has assisted me in submitting this project and proposal to the University of Akron Idea Exchange. I would also like to thank Tim Zappitelli and Tyler Bissel for their assistance in preparing me for this project. They are SOC Analysts at The University of Akron and my supervisors at my job. Without the education these individuals have provided for me, I would not be where I am today with this project or my education.

Dedication

I would like to first thank all of those listed in the Acknowledgements for their guidance and assistance in both this project and my tenure of education at The University of Akron: Dr. Ebrahim, Professor Smith, Professor Kropff, Dr. Safaei, Tim Zappitelli, and Tyler Bissel. Without their help and encouragement, I would not be where I am today.

I was inspired to undergo this project because of the encouragement from The University of Akron, the Williams Honors College my family, my friends, and most importantly myself. Without these organizations, and people I would not have undertaken the arduous task of furthering my education and finally this project to acquire a bachelor's degree in CIS: Cybersecurity.

Chapter 1: Introduction

1.1 Overview

To demonstrate the principles of network hardening, I have built a network topology that contains three Cisco routers, three Cisco switches, a GNS-3 VPC, a Kali Linux VM, and an Ubuntu Server VM. The network was configured with four subnets. This network has been constructed and configured using the software GNS-3. Each of the routers and switches follow the naming system R# or S#. For example, R1 is router 1 and S1 is switch 1. The switches were configured to run STP as they were configured with redundancy. R2 was configured as the DHCP server for the subnet with the VPC and the Kali Linux VM. The Ubuntu Server was configured to act as a web server hosting the online sporting goods store “SportsBallStore.” This is a webpage that was created by me to represent a real online retail store. The Kali Linux VM acted as the attacker in this project and all three attacks were conducted from this VM.

GNS-3 is a software that is used to build and configure networks using proprietary device images. GNS-3 can also integrate standalone VMs into the topology as well to act as endpoints. A VM image file is a copy of an OS that can be manipulated and configured as if it is a physical device. Kali Linux is a Virtual Machine image that emulates a Linux Operating System that is often used for penetration testing because it has many penetration testing tools preinstalled on the machine. An Ubuntu Server VM is an image of the Linux server OS and can be configured to act as a real server that hosts data on it.

I have purchased the Cisco Modeling Labs – Personal license. This license is used to legally acquire any Cisco device image files. Cisco is a company that manufactures enterprise level networking devices such as: routers, switches, servers, and ASAs. An image is a file that contains the operating system of a device. These images are used to create VMs. The license allowed me to download the images for the routers and switches used in the network topology. The price of the Cisco Modeling Labs – Personal license is set to \$199.00 USD. For the topology that I built on GNS-3, I downloaded the Cisco CSR1000v 16.6.1 router image and IOSvL2 15.2.1 switch image files. These files had then been added into GNS-3 as router and switch templates.

The network has been configured with four subnets named Subnet 1, 2, 3, and 4. Subnet 1 is a /26 network that contains the three switches: S1, S2, and S3; the GNS-3 VPC and the Kali Linux VM. These devices are all connected to R1 which acts as the default gateway. Subnet 2 is another /26 network that contains the Ubuntu server connected to R3 as the default gateway. Finally, Subnets 3 and 4 are /30 networks that are the connections between R1 and R2 and R3 respectively. Once the configurations were completed on the topology, three attacks were conducted from the Kali Linux VM against the network devices and the Ubuntu Server. The attacks that were run are a STP man-in-the-middle attack, DHCP Starvation, and a brute force attack. The software Yersinia was used to conduct the STP and DHCP attacks, and Hydra was used to conduct the brute force attack on the Ubuntu Server.

1.2 Research Objectives

The objectives of this research project are to build and configure a hardened network that is impervious to the three attacks that were specified. These attacks are STP man-in-the-middle, DHCP starvation, and Brute Force attacks. This means that the objective of the configurations made on the GNS-3 network is to prevent these attacks and cause them to fail when they are conducted from the Kali Linux VM.

I have conducted extensive research on the configuration of a network and the steps that are needed to be taken to properly harden the network from the attacks discussed above. This includes both the previous work that has been done about the topic of network hardening and research on the actual configuration of the topology created in this project. The research and been done regarding router, switch, and Linux server configuration to prevent attacks that could be run against the network. This is a major pillar to achieving the objectives previously stated.

The final objective of my research was to develop a better understanding of the three attacks that I conducted in this project. As I have never actually conducted a cyber-attack prior to this project I had to learn both the purpose and the process behind the attacks I ran. This included why an attacker might conduct the attack along with how to run it myself. Finally, how and why the attack needs to be prevented or mitigated.

1.3 My Contribution in Brief

My Contribution to network hardening research is providing proof on concept on protecting the network from the attacks discussed above. These attacks include STP man-in-the-middle, DHCP Starvation, and Brute Force attacks.

The STP man-in-the-middle attack is a layer 2 attack where the attacker sends BPDUs to the switches to initiate an election of the STP Root Bridge. When a STP man-in-the-middle attack is successful the attacker's device will be elected as the Root Bridge. If the attacker's device is elected as the Root Bridge, then all network traffic flows through their device. This allows the attacker to intercept and interpret all data sent over the network. The attacker at this point could steal any personal information, eavesdrop in conversations, or collect and interpret network information. To prevent this attack, I needed to configure the network with port security and sticky MAC addresses on the switches.

The DHCP Starvation attack is a layer 3 attack that is conducted to interrupt the functionality of the network. This is done by the attacker's device sending many DHCP Discover packets in a very short span of time to lease all the available IP addresses from the DHCP server. Because a DHCP server will only lease one IP address to a single MAC address, the attacker's device will spoof, or create fake, MAC addresses to use as the source MAC address for each DHCP Discover packet. This then confuses the DHCP server into thinking that each request is coming from a new device on the network. Once all the available addresses have been leased to the spoofed MAC addresses and recorded in the DHCP server's MAC address table, then it can no longer issue any new devices an IP address. This removes the

ability for any legitimate devices to connect to the network and function properly. To prevent this attack, I needed to configure the switches with port security, sticky MAC addresses, and configure both the routers and switches with DHCP snooping to monitor the DHCP traffic.

Brute Force attacks are when the threat actor creates a long list of passwords to input into a software to run a series of login attempts against a service to crack the password in use. When this attack method is employed, the software issues a login attempt and uses a password from the list of passwords provided and if the attempt fails the software initiates the next login attempt and moves down the list. If the attack is successful the attacker has then compromised the service or account they are brute forcing, and the attacker can then sign into the service or account and do anything the legitimate user has access to do. This can be dangerous when it refers to web services, privileged accounts, or configuration protocols such as SSH. To prevent this attack, I have configured the SSH service on the Ubuntu Server VM to have a max password attempt to 1 and configured the firewall to block sign-in attempts for 2 minutes if the incorrect password is issued too many times. This will cause any brute force attempt to time out.

Chapter 2: Background and Related Work

2.1 Introduction

The purpose of this chapter is to discuss the research that has been conducted on the topic of both hardening networks and attacking them. The scope of the research is limited to previous work done on the subject, the configuration of the network, and the three attacks (STP man-in-the-middle, DHCP Starvation, and Brute Force attacks). The previous work includes any research, work, or projects that have been conducted prior to this project being conducted. This does not include prior configurations and work done by me. The research on the configuration is limited around the protocols and services configured to properly secure the network from attacks. Finally, the research on the attacks that has been conducted is limited to the scope of this project. This includes the effects of the attack, how to conduct the attack and why the attack is useful for the attacker.

2.2 Previous Work

Based on the research conducted there are several devices that need to be hardened: endpoints, switches, and routers. Endpoints or “hosts” are devices that users, data, and applications utilize to interface with the network itself including computers, servers, mobile devices, and the IoT (smart devices: thermometers, cameras, printers, etc...) [16]. There are numerous strategies that are used to harden these devices. These methods include auditing, physical access control,

closing open ports, network segmentation, removing unused devices, software, or services, restricting administrator access to devices and services, MFA, and securing remote access [16].

According to the NSA, hardening a network reduces the risk of unauthorized access to a network's infrastructure. Network and device vulnerabilities allow threat actors to exploit weaknesses to gain presence and persist within a network. These malicious actors have shifted their focus from attempting to gain access to an endpoint to trying to gain access to embedded network devices such as routers and switches. The vulnerabilities these threat actors exploit include but are not limited to manipulating configuration weaknesses, implanting malware in a device's OS, and taking control of routing protocols. These techniques that hackers use can result in them gaining access to the network, loss of network function, and even man-in-the-middle attacks [1].

The *Hardening Network Devices* article, published by the NSA states, every network device comes with services enabled by their manufacturer. These services and protocols are enabled by default to simplify network configuration; however, many of these services are not secure. Any service or protocol that is disabled cannot be exploited by a threat actor. Therefore, according to the NSA, any service that is not being directly utilized by the network should be disabled. Both Cisco and the NSA recommend that SSH and TLS be enabled by the network administrator to provide secure communication through encryption. They both recommend disabling services such as telnet, HTTP, and FTP as they are not encrypted protocols. SSH and TLS are good replacements for telnet as they allow the user to communicate

with the router over an encrypted medium [1]. HTTPS and SCP are secure replacements for HTTP and FTP. The NSA also recommends that the network administrator disables services such as: SNMP, any discovery protocols, IP Source Routing services, ping, and Zero Touch Provisioning [1].

SNMP can be exploited by an adversary to manipulate network configurations to disrupt or gain access to the system. Discovery and source routing protocols can also be used to gain access. FTP can be exploited to both extract files from the network (such as configuration files) and upload malicious files to the network. This can be used to both gain access and install viruses and worms on the network [1]. Zero Touch Provisioning is a dangerous service as it allows the network devices to download files without user interaction [1]. This means that a malicious actor could disguise malware as a legitimate file for the network devices to download.

At the hardware level the NSA, NIST, and the notable source “BeyondTrust.com” state that the network administrator should enable port security, shut down unused interfaces and switch ports, and place these switchports on an unrouted and heavily audited VLAN [1] and [3]. They also recommend that all unused routing protocols be disabled.

The third category that these sources listed above provide recommendations on is system security. This contains processes such as ensuring that all device OSs are updated regularly [3]. They also recommend that firewalls are put in place and properly configured on the network and on end devices to block suspicious or unwanted traffic. This involves configuring ACLs to properly achieve this security.

Finally, identity management should be put in place. This involves services such as MFA in case an account or password on the network is compromised. This extra layer of security will allow the network administrator to properly remediate the intrusion before the attacker can fully infiltrate the system or network [1].

According to IBM, penetration testing is the process of conducting a fake cyberattack on a network or system. The purpose of this is to determine the network or system's vulnerabilities to secure them from future attacks. It also provides the security professionals with a comprehensive understanding of how these vulnerabilities can be exploited. There are four types of penetration tests: Application, Network, Hardware, and Personnel pen-tests [6]. Application pen-tests are used to test the security of applications. Network pen-tests look to find vulnerabilities in the network itself. Hardware pen-tests test the security of a device connected to the network. Personnel pen-tests look for weaknesses in the employee's cyber hygiene [6].

There are many tactics and techniques an attacker can use to compromise a device or network. The MITRE Attack Framework outlines these in a comprehensive and easily understood way. It can give the cybersecurity professional a step-by-step guide on how the attacker may exploit vulnerabilities to gain access to the system they are securing [20]. This includes the tactics: reconnaissance, resource development, initial access, execution, persistence, privilege escalation, defense evasion, credential access, discovery, lateral movement, collection, command and control, exfiltration, and impact [20]. Each of these tactics has numerous techniques that are outlined and discussed within [20]. All three of the attacks that I conducted

during my penetration test of the topology I built can be categorized into the tactics and techniques MITRE outlines.

2.3 Technology Used

In the process of network security, there are numerous technologies that are used. This technology includes protocols discussed earlier such as TLS, SCP, SSH, ACLs, VLANs, HTTPS, MFA, STP, DHCP, routing protocols, and encryption protocols [1]. There are devices that can be installed in networks as well to provide security. The most common device is an ASA. An ASA is a physical device that acts as a firewall by configuring ACLs on the device [16].

SSH is a protocol used to remotely connect to a device. This is a more secure alternative to the Telnet protocol. The connection provided by SSH establishes an encrypted tunnel between the source and destination of the connection [1]. ACLs are lists of IP addresses and protocols that can be allowed or denied connection to the network. ACLs are used to configure firewalls or ASAs. ASAs are hardware-based firewalls that can act as Intrusion Detection or Prevention Systems. These devices are considered best practice for securing networks [1] and [16].

HTTPS is the secure version of HTTP. HTTP is a protocol used to establish connections to websites, webpages, and web applications via a browser. Browsers are Search Engines such as Google, Bing, or Yahoo. The HTTPS protocol functions almost identically to the HTTP protocol however it provides an encrypted connection between the user's device and the web server [17]. Encryption protocols such as

TLS, SSH, or SSL obfuscate data into strings of characters that are completely unreadable without the encryption key [17].

For penetration testing to prove that a network has been properly hardened, Cybersecurity professionals use Kali Linux. Kali Linux is an open-source Linux OS that comes with many penetration testing tools installed by default [18]. Linux is a very bare, open-source OS that offers much higher levels of configuration compared to its Windows or Mac (Apple) counterparts. The Linux OS comes in more than one version. The other popular Linux installation is Ubuntu. Ubuntu is a barer and user-friendly version of the Linux OS [19]. Ubuntu also offers a Linux server VM as well. The Ubuntu Linux Server VM can act as and do everything that a physical server can do [19].

2.4 Opportunities for Contribution

Sections 2.2 Previous Work and 2.3 Technology Used discuss some of the background research conducted on the network hardening subject. This research has opened various opportunities for me to contribute knowledge to this topic. This section will discuss these opportunities. The contribution that I will be making is providing proof of concept for a hardened network. Proof that a network has been hardened needs to be provided. The best way to do this is to conduct a penetration test on the network. To prove that the network is secure the attacks conducted during the penetration test cannot be successful. This would show that the network

has been successfully secured to protect or mitigate the attacks conducted by the threat actor.

There is a plethora of attacks that can be run during a penetration test. To show the security of only the network, the best options are to run attacks against network protocols and devices. This includes protocols and devices such as switches, routers, STP, DHCP, VPN, ARP, CDP, DTP, VTP, SSH, and more [7]. Yersinia is a great penetration testing tool for the protocols listed above. It is a layer 2 attack tool that can be used to attack everything listed except SSH [7]. The method to break SSH involves attacks such as a brute force attack to break the encryption key or password that is used to protect the system.

In many cases, to show the security capabilities of a network topology, the network professionals will create vulnerability reports. This is an effective way to showcase this and state what parts of the network may need to be hardened further [6]. Penetration testing, however, is much more effective to do this because it provides proof of the vulnerability or of the security in place [6]. They also provide an in-depth understanding of the path and process a threat actor may use to exploit the vulnerabilities. For these reasons, Cybersecurity experts highly recommend penetration testing your network [6].

2.5 Conclusion

This chapter discussed research that I conducted about network hardening and penetration testing. The NSA and NIST are two very prominent organizations

regarding cybersecurity and how it is conducted. They are both organized and funded by the United States government to conduct research and testing on protocols and devices used in our network today. They as well as private businesses such as IBM, Kali Linux, Ubuntu, Cisco, and many more strive to advance the technology we use for our networks to not only provide better and faster connectivity but better security as well.

There are many technologies that are put in place and configured to harden a network. Most of these aspects of security and networking are placed on the routers and switches; however, some are placed on the endpoints like desktops, laptops, servers, mobile phones, and IoT devices. The reason we secure all of the devices on a network and not just the network devices themselves is because it only takes one device getting compromised for an attacker to gain access to the whole network.

Chapter 3: Problem statement and Solution

3.1 Introduction to the Problem

The first problem to solve is the creation and configuration of the topology. I built a network topology that contains three Cisco routers, three Cisco switches, and three endpoints. The endpoints are one Kali Linux VM that acts as both a user and the attacker, an Ubuntu Server VM that is configured as a web server hosting the website for a fake sporting goods store that I created called “SportsBallStore,” and finally, a GNS-3 VPC that will act as a user on the network [18] and [19]. This topology was created on GNS-3, which is a network virtualization software that can utilize virtual devices of any kind. This topology was split into four subnets that are named Subnet 1, 2, 3, and 4 accordingly. All the data for the topology is saved in a GNS-3 Virtual server that I configured using the VMWare hypervisor [23]. The Kali Linux VM and Ubuntu Server VM are stored on the hypervisor Oracle VirtualBox. The configuration of this network has caused numerous problems.

The first issue that I faced in the configuration of this network was that I had to buy a Cisco Modeling Labs – Personal license to obtain the images for the OS of the router and switch. This license costs \$199.00 USD [21]. Without obtaining this license any use of a Cisco OS image file would be considered pirating and be considered a criminal act.

The next issue I faced was the device requirements to run the network, properly and effectively, I built on GNS-3. My laptop had only 16 GB of RAM and the topology required nearly 20 GB of RAM function. Originally, this project’s network was

supposed to contain six Cisco switches instead of 3 and four endpoints instead of three. Three switches in Subnets 1 and 2 instead of just Subnet 1 and two VPCs instead of just one. I could only run parts of the topology at one given time, and it would cause me not to be able to run any other programs while GNS-3 was running. The major issue that this project was meant to solve was the security issues around the network. A network with basic configuration is not secure whatsoever. Basic configuration includes hostnames, IP addresses, subnetting, and routes. While a network that is configured at this level will provide the users with connectivity to all aspects of the network, it does not provide security from any threats as the network is not hardened.

Some of the threats that an unhardened network faces are vulnerabilities to many attacks including STP, DHCP, and attacks on SSH. This is because most networks are configured to introduce redundancy with STP and scalability with DHCP. Also, SSH provides the ability for network and security technicians to remotely connect to the network devices and some endpoints such as servers. The reason they may establish an SSH connection is to edit configurations of that device remotely.

STP can be attacked in a couple of ways. One method is an STP man-in-the-middle attack. This attack is when the attacker uses software to declare themselves as the Root Bridge. The next attack is STP denial of service. This is when the attacker sends BPDUs to the switches to continuously restart the STP process [5]. The protocol STP can be attacked using the software Yersinia. Yersinia is a tool that

is used on Kali Linux to run attacks on network protocols in layer 2 of the OSI framework.

The next protocol that needs to be protected from this project's solution is DHCP. DHCP can also be attacked in multiple ways. The most effective way to attack it is via DHCP Starvation [9]. DHCP Starvation is an attack where the attacker uses software such as Yersinia to spoof a very large amount of MAC addresses to send many DHCP Discover packets to the DHCP server at once to lease all available IP addresses [9].

The next problem to solve is securing the SSH protocol that is configured on the Ubuntu Server. SSH can be attacked via a Brute Force attack using software such as Hydra [22]. Hydra is a password cracking tool on Kali Linux that can be used to Brute Force numerous protocols and authentication methods including SSH [22].

Another problem that I looked to solve is the possibility of a threat actor just consoling into a network device and viewing configurations or even changing them. This can be done if passwords are not configured onto each device and if the passwords are not strong. A strong password is at least 8 – 10 characters long and includes lower-case letters, upper-case letters, number, and special characters like \$, #, _, @, etc... [16].

The final problem that I needed to solve was conducting the proper research to adequately conduct this project and write this report. This assignment required me to pull together and apply all the knowledge that I have gathered while both

studying at the University of Akron and working in the Cybersecurity field. This was the most extensive project or assignment I have ever had to do and the research to complete this was no different. Being the capstone project of this major, makes this the most important report and project I have ever created. I have had to pour countless hours into the creation of both this network and the report that followed.

3.2 Impact of the Problem

The problems discussed in section 3.1 above have several impacts on the creation of the network itself. This was the most time-consuming problem to solve. It is also the most important problem because without completing this step there would be nothing to use to prove the proof of concept that I have discussed earlier in this report. The proof is extremely important because without it there would be no project at all.

The problem of the cost of the Cisco Modeling Labs – Personal license has a large impact on this project, since I am just a college student, a \$199.00 expenditure is not cheap and can be quite detrimental to my savings. However, this was a very necessary and needed purchase as I needed to acquire the Cisco OS images to be able to begin the project itself. Without this, I would have been perpetually stuck at square one with my network as I would not have been able to create a working topology that meets the requirements of this project and of the research being conducted.

The next issue I face when configuring the network was the RAM issue that I stated in section 3.1. This was that the laptop I own only had 16 GB of RAM installed on it and the topology that I was building in GNS-3 required more RAM than what I had. This was causing the routers and switches to loop in their boot up sequences. This rendered the entire topology useless until the issue was solved. Unfortunately, due to this, I had to remove three of the switches and one of the VPCs from the original topology resulting in the topology that will be shown in section 3.4 and the Appendix. After reducing the network, I still had to increase the RAM of my laptop to 32 GB instead of 16 GB to fully run the network topology without any errors. This added another \$110.00 USD roughly to the final cost of this project for the new RAM cards and the tools to install them, bringing my total cost to over \$300.00 USD.

The next issue was securing the network from the three attacks that were discussed. The first attack is the STP man-in-the-middle attack. This is an attack where the threat actor connects their device to the network and then uses Yersinia or a similar tool to elect themselves as the Root Bridge [5]. The impact of this is that if the attack is successful, then the attacker will have all the network traffic on that subnet routed through their personal device. This is very dangerous for a network such as this one because there are users connecting to an online retail store. Meaning that PII such as credit card numbers, and account information is being transmitted across the network. If the attacker gains access to this network traffic, they can then steal information that is very confidential to each user on the network.

The next attack that the network needed to be hardened to protect itself from is DHCP Starvation. This attack is when the attacker connects to the network and

uses Yersinia to send many DHCP Discover packets in a very quick succession. Each request gets sent from a unique spoofed MAC address to confuse the DHCP server that is the target of these attacks into thinking each request is sent from a new device [9]. This is because the source of each request is separate from the previous request. The reason this step of the attack is important is because a DHCP server is programmed to only lease one IP address to a MAC address [9]. If this attack is successful then legitimate users will no longer be able to lease an IP address from the DHCP server. This then makes it impossible to use the network by denying service to anyone but the attacker.

The final attack that the network needed to be secured from are Brute Force attacks. This is because SSH was configured on the Ubuntu server to allow me to emulate a real-world network where the network may be using a server that is not directly on the premises of the physical network. In these cases, a network administrator would need to establish a SSH to the server to issue any configuration changes to the server itself or the webpage. The Brute Force attack can prove to be a very dangerous attack if the SSH service is not properly configured. If the attacker manages to Brute Force the SSH connection, then they can possibly sign-into the server with root privileges and issue any changes they want or steal any information that is saved on the server. A Brute Force attack on SSH is when the attacker uses software such as Hydra to initiate sign-in attempts and try any number of passwords from a list to attempt to break into the account [22].

The next problem is the possibility of a random person or threat actor consoling into the network devices. If a strong password as discussed in section 3.1

is not configured on each device, then the configurations and data that is saved on a device is not secure. This is the same principle as configuring a password on your personal computer. Even if the person that consoles into the device is not a malicious actor they could accidentally change the configuration and break the network altogether.

The final problem was conducting the research for this project. This problem is imperative to the completion of this project and this report because it enabled me to properly configure and harden the topology from the attacks discussed. It also allows me to discuss the information being discussed in the proper manner. Without conducting the proper research, I would not be able to even begin to complete such an in-depth analysis of the cybersecurity techniques and configurations that have been involved in this process.

3.3 Solution

The first problem that I solved is the acquisition of the proper licensing, image files, and software to build this network. I purchased the Cisco Modeling Labs – Personal license for \$199.00. This is a Cisco learner’s license that provides the user with IOS images for Cisco routers, switches, and ASAs. I have utilized the router and switch image files from this license. These image files are distributed with a default configuration that is identical to the Cisco devices themselves [21]. Next in the preparation phase was to install Kali Linux and an Ubuntu Server VM. The Kali Linux VM will represent an attacker’s device connecting to the network [18] and [19].

Here is where Yersinia and Hydra have been installed. The Ubuntu Server VM is configured to run as a web server hosting a basic webpage created by me [19]. It is also running OpenSSH for secure remote connection to the device. These virtual devices will be used to build a network on the GNS-3 software. The GNS-3 software as well as its respective GNS-3 Virtual Server has been installed on my computer [23].

To create the network topology, I used GNS-3. This network contains three routers, three switches, and three endpoints including the Kali Linux and Ubuntu Server VMs. There are three switches on this topology to ensure that redundancy can be configured on each Subnet 1. Each of the three end devices are connected to different points on the network. The GNS-3 VPC is connected to switch 1, the Kali Linux VM to switch 2 and the Ubuntu server to router 3.

I used two different software tools on the Kali Linux VM to conduct these attacks. Yersinia was utilized to conduct the DHCP starvation attack and the STP Man-in-the-middle attack. Yersinia is a software-based penetration testing tool on Linux that is used to attack layer 2 protocols [7] and [9]. Hydra was used to conduct a Brute Force attack to attempt to gain administrative access to the network configuration. Hydra is a Linux-based password-cracking tool that can be used to Brute Force the SSH protocol configured on the Ubuntu Server VM [22].

I first configured this hardened network to solve the issue of connectivity between each device. This involved making any basic configurations such as subnetting,

hostnames, IP addresses, default gateways, and routes between the subnets using OSPF.

After configuring the basic connectivity of the topology, I ran into the RAM issue presented in section 3.1 and 3.2 above. This was that the 16 GB of RAM my laptop has was not enough to run the topology on the GNS-3 software because I could not allocate enough memory to the GNS-3 Virtual Server. This made it impossible for the Virtual Server to host all the devices at the same time. To solve this issue, I paid approximately \$110.00 to buy a set of two 16 GB Crucial RAM cards to replace the two 8 GB HP RAM cards on my laptop. This effectively doubled my RAM capacity from 16 GB to 32 GB, allowing me to allocate 24 GB of RAM to the Virtual Server to run all the devices on the topology at one given time. This also allowed me to run MS Teams to enter a meeting by myself to record the STP, DHCP, and Brute Force attacks being run on the network.

I have secured the network to be able to prevent DHCP starvation, STP Man-in-the-middle, and Brute force attacks, along with other network security features. These features include password protecting the configurations, implementing encryption, disabling unencrypted protocols.

To prevent DHCP starvation, I have configured the MAC addresses to be static with the leases. I then configured port security to mitigate the risk of a STP Man-in-the-middle attack and DHCP attacks. This is prevented because port security negates the ability for a new network device to be connected to the network. The port security that has been configured includes a maximum of the number of MAC addresses

that can connect to a switch port, sticky MAC addresses, and DHCP snooping. I have also configured the switch ports to shutdown in the event of a security violation.

Disabling unencrypted protocols and enabling encryption provides confidentiality to the data being sent on the network. This includes enabling SSH and HTTPS. I have configured the web server to only utilize HTTPS instead of HTTP because this will help prevent the data on the website from being intercepted by the malicious actor [17]. I configured strong passwords to prevent a Brute Force attack from being feasible. Brute Force attacks are based on the amount of time it will take for the software to crack a password [22]. The stronger the password is the longer the software will take to crack it. This means if the password used is strong enough, then the Brute Force attack is virtually impossible. The second step I took to prevent the Brute Force attack on the Ubuntu Server SSH connection was configuring firewall rules on the switch itself to block a connection for two minutes in the event of multiple incorrect sign in attempts [13].

3.4 Methods and Configuration

To begin the configuration of the project I had to first download and configure the needed software. This includes GNS-3, a Kali Linux VM and an Ubuntu Server VM. When GNS-3 was downloaded I had to install the GNS-3 Virtual Server as well and configure it [23]. This server was configured to run on my laptop's local host. This is the Loopback interface and IP address 127.0.0.1. The GNS-3 Virtual Server has been named "GNS-3 VM" and has been allocated 24 GB of RAM and 8

processors. From this point on the GNS-3 Virtual Server will be referred to as GNS-3 VM. The reason for RAM is that all the network devices and the GNS-3 VPC will be run and saved on the GNS-3 VM.

After downloading and configuring the GNS-3 VM, I downloaded and configured the Kali Linux VM and the Ubuntu Server VM. The Kali Linux VM was given the hostname LinuxVM and will be referred to as that going forward. The image file downloaded for the Linux VM is “kali-linux-2023.4-virtualbox-amd64.vbox” [18]. The LinuxVM was then allocated 2 processors and 4 GB or 4096 MB of RAM. This is more than enough to properly run the Linux VM. Once the initial configuration was completed the user “kali” was created. It was on this user account that I downloaded the Yersinia and Hydra software packages [7] and [22]. The process for these installations was to update the VM using the “sudo apt update && sudo apt upgrade” command and typing “yes” to the confirmation prompt. Then the software was installed with the “sudo apt install yersinia” and “sudo apt install hydra” commands and again typing “yes” to the confirmation prompts.

The file “ubuntu-22.04.4-live-server-amd64.iso” was then downloaded to install the Ubuntu Server VM [19]. This process is quite like the process to install and configure this VM as with the LinuxVM. The Ubuntu Server VM was then created with the user account “ubuntu.” The hostname “Ubuntu_Server” was given to the Ubuntu Server VM and it will be referred to as such from now on. It was then updated with the same process: “sudo apt update && sudo apt upgrade” command and typing “yes” to the confirmation prompt. Then I installed the Apache2 service to enable the Ubuntu_Server to host a webpage [24]. This was done via the “sudo apt

install apache2” command followed by entering “yes” to the confirmation prompt [24]. After that, I used the “sudo apt install openssh” command followed by entering “yes” to the confirmation prompt to install the OpenSSH service. This software is used on a Linux server to configure SSH to enable secure and encrypted connections to the Ubuntu_Server [14]. The final installation that I made on the Ubuntu_Server was to install a GUI on it for ease of use and configuration. The GUI I installed is called LightDM. This is the GUI that is used on an Ubuntu Workstation VM which both looks good and is user friendly. Installing the GUI was done via the command “sudo apt install lightdm” followed by “sudo apt install ubuntu-desktop.” Then I had to restart the Ubuntu_Server to allow the GUI to start and be used [11].

The next step that I took to configure the network was to buy the Cisco Modeling Labs – Personal license. This license costs \$199.00 and gives me access to the Cisco networking device OS images. Refer to Appendix A for a screenshot of the license. The devices I decided to use from the license are the CSR1000v 16.6.1 for the router and IOSvL2 15.2.1 for the switch. The image file for the CSR1000v 16.6.1 router OS is “csr1000v-universalk9.16.6.1.qcow2.” The image file for the IOSvL2 15.2.1 OS is “vios_l2-adventerprisek9-m.SSA.high_iron_20180619.qcow2” [21] Once the files had been downloaded via the license, they were able to be added to the GNS-3 software and saved as templates in the GNS-3 VM. This was done by selecting the “All Devices” tab on the left of the GNS-3 GUI, followed by selecting the “New Template” button at the bottom of the “All Devices” popup. At this point by following the steps in the importation wizard I was able to import both the switch and

router OS image files: “csr1000v-universalk9.16.6.1.qcow2” and “vios_l2-adventerprisek9-m.SSA.high_iron_20180619.qcow2.”

The last step before being able to configure the network was to import the LinuxVM and Ubuntu_Server into the GNS-3 software and topology. To do this I selected the “Edit” tab at the top of the screen then selected “Preferences.” In the “Preferences” popup window I was able to select the “VirtualBox VMs” tab in the “VirtualBox” section and then click the “New” button at the bottom of the popup window. From here I was able to again follow the steps provided by the importation wizard to import the VM image files: “kali-linux-2023.4-virtualbox-amd64.vbox” and “ubuntu-22.04.4-live-server-amd64.iso.”

Once all these devices had been installed and imported into the GNS-3 software and GNS-3 VM, I was ready to begin building and configuring the topology. The first step was to build and cable the network as shown in Figure 1 below. To create this topology, I had to create a new project file named “CIS_Senior_Project.gns3.”

Figure 1: Topology

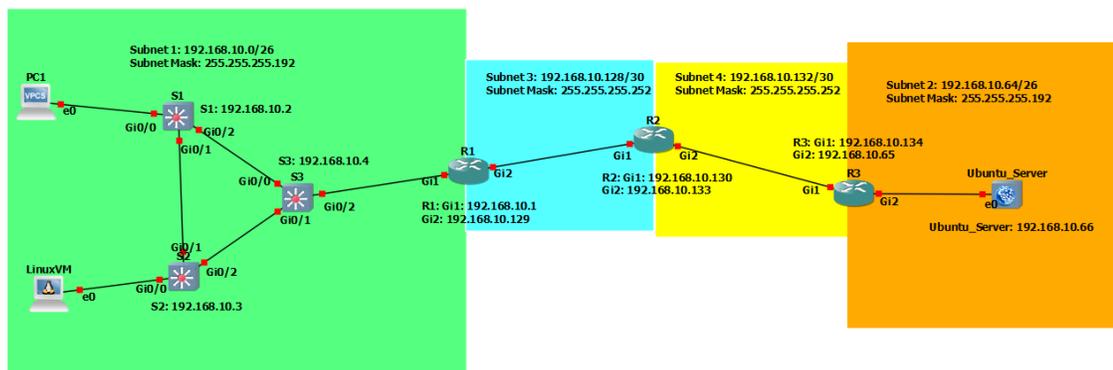


Figure 1 is a screenshot of the topology that I built in GNS-3. This topology is saved on the GNS-3 VM.

As shown in “Figure 1” above, I created a topology that contains three routers, three switches, and three endpoints. The next step was to subnet the topology into four subnets using the strategy VLSM. These are named Subnet 1, Subnet 2, Subnet3, and Subnet 4 accordingly. Figure 2 below shows a shortened version of the subnetting shown as a table. The full table can be seen in Appendix A.

Figure 2: VLSM Table (Subnets Only)

Subnet	Network Address	Subnet Mask	Broadcast Address	Available IP Address Range
1	192.168.10.0	255.255.255.192	192.168.10.63	192.168.10.1 – 192.168.10.62
2	192.168.10.64	255.255.255.192	192.168.10.127	192.168.10.65 – 192.168.10.126
3	192.168.10.128	255.255.255.252	192.168.10.131	192.168.10.129 - 192.168.10.130
4	192.168.10.132	255.255.255.252	192.168.10.135	192.168.10.133 - 192.168.10.134

Figure 2 is a shortened version of the VLSM table that is shown in Appendix A.3. This table shows the IP addresses assigned to each of the four subnets.

Subnet 1 is a /26 network (subnet mask: 255.255.255.192) that can hold 64 hosts total. The network address for Subnet 1 is 192.168.10.0/26. Subnet 1 contains the three switches which have been assigned the hostnames S1, S2, and S3 and they will be referred to as those names from here on. It also contains the LinuxVM and PC1. PC1 is the hostname that was assigned to the GNS-3 VPC. Both hosts in this subnet are assigned IP addressed via DHCP from R2 which is the hostname

for router 2. The DHCP pool will be discussed in more detail when discussing the configuration of R2. Subnet 1 is connected to interface Gi1 on R1 which is the hostname for router 1. The IP address assigned to Gi1 on R1 is 192.168.10.1. This is the default gateway that has been assigned to Subnet 1. Subnet one is colored in light green in Figure 1.

Subnet 2 is also a /26 (subnet mask: 255.255.255.192) network that can hold 64 hosts. The network address for Subnet 2 is 192.168.10.64/26. Subnet 2 contains only the Ubuntu_Server. DHCP is not configured on Subnet 2 because the only host on the subnet is Ubuntu_Server. Servers are not generally assigned addresses via DHCP. They are generally assigned IP addresses statically. The subnet is connected to the interface Gi2 on R3 which is the hostname assigned to router 3. The IP address assigned to Gi2 on R3 is 192.168.10.65 and this is the default gateway for Subnet 2. Subnet 2 is colored orange in Figure 1.

Subnet 3 and Subnet 4 are both /30 (subnet mask: 255.255.255.252) networks that represent the connections between the routers. Subnet 3 is the subnet that is housed between R1 and R2. R1 interface Gi2 is connected to R2 interface Gi1. Subnet 3 is assigned the network address 192.168.10.128/30 and does not have a default gateway as there are only routers in this network. Subnet 3 is colored light blue in Figure 1. Subnet 4 is the subnet that is housed between R2 and R3. R2 interface Gi2 is connected to R3 interface Gi1. Subnet 4 is assigned the network address 192.168.10.132/30 and does not have a default gateway as there are only routers in this network. Subnet 4 is colored yellow in Figure 1.

The next step of the project was to configure the routers and switches with basic configuration. The configuration of the routers and switches involves the configuration of the hostnames, IP addresses, passwords, encryption, STP, port security, DHCP snooping, and routes. All switches and routers were assigned their respective hostnames via the "hostname" global configuration command. First, I will discuss the basic configuration of the switches. S1 is the switch connected to PC1. PC1 is connected to Gi0/0 on S1. S1 is also connected to S2 via Gi0/1 on S1 and S2; and connected to S3 via Gi0/2 on S1 and Gi0/0 on S3. S2 is connected to the Linux VM via interface Gi0/0 on S2 and it is connected to S3 via interface Gi0/2 on S2 and interface Gi0/1 on S3. S3 is connected to R1 via interface Gi0/2 on S3 and interface Gi1 in R1.

All three switches are configured with the static IP addresses on VLAN 1. VLAN 1 is the VLAN that is configured for Subnet 1. S1 is assigned the IP address 192.168.10.2 with a subnet mask of 255.255.255.192. S2 is assigned the IP address of 192.168.10.3 255.255.255.192 on VLAN 1. S3 is assigned 192.168.10.4 255.255.255.192 on VLAN 1. Each of the three switches also have the IP address 192.168.10.1 assigned as the default gateway for VLAN 1. These were assigned starting from global configuration mode via the commands: "interface vlan 1" "ip address 192.168.10.2 255.255.255.192" (192.168.10.3 for S2 and 192.168.10.4 for S3) and then "ip default-gateway 192.168.10.1." 192.168.10.1 is the IP address assigned to Gi1 on R1.

The next configuration was to assign passwords to the privilege executive mode, the console, and VTY lines. The privilege executive mode password is

encrypted by default and is assigned via the command “enable secret ...” starting from global configuration mode. For security reasons, the passwords that are used for the devices will not be shown in this report. Next the console and VTY lines are assigned passwords starting from global configuration mode via the commands: “line console 0” (“line vty 0 4” for VTY lines), “password ...” “login.” This will assign the password that I have selected to use for the project; however, these two passwords are not encrypted by default therefore the command “service password-encryption” in global configuration mode. This command encrypts any plaintext passwords in the devices configuration files. Refer to Figure 3 below for an example of the encrypted passwords. Refer to Appendix B for all switch configurations.

Figure 3: Encrypted Passwords

```
banner motd ^CUnauthorized Access Prohibited^C
!
line con 0
  password 7 096F673A2616121C341C1625211B1B62362D1D
  login
line aux 0
line vty 0 4
  password 7 0802657D360A00192D1B1E0B2014177925213B
  login
!
!
end
S1#
```

Figure 3 is a screenshot of the encrypted passwords for the console and VTY lines on S1. These are encrypted via the global configuration command “service password-encryption.”

STP is the next protocol to configure on the switches. STP was configured on all three switches (S1, S2, and S3). This was configured by entering the global configuration command “spanning-tree vlan 1 root primary” on S3. This command initiates STP on the switch and designates S3 as the primary root bridge. This means that that all the network traffic in Subnet 1 will flow through S3 on its way to its destination. S1 and S2 were configured with the global configuration command “spanning-tree vlan 1 root secondary.” This initiates STP on those devices and states that they are not the primary root bridge. A default path for traffic to flow is then created on the three devices and BPDUs are sent out of all active interfaces on the switches. Because S1 and S2 are connected to end devices, they also need to have the commands: “switchport mode access” “spanning-tree portfast” and “spanning-tree bpduguard enable” entered on the interfaces that are connected to hosts. In this topology the interfaces connected to hosts are Gi0/0 on S1 and S2. Figure 4 below shows the STP configuration on S3.

Figure 4: STP Configuration

```
S3#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
            Address     0ce7.ed0f.0000
            This bridge is the root
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24577 (priority 24576 sys-id-ext 1)
            Address     0ce7.ed0f.0000
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time  300 sec

Interface    Role Sts Cost      Prio.Nbr Type
-----
Gi0/0        Desg FWD 4         128.1   P2p
Gi0/1        Desg FWD 4         128.2   P2p
Gi0/2        Desg FWD 4         128.3   P2p
```

Figure 4 shows a screenshot of the STP configuration for S3. The Root ID section of the output describes the primary root bridge while the Bridge ID section describes the switch the command is being run on (S3 in this case). As shown the MAC addresses for the root bridge and S3 are the same because S3 is the primary root bridge.

The final configurations to discuss on the switches are port security and DHCP snooping. Port security protects the network from STP and DHCP attacks because these attacks require the attacker to enter many spoofed MAC addresses. Port security was configured on S1 and S2. Port security is configured on each active interface that is connected to an end device. This includes the initiation and configuration of it. I attempted to configure port security on all active interfaces; however, this caused the network to completely lose functionality as the ports not connected to an end device would continuously shut down. Port security was thus configured on interface Gi0/0 on S1 and S2. The first step is to switch the port to access mode with the “switchport mode access” command. Then, set the maximum number of MAC addresses that can be saved on a port. In this case it was set to 2 via the “switchport port-security maximum 2” command. Next, initiate port security itself and set sticky MAC addresses to ensure that new devices cannot be added to the topology via this interface using the commands: “switchport port-security”; “switchport port-security mac-address sticky”; and “switchport port-security mac-address sticky 0800.2721.b1d0”; (different MAC address on S1). Finally, set the violation mode. The Gi0/0 port was set to shut down the port in the case of a violation via the command “switchport port-security violation shutdown.” The commands

shown here were used on S2. On S1, all of the commands are the same and on the same interface; however, the MAC address that was set to sticky is different as a different device is connected to S1. Finally, all unconnected ports were set to access mode and then issued the “shutdown” command on all three switches. This ensures a new device cannot connect to them without the network administrator enabling the interface. Figure 5 shows an example of port security. Refer to the Appendix B section for all port security configurations.

Figure 5: S2 Port Security

```
S2#sh port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
              (Count)          (Count)          (Count)
-----
          Gi0/0             2             0             11             Shutdown
-----
Total Addresses in System (excluding one mac per port)  : 0
Max Addresses limit in System (excluding one mac per port) : 4096
S2#
```

Figure 5 is a screenshot of the port security on S2. As shown in Figure 5 and described above there is a max number of addresses set to 2 and there have been 11 security violations. Also, the security action is set to shut down the port.

DHCP snooping was also configured on all three switches. This is to assist port security in protecting the network from a DHCP attack. To enable DHCP snooping I configured each of the three switches with the global configuration commands: “ip dhcp snooping”; “ip dhcp snooping vlan 1”; and “ip dhcp snooping verify mac-address.” These commands will enable dhcp snooping on the device and on the specified VLANs. The command “ip dhcp snooping verify mac-address” command

tells the switch to check the MAC address that is sending a DHCP Discover packet to the network. On S1 and S2 the commands “ip dhcp snooping trust” and “ip dhcp snooping check request message” were entered on interface Gi0/0. These two commands were entered on the interfaces Gi0/0, Gi0/1, and Gi0/2 on S3. These two commands will tell the DHCP snooping protocols to 1) trust the interface it is entered on, and 2) verify the DHCP Discover packet that is traveling through that specific interface. Figure 6 is an example of DHCP snooping. Refer to Appendix B for all DHCP snooping configurations.

Figure 6: S1 DHCP Snooping

```
S3#sh ip dhcp snooping
Switch DHCP snooping is enabled
Switch DHCP gleaning is disabled
DHCP snooping is configured on following VLANs:
1
DHCP snooping is operational on following VLANs:
1
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is enabled
  circuit-id default format: vlan-mod-port
  remote-id: 0ce7.ed0f.0000 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                Trusted    Allow option    Rate limit (pps)
-----                -
GigabitEthernet0/0      yes       yes              unlimited
  Custom circuit-ids:
GigabitEthernet0/1      yes       yes              unlimited
  Custom circuit-ids:
GigabitEthernet0/2      yes       yes              unlimited
Interface                Trusted    Allow option    Rate limit (pps)
-----                -
  Custom circuit-ids:
S3#
```

Figure 6 is a screenshot of the DHCP snooping on S3. As shown DHCP snooping is enabled on S3, and interfaces Gi0/0 – 2 are trusted ports. Also address verification is enabled.

The next configurations that were made on the network were the router configurations. First, the routers were configured with the hostnames R1, R2, and R3. R1 is the router connected to Subnet 1 via Gi1. R1 one interface Gi2 is connected to R2 Gi1. R2 Gi2 is connected to R3 Gi1. Finally, R3 is the router connected to Subnet 2 via interface Gi2. Each router was also assigned passwords following the same steps and commands as the switches. As stated before the passwords will not be listed in this report to keep the network devices secure.

The next step was to assign IP addresses. This is done in the same manner as on the switches however a default gateway can not be assigned on a router as the router acts as the default gateway for a LAN or subnet. R1 was assigned the IP address 192.168.10.1 on Gi1 with the subnet mask 255.255.255.192 and 192.168.10.129 on Gi2 with the subnet mask 255.255.255.252. On R2 Gi1 was assigned the IP address and subnet mask 192.168.10.130 255.255.255.252 and on Gi2 it was assigned 192.168.10.133 255.255.255.252. Finally, R3 was assigned the IP address 192.168.10.134 255.255.255.252 on Gi1 and the IP address 192.168.10.65 255.255.255.192 on Gi2. Interface Gi1 on R1 is the default gateway for Subnet 1, and interface Gi2 on R3 is the default gateway for Subnet 2. Refer to Appendix A section for the full IP addressing table.

The next router configuration was OSPF. OSPF needs to be configured on all three routers to function properly. OSPF functions differently than most routing protocols. Unlike configuring static and dynamic routes, OSPF is configured by assigning the networks that a router is directly connected to that router. Once all three routers are configured they will share their routing information with one another to automatically

create the routes between the networks. OSPF is configured using the global configuration command: "router ospf 10." This command creates the OSPF pool named "10." This is needed because the router will only share the information with other devices configured with this OSPF pool. Next the network information is added into the pool via the command "network 192.168.10.128 0.0.0.3 area 0." The command is entered for each network that the router is connected to. This is the command to add the network information on R1 for Subnet 3. "0.0.0.3" is the wildcard mask which is the opposite of the subnet mask. "255.255.255.252" is the subnet mask for Subnets 3 and 4 therefore the wildcard mask is "0.0.0.3." For Subnets 1 and 2 the subnet mask is "255.255.255.192" therefore, the wildcard mask is "0.0.0.63." Figure 7 shows an example of the OSPF configurations for R1. For all OSPF configurations refer to Appendix C.

Figure 7: OSPF

```
R1#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from Pfr

Gateway of last resort is not set

    192.168.10.0/24 is variably subnetted, 6 subnets, 3 masks
C       192.168.10.0/26 is directly connected, GigabitEthernet1
L       192.168.10.1/32 is directly connected, GigabitEthernet1
O       192.168.10.64/26
        [110/3] via 192.168.10.130, 03:54:24, GigabitEthernet2
C       192.168.10.128/30 is directly connected, GigabitEthernet2
L       192.168.10.129/32 is directly connected, GigabitEthernet2
O       192.168.10.132/30
        [110/2] via 192.168.10.130, 03:54:29, GigabitEthernet2
R1#
```

Figure 7 is a screenshot of the routing table for R1. All routes labeled with an “O” are OSPF routes. These routes are described in the paragraph above.

The final configuration that was made on the routers was configuring a DHCP server. DHCP is a protocol that assigns the devices on a network IP addresses from a specified pool of addresses. R2 was configured as the DHCP server and R1 is configured as a DHCP relay. To configure R2 as the DHCP server I entered the commands as follows. First, I needed to exclude the IP addresses used for the switches and R1. This was done via the global configuration command “ip dhcp excluded-address 192.168.10.1 192.168.10.10.” This excludes the IP addresses 192.168.10.1 to 192.168.10.10 from the pool of available IP addresses. This allows you to use these addresses as statically assigned IP addresses. Next the DHCP pool itself can be created. This is done with the global configuration commands: “ip dhcp pool R1_LAN” “network 192.168.10.0 255.255.255.192” “default-router

192.168.10.1” and “dns-server 8.8.8.8.” The first command created the DHCP pool “R1_LAN.” This is the DHCP pool for Subnet 1. The default router is set to the IP address of R1 Gi1 because this is the default gateway of Subnet 1. Next the DNS server is set to 8.8.8.8 which is the google DNS server. This creates the DHCP pool with the usable addresses of 192.168.10.11 to 192.168.10.62. The last step to configure and enable DHCP for Subnet 1 is to configure R1 as a DHCP relay. This will allow R1 to relay both DHCP Discover packets and DHCP leases to their destinations because R2 is not directly connected to Subnet 1. This is done using the command “ip helper-address 192.168.10.130.” Refer to Figure 8 for an example the DHCP Server information.

Figure 8: DHCP Server

```
R2#sh ip dhcp server
% Incomplete command.

R2#sh ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/      Lease expiration    Type      State      Interface
                Hardware address/
                User name
192.168.10.11   0100.5079.6668.00    Apr 01 2024 11:56 AM  Automatic Active      GigabitEthernet1
192.168.10.12   0108.0027.21b1.d0    Apr 01 2024 11:55 AM  Automatic Active      GigabitEthernet1
R2#sh ip dhcp pool

Pool R1_LAN :
Utilization mark (high/low) : 100 / 0
Subnet size (first/next)    : 0 / 0
Total addresses              : 62
Leased addresses             : 2
Excluded addresses          : 10
Pending event                : none
1 subnet is currently in the pool :
Current index      IP address range      Leased/Excluded/Total
192.168.10.12     192.168.10.1 - 192.168.10.62    2 / 10 / 62
R2#
```

Figure 8 is a screenshot of the DHCP server configuration on R2. This shows the leased addresses, as well as the DHCP pool information such as: total addresses, excluded addresses, and the network information.

Now that the network has been configured the final step of the configuration to complete before attacking the network was to configure the Ubuntu_Server. This entails configuring an IP address and a route to the default gateway for the Ubuntu_Server along with configuring the Apache2 and OpenSSH service. The IP address was configured by entering the command “sudo ip address add 192.168.10.66/26 dev enp0s3.” This address is assigned according to the IP addressing table in Appendix A and “enp0s3” is the interface that the IP address has been assigned on for the server. Next the route to the default gateway was added to the Ubuntu_Server via the command “sudo ip route add default via 192.168.10.65 dev enp0s3.” This command assigns the IP address 192.168.10.65 as the default gateway on the server interface “enp0s3.”

Next to configure the Apache2 service to configure the Ubuntu_Server as a web server, I had to create a new web page in the “/var/www/html” directory. To do this I created a directory named “/var/www/html/SportingGoodsStore.” Then I created HTML and CSS code files to create the web page itself and saved them in the “/var/www/html/SportingGoodsStore” directory. Then I entered the “sudo nano /etc/apache2/sites-enabled/000-default.conf” to edit the “000-default.conf” file to point the web page to the “/var/www/html/SportingGoodsStore” directory [24]. I then enabled HTTPS by entering the following configurations. First, I entered the command “sudo a2enmod ssl” to enable SSL on Apache2. Then I issued the command “sudo systemctl restart apache2” to restart the service. Next, I issued the command to generate a self-signed SSL certificate: “sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/server.ubuntuserver.com.key -

out `/etc/ssl/certs/server.ubuntu-server.com.crt`." Then I entered the country code "US," state name "Ohio," the city "Akron," and organization name "University of Akron." This created the certificate. Then, I edited the SSL configuration file via the command: `sudo nano /etc/apache2/sites-available/default-ssl.conf`." In this file I changed the `<VirtualHost *:80>` to `<VirtualHost *:443>` and then added the configuration lines: `SSLEngine on` `SSLCertificateKeyFile /etc/ssl/private/server.ubuntu-server.com.key` and `SSLCertificateFile /etc/ssl/certs/server.ubuntu-server.com.crt`." Finally, I enabled the "default-ssl.conf" file via the command `sudo a2ensite default-ssl.conf`," and then restarted Apache2 once more [17].

The OpenSSH service needed to also be configured on the Ubuntu_Server to enable the ability to establish an SSH connection to the Ubuntu_Server. To do this I edited the file: `/etc/ssh/sshd_config` via the command `sudo nano`." I then edited the file to allow root sign is by making the `PermitRootLogin` set to "yes." I set the `MaxTries` to 1. Then, I enabled passwords by setting `PasswordAuthentication` to "yes" and `PermitEmptyPasswords` to "no." I then restarted OpenSSH with the command `sudo systemctl restart sshd.service` [14].

Finally, I configured the server firewall to catch a brute force attack on SSH. I configured the firewall to create a log entry in the `/var/log/syslog` file and drop a connection if there are 4 failed logins in immediate succession. This was done by entering the commands: `iptables -N SSHATTACK`." This command creates a new rule chain. Then I entered `iptables -A SSHATTACK -j LOG --log-prefix "Possible SSH attack! " --log-level 7`" to create the rule to log the "Possible SSH attack." Then

I told the firewall to drop the connection via the command “iptables -A SSHATTACK -j DROP.” Finally, I set the interface and port that the rule is applied to with the command: “iptables -A INPUT -i enp0s3 -p tcp -m state --dport 22 --state NEW -m recent --set,” and then set the time the connection is blocked for to 120 seconds and the number of attempts to enact the rule to 4: “iptables -A INPUT -i enp0s3 -p tcp -m state --dport 22 --state NEW -m recent --update --seconds 120 --hitcount 4 -j SSHATTACK” [13]. Refer to Appendix D to see these commands.

Now that the configurations are complete, I could begin attacking the network. Each attack is being conducted from the LinuxVM using the software Yersinia and Hydra. The goal of this project is to prevent these attacks from being successful against my network topology. The first attack I ran was the STP man-in-the-middle attack via the software Yersinia. To initiate the attack, I entered the command “sudo yersinia -l” to enter the Yersinia software. By default, Yersinia is set to STP attacks. I then pressed the “x” key to open the exploits menu and selected the “Claiming Root Role” option by pressing “4.” This initiates the attack. Yersinia then begins sending BPDUs to the switches to initiate a re-election of the Root Bridge [5] and [7]. The results of the attack will be discussed in Chapter 4.

The second attack I ran was the DHCP Starvation attack. This attack is also run using the Yersinia software. A DHCP Starvation attack is when the attacker sends many DHCP Discover packets to the DHCP server from unique spoofed MAC addresses to cause the DHCP server to lease all available IP addresses to these fake MAC addresses. To run this attack, I entered the command “sudo yersinia -l” and opened the Yersinia software. I then had to press the “F2” key to enter the

DHCP attack mode. Next I pressed “x” to open the “Attack Panel” and selected the DHCP starvation attack by pressing the “1” key. This initiated Yersinia to begin spoofing MAC addresses and sending DHCP Discover packets to request an IP address from the DHCP server [7] and [9]. The results of this attack will be discussed in Chapter 4.

The final attack that I ran was the Brute Force attack on the SSH connection to the Ubuntu_Server. A Brute Force attack is when an attacker uses a software or malicious code to initiate sign-in attempts and guess passwords or encryption keys until the software guesses the password or key. This attack was conducted using the software Hydra. Hydra is a password cracking tool that can be used to initiate Brute Force attacks on the SSH protocol. To run this attack, I had to create a list of passwords on LinuxVM. To do this I created a text file with the 200 most popularly used passwords in the year 2023. I named this file “Password_List.” I then ran the Hydra software using the command “sudo hydra -l root -P ‘/home/kali/Desktop/Password_List’ 192.168.10.66 ssh.” This ran a Brute Force attack on the IP address 192.168.10.66 which is the Ubuntu_Server via the protocol SSH [22]. The results of this attack will be discussed in Chapter 4.

3.5 Conclusion

This chapter discusses the problems that I faced while completing this project and how I solved each problem. The problems included buying the Cisco Modeling Labs – Personal license to legally acquire the Cisco router and switch OS images, buying

new RAM cards for my laptop to effectively run the GNS-3 software and my topology, researching, and studying both the tactics and techniques used to secure the network and to attack it, and finally configuring and securing the network. Lastly, I had to attack the network to provide proof that the network has been hardened as I described. I discussed these issues and how they impact my project and their impact in the real world. I then explained the actual process that I used and the configurations that I had to make to complete this process and harden the network to the extent that I had stated in my project proposal. Finally, I explained the methodology in which I attacked the network and how these attacks work and affect a network when they are successful.

The next chapter will discuss the actual results of these attacks. I will also be analyzing the results that I have found from these attacks. The next chapter will provide the finalized proof that the network was adequately hardened to the extent that I had stated originally in my proposal and in this report.

Chapter 4: Results and Analysis

4.1 Introduction

The point of this chapter is to provide the results of each of the three attacks that I conducted on my network. These include STP man-in-the-middle, DHCP Starvation, and a Brute Force attack on an SSH connection to the Ubuntu_Server. I will also provide my analysis of these results and how they show the success of my senior project. The attacks were effectively and adequately prevented showing the network has been secured as stated in both my project proposal and this report.

4.2 Results and Analysis

The first attack that was conducted was the STP man-in-the-middle attack. For a complete description of how this attack was conducted please refer to section 3.4 of this report. An STP man-in-the-middle attack is when the attacker attempts to get his device to be elected as the Root Bridge of a network that is configured with redundancy in the switches. If this attack is successful the attacker will then have access to eavesdrop on all the traffic that is sent across the network because the switches will think that his device is the main pathway to the router and to the other end devices.

This attack was not successful in electing the LinuxVM as the root bridge of the network. Due to the port security that I configured the network was able to recognize that an attack was being conducted and a security violation was logged. In the event of a security violation the port that the LinuxVM was connected to was designed to shut down until I cleared the MAC address table and re-enabled the port manually. This means that as soon as the LinuxVM began sending the BPDU packets to the

network that BPDUguard is enabled on the interface Gi0/0 on S2 that the LinuxVM was connected to was shut down automatically. This completely removes the LinuxVM's ability to connect to and utilize the network in any way whatsoever. The security measures that I have configured on the network adequately and effectively mitigated and prevented the STP man-in-the-middle attack. Figure 9 shows the results of this attack. A video of the attack will also be shown in Appendix E.

Figure 9: STP Man-in-the-Middle

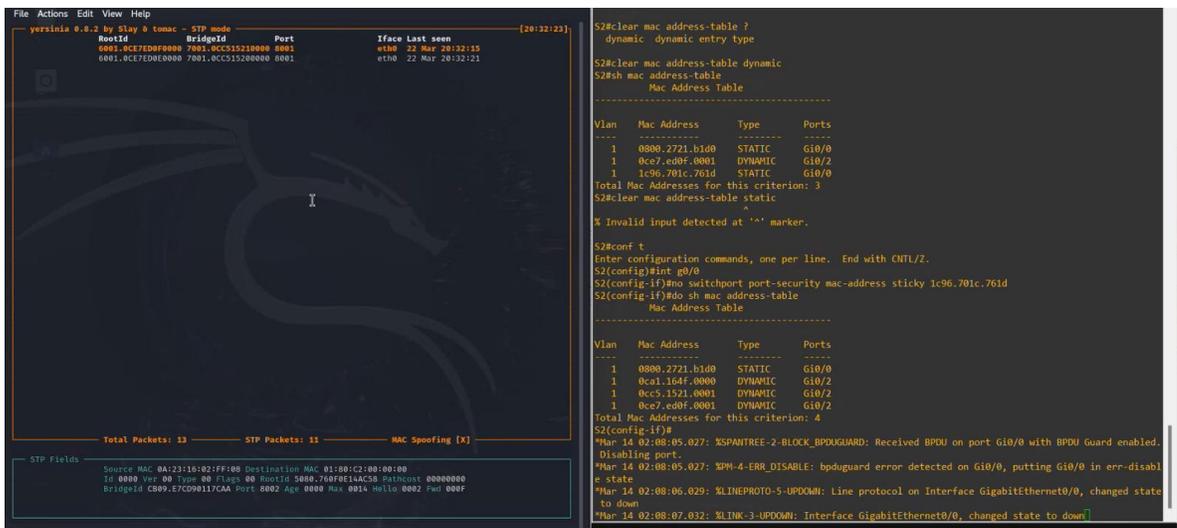


Figure 9 is a screenshot of the interface Gi0/0 on S2 being shut down in the event of the STP man-in-the-middle attack being conducted on the GNS-3 topology.

The second attack that was conducted was the DHCP Starvation attack. For a complete description of how this attack was conducted please refer to section 3.4 of this report. A DHCP Starvation attack is when the attacker sends an extremely large amount of DHCP Discovery packets to the DHCP server from unique spoofed MAC addresses. This will confuse the DHCP server into thinking that each Discovery packet is being sent from a unique device and it will issue that MAC

address an IP from the list of available IP addresses. This attack will cause the DHCP server to issue all of the available IP addresses, which will cause legitimate users to not be able to connect to the network because they will not be able to receive an IP address from the DHCP server.

This attack was also unsuccessful in fulfilling its purpose of leasing all available IP addresses. This is because of two configurations that I made on the network. First, DHCP snooping was enabled on the network which monitors for DHCP attacks. Second the port security that was configured on the network was set in a manner that as soon as more than 2 MAC addresses were seen and recorded on the switch interface, the interface was shut down. Interface Gi0/0 on S2 that is connected to the LinuxVM was immediately shut down when the attack started. This is shown in Figure 10 below and it adequately provides proof that the network was secured from DHCP attacks.

The Brute Force attack that was conducted on the SSH connection to the Ubuntu_Server was not successful. This is because as discussed in section 3.4, I was able to configure a firewall rule onto the Ubuntu_Server to block any connection that is being made from a single IP when there are more than 4 failed attempts. This blocks the connection for 120 seconds or 2 minutes. I also set the max number of attempts to 1 per SSH connection because most of the available Brute Force software will make one attempt per connection to attempt to circumvent firewall rules. The connection that was made with the Hydra software was blocked by my firewall rules and the Brute Force attempt timed out. This shows that the network was adequately hardened from Brute Force attacks on SSH connections. Figure 11 below will show proof of the attack failing.

Figure 11: Brute Force

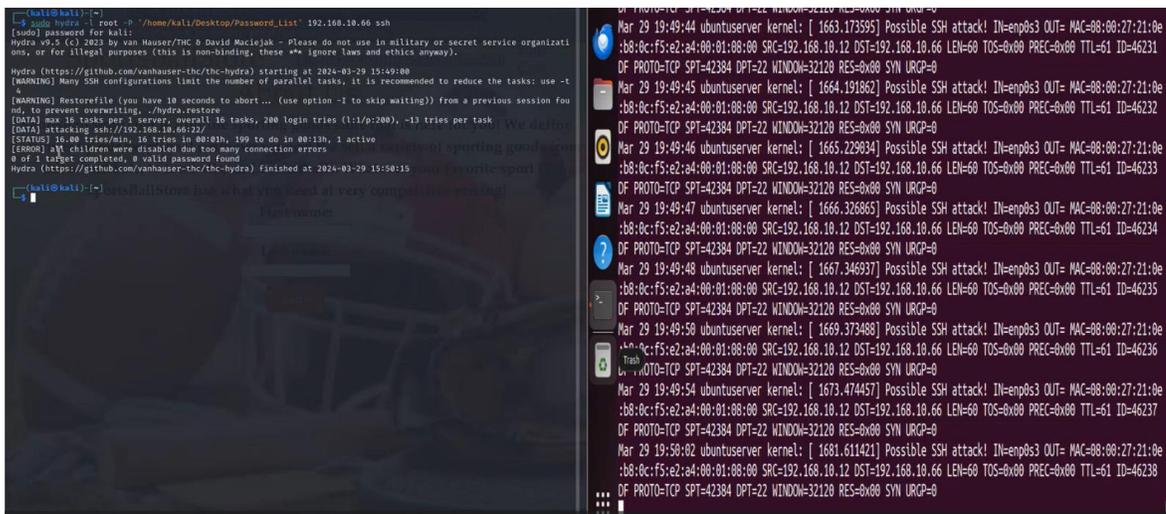


Figure 11 is a screenshot of the Brute Force attack that I conducted with the Hydra software timing out after being blocked by the firewall rules put in place on the

Ubuntu_Server. Also, it shows the attack being logged in the “/var/log/syslog” file on the Ubuntu_Server.

Chapter 5: Conclusions

5.1 Conclusion

My senior project is a culmination of all the knowledge and skills I have learned over the past four years at the University of Akron. In this workbook I have presented all the details that are included in my project.

During my project I conducted extensive research on the topics that are being covered. This includes the previous work and research by other people and organizations, the tactics, and techniques of the attacks that I am conducting, and the methods and configurations that can be put in place to mitigate the attacks. Configuring security from cyber-attacks into your network is called network hardening.

I have also created and configured a network with three cisco routers, three cisco switches, and three endpoints using the GNS-3 software. This network contains four subnets that have been created using VLSM. The network has been configured with DHCP, OSPF routing, and a web server. This network was then hardened with security features and protocols to protect it from STP, DHCP, and Brute Force attacks.

The final step of this project was to conduct an STP man-in-the-middle attack, DHCP Starvation attack, and a Brute Force attack on the network to see if they would be successful. Each attack was prevented by the security that I configured on the network. This proves that the network was hardened to protect the users of the network from being attacked by a threat actor.

5.2 Contributions

The contributions that I have made to this field of study are that I have provided proof of concept that a network can be protected from STP, DHCP, and Brute Force attacks. These are all very dangerous attacks if the network is not properly configured to protect against them. STP and DHCP attacks can result in the attacker gaining access to all the data being sent on the network, giving them the ability to eavesdrop on the data. This data, depending on the network, could be highly confidential including PII or even government data and secrets. They can also result in the function of the network being disrupted or even broken completely. This could cost an organization thousands or even millions of dollars to repair including the loss of production. Brute Force attacks can allow the attacker to gain access to accounts or services that cannot be open to the public. This could result in the attacker gaining access to a privileged account that can make high level changes or the ability to access classified or secret information from a business or government. It is imperative that a network be protected from Brute Force attacks by configuring both firewalls and setting strong passwords and encryption.

5.3 Future Work

There is plenty of work to still be done in this field and on this topic. This project shows security configuration and penetration testing on a very small scale. Cybersecurity is an ever-changing and evolving field. This means that soon the security measures that I put in place on this network may become obsolete and ineffective against new and other attacks alike. It is imperative that both myself and all others who are in the cybersecurity field continue to improve our understanding

and knowledge on the attacks that become possible and the ones that are possible today. This includes growing this knowledge and understanding on how to prevent attacks and mitigate the ones that cannot be prevented. In the future another may take the network that I have configured and expand upon both the scale and security on the network if they possess the knowledge and skill to do so.

References

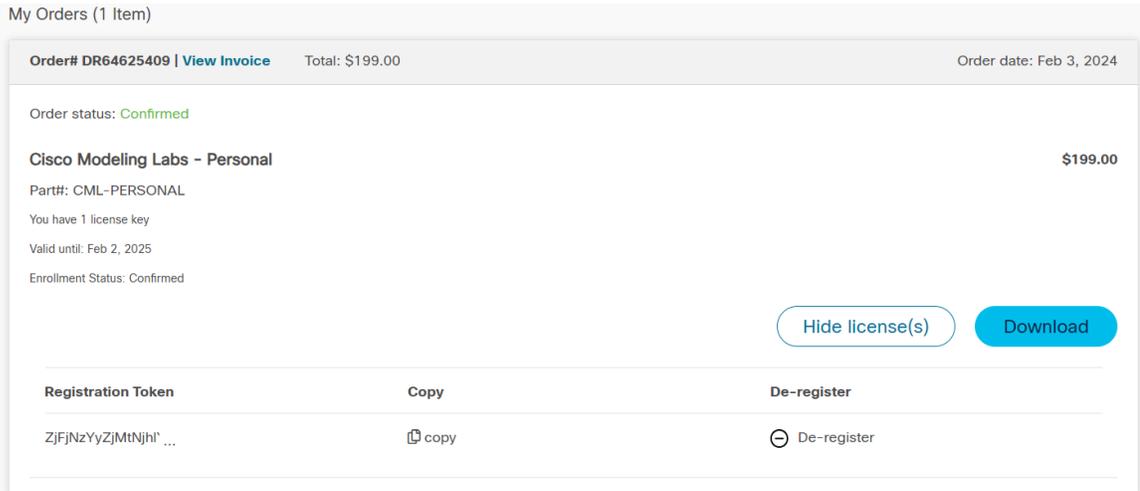
- [1] "Hardening network devices," Cybersecurity Information, https://media.defense.gov/2020/Aug/18/2002479461/-1/-1/0/HARDENING_NETWORK_DEVICES.PDF (accessed Mar. 28, 2024).
- [2] "What is an open port & what are the security...", What is an Open Port & What are the Security Implications?, <https://www.beyondtrust.com/blog/entry/what-is-an-open-port-what-are-the-security-implications> (accessed Mar. 28, 2024).
- [3] "What is systems hardening?," Systems Hardening, <https://www.beyondtrust.com/resources/glossary/systems-hardening#:~:text=Network%20hardening%3A%20Ensure%20your%20fire%20wall,access%20lists%3B%20encrypt%20network%20traffic.> (accessed Mar. 28, 2024).
- [4] "Cisco Guide to harden cisco IOS devices," Cisco, <https://www.cisco.com/c/en/us/support/docs/ip/access-lists/13608-21.html#anc41> (accessed Mar. 28, 2024).
- [5] Danielsekot, "Spanning tree protocol attacks: 3 attacks and defenses," Spanning Tree Protocol Attacks: 3 Attacks and Defenses, <https://www.prosec-networks.com/en/blog/spanning-tree-protokoll-angriffe-3-attacks-und-schutzmassnahmen/#:~:text=Spanning%20Tree%20Protocol%20Attack%2003%3A%20STP%20Man%20in%20the%20Middle,-By%20doing%20the&text=In%20the%20attack%20menu%20in,that%20the%20computer%20will%20crash.> (accessed Mar. 28, 2024).
- [6] "What is penetration testing?," What is Penetration Testing? | IBM, <https://www.ibm.com/topics/penetration-testing> (accessed Mar. 28, 2024).
- [7] "Yersinia: Kali linux tools," yersinia | Kali Linux Tools, <https://www.kali.org/tools/yersinia/> (accessed Mar. 28, 2024).
- [8] "Security and Privacy Controls for Information Systems and Organizations ," NIST Special Publication 800-53 Revision 5 , <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf> (accessed Mar. 28, 2024).
- [9] R. Sankar, "Kali Linux Tutorials," Yersinia for layer 2 - vulnerability analysis & DHCP starvation attack, <https://kalilinuxtutorials.com/yersinia/> (accessed Mar. 28, 2024).

- [10] “2. Bringing interfaces up/down,” 2. bringing interfaces up/down, <https://tldp.org/HOWTO/Linux+IPv6-HOWTO/ch05s02.html> (accessed Mar. 28, 2024).
- [11] M. Aleksic, “How to install a desktop (GUI) on an ubuntu server,” Knowledge Base by phoenixNAP, <https://phoenixnap.com/kb/how-to-install-a-gui-on-ubuntu> (accessed Mar. 28, 2024).
- [12] “Configuring DHCP Snooping,” Security - configuring DHCP snooping [support], https://www.cisco.com/en/US/docs/general/Test/dwerblo/broken_guide/snoodhcp.html (accessed Mar. 28, 2024).
- [13] “SSH / SSHD - how do I set Max Login attempts?,” Server Fault, <https://serverfault.com/questions/275669/ssh-sshd-how-do-i-set-max-login-attempts> (accessed Mar. 28, 2024).
- [14] “OpenSSH Server,” OpenSSH server | Ubuntu, <https://ubuntu.com/server/docs/service-openssh> (accessed Mar. 28, 2024).
- [15] V. Gite, “Debian / Ubuntu Linux Setting a Default Gateway,” Debian / Ubuntu Linux Setting a Default Gateway - nixCraft, <https://www.cyberciti.biz/faq/howto-debian-ubutu-set-default-gateway-ipaddress/> (accessed Mar. 28, 2024).
- [16] J. Carson, Guide to Network Security and Hardening, <https://delinea.com/blog/network-security-and-hardening> (accessed Mar. 28, 2024).
- [17] “What is HTTPS?,” What is HTTPS? | Cloudflare, <https://www.cloudflare.com/learning/ssl/what-is-https/> (accessed Mar. 28, 2024).
- [18] “Penetration Testing and Ethical Hacking Linux Distribution,” Kali Linux, <https://www.kali.org/> (accessed Mar. 28, 2024).
- [19] “Enterprise Open Source and Linux,” Ubuntu, <https://ubuntu.com/> (accessed Mar. 28, 2024).
- [20] “Mitre ATT&CK®,” MITRE ATT&CK®, <https://attack.mitre.org/#> (accessed Mar. 29, 2024).
- [21] Cisco Modeling Labs, <https://www.cisco.com/c/en/us/products/cloud-systems-management/modeling-labs/index.html> (accessed Mar. 29, 2024).
- [22] “Hydra Documentation,” Hydra: Kali linux tools, <https://www.kali.org/tools/hydra/> (accessed Mar. 29, 2024).
- [23] “GNS-3,” Gns3.com, <https://www.gns3.com/> (accessed Mar. 30, 2024).
- [24] D. Group, “Essentials - Apache,” Welcome! - The Apache HTTP Server Project, <https://httpd.apache.org/> (accessed Mar. 30, 2024).

Appendix

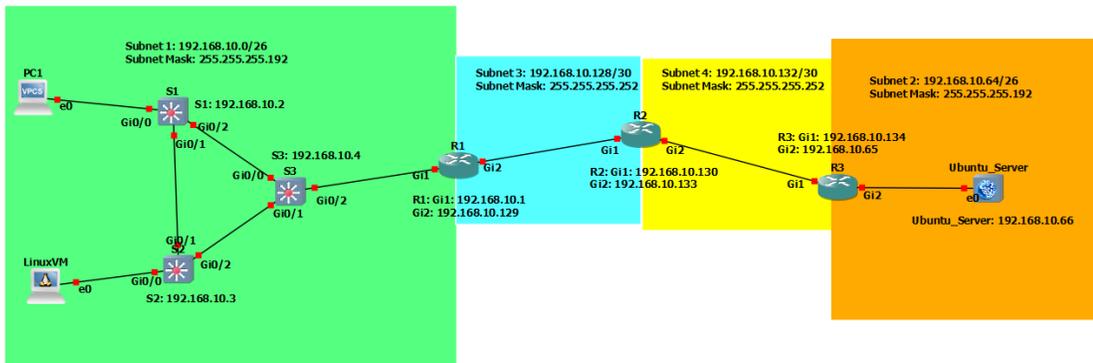
Appendix A: General Figures

Appendix A.1: Cisco Modeling Labs – Personal License



Appendix A.1 is a screenshot of the Cisco Modeling Labs – Personal license that I purchased for \$199.00 USD.

Appendix A.2: GNS-3 Topology



Appendix A.2 is a screenshot of the topology created for this project. It contains three routers, three switches, and three end devices.

Appendix A.3: Full VLSM Table

Device Hostname	Interface	Subnet Number	Network Number	IP Address	Subnet Mask	Default Gateway
R1	Gi1	Subnet 1	192.168.10.0	192.168.10.1	255.255.255.192	X
	Gi2	Subnet 3	192.168.10.128	192.168.10.129	255.255.255.252	X
R2	Gi1	Subnet 3	192.168.10.128	192.168.10.130	255.255.255.252	X
	Gi2	Subnet 4	192.168.10.132	192.168.10.133	255.255.255.252	X
R3	Gi1	Subnet 4	192.168.10.132	192.168.10.134	255.255.255.252	X
	Gi2	Subnet 2	192.168.10.64	192.168.10.65	255.255.255.192	X
S1	Vlan 1	Subnet 1	192.168.10.0	192.168.10.2	255.255.255.192	192.168.10.1
S2	Vlan 1	Subnet 1	192.168.10.0	192.168.10.3	255.255.255.192	192.168.10.1
S3	Vlan 1	Subnet 1	192.168.10.0	192.168.10.4	255.255.255.192	192.168.10.1
PC1	e0	Subnet 1	192.168.10.0	DHCP Assigned	255.255.255.192	192.168.10.1
LinuxVM	e0	Subnet 1	192.168.10.0	DHCP Assigned	255.255.255.192	192.168.10.1
Ubuntu_Server	e0	Subnet 2	192.168.10.64	192.168.10.66	255.255.255.192	192.168.10.65

Appendix A.3 is a table that shows the full VLSM table created for my Senior Project. This table includes each router interface, each switch, and each end device on the network.

Appendix B: Switch Configurations

Appendix B.1: S1 Running Configuration 1

```
S1#sh run
Building configuration...

Current configuration : 4559 bytes
!
! Last configuration change at 00:46:25 UTC Thu Mar 14 2024
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
service compress-config
!
hostname S1
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$Ggt5$QT6NE6jGHQR/4s1SPjNrB1
!
```

Appendix B.1 is a screenshot of part 1 of the “show run” command output on S1. This shows the encryption of passwords, the hostname, and the hash of the privilege executive password.

Appendix B.2: S1 Running Configuration 2

```
ip dhcp snooping vlan 1
ip dhcp snooping
no ip domain-lookup
ip cef
no ipv6 cef
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
spanning-tree vlan 1 priority 28672
!
```

Appendix B.2 is a screenshot of part 2 of the “show run” command output on S1. This part shows that DHCP snooping is enabled on VLAN 1, DNS lookup is disabled, and that STP is enabled.

Appendix B.3: S1 Running Configuration 3

```
interface GigabitEthernet0/0
  switchport mode access
  switchport port-security maximum 2
  switchport port-security mac-address sticky
  switchport port-security mac-address sticky 0050.7966.6800
  switchport port-security
  negotiation auto
  spanning-tree portfast edge
  spanning-tree bpduguard enable
  ip dhcp snooping trust
  !
interface GigabitEthernet0/1
  switchport mode access
  switchport port-security maximum 2
  negotiation auto
  !
interface GigabitEthernet0/2
  switchport mode access
  switchport port-security maximum 2
  negotiation auto
  ip dhcp snooping trust
```

Appendix B.3 is a screenshot of part 3 of the “show run” command output on S1. This part shows the configuration of the 3 active interfaces on S1 Gi0/0 – 2.

Appendix B.4: S1 Running Configuration 4

```
interface Vlan1
  ip address 192.168.10.2 255.255.255.192
  shutdown
  !
ip default-gateway 192.168.10.1
ip forward-protocol nd
!
ip http server
!
ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr
ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr
!
```

Appendix B.4 is a screenshot of part 4 of the “show run” command output on S1. This part shows the configuration of VLAN 1 including the IP address and Default gateway.

Appendix B.5: S1 Running Configuration 5

```
banner motd ^CUnauthorized Access Prohibited^C
!
line con 0
 password 7 096F673A2616121C341C1625211B1B62362D1D
 login
line aux 0
line vty 0 4
 password 7 0802657D360A00192D1B1E0B2014177925213B
 login
!
!
end
S1#
```

Appendix B.5 is a screenshot of part 5 of the “show run” command output on S1. This part shows the banner that I configured and the encrypted password hashes for the console and VTY lines.

Appendix B.6: S1 IP Configuration

```
S1#sh ip int brief
Interface          IP-Address      OK? Method Status          Protocol
GigabitEthernet0/0 unassigned      YES unset  up              up
GigabitEthernet0/1 unassigned      YES unset  up              up
GigabitEthernet0/2 unassigned      YES unset  up              up
GigabitEthernet0/3 unassigned      YES unset  administratively down down
GigabitEthernet1/0 unassigned      YES unset  administratively down down
GigabitEthernet1/1 unassigned      YES unset  administratively down down
GigabitEthernet1/2 unassigned      YES unset  administratively down down
GigabitEthernet1/3 unassigned      YES unset  administratively down down
GigabitEthernet2/0 unassigned      YES unset  administratively down down
GigabitEthernet2/1 unassigned      YES unset  administratively down down
GigabitEthernet2/2 unassigned      YES unset  administratively down down
GigabitEthernet2/3 unassigned      YES unset  administratively down down
GigabitEthernet3/0 unassigned      YES unset  administratively down down
GigabitEthernet3/1 unassigned      YES unset  administratively down down
GigabitEthernet3/2 unassigned      YES unset  administratively down down
GigabitEthernet3/3 unassigned      YES unset  administratively down down
Vlan1              192.168.10.2   YES NVRAM  administratively down down
S1#
```

Appendix B.6 is a screenshot of the IP interface information on S1. As shown all inactive ports are shut down and the IP address is assigned to VLAN 1.

Appendix B.7: S1 STP Configuration

```
S1#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
            Address    0ce7.ed0f.0000
            Cost      4
            Port      3 (GigabitEthernet0/2)
            Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    28673 (priority 28672 sys-id-ext 1)
            Address    0c5d.4603.0000
            Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time 300 sec

Interface          Role Sts Cost      Prio.Nbr Type
-----
Gi0/0              Desg FWD 4         128.1   P2p Edge
Gi0/1              Desg FWD 4         128.2   P2p
Gi0/2              Root FWD 4         128.3   P2p

S1#
```

Appendix B.7 is a screenshot of the STP configuration for S1. This shows the Root ID which is the information that is recorded about the Primary Root Bridge and the Bridge ID that is recorded about S1.

Appendix B.8: S1 DHCP Snooping Configuration

```
S1#sh ip dhcp snooping
Switch DHCP snooping is enabled
Switch DHCP gleaning is disabled
DHCP snooping is configured on following VLANs:
1
DHCP snooping is operational on following VLANs:
1
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is enabled
  circuit-id default format: vlan-mod-port
  remote-id: 0c5d.4603.0000 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                Trusted    Allow option    Rate limit (pps)
-----                -
GigabitEthernet0/0      yes       yes             unlimited
  Custom circuit-ids:
GigabitEthernet0/2      yes       yes             unlimited
  Custom circuit-ids:
S1#
```

Appendix B.8 shows a screenshot of the DHCP snooping Configuration for S1. This shows that DHCP snooping is enabled on the device and VLAN 1 and that verification is enabled. It also shows what interfaces are trusted on S1.

Appendix B.9: S1 Port Security Configuration

```
S1#sh port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
              (Count)          (Count)          (Count)
-----
          Gi0/0             2             1             0             Shutdown
-----
Total Addresses in System (excluding one mac per port)    : 0
Max Addresses limit in System (excluding one mac per port) : 4096
S1#
```

Appendix B.9 shows the Port Security Configuration for S1. This shows the interfaces it is configured on, the number of MAC addresses that can be recorded, the current number of addresses recorded, the violation count, and the violation action.

Appendix B.10: S2 Running Configuration 1

```
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
service compress-config
!
hostname S2
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$G.SZ$NSK30Axxnn5nrJXAeMixg1
!
```

Appendix B.10 is a screenshot of part 1 of the “show run” command output on S2. This shows the encryption of passwords, the hostname, and the hash of the privilege executive password.

Appendix B.11: S2 Running Configuration 2

```
ip dhcp snooping vlan 1
ip dhcp snooping
no ip domain-lookup
ip cef
no ipv6 cef
!
!
!
spanning-tree mode pvst
spanning-tree extend system-id
spanning-tree vlan 1 priority 28672
!
```

Appendix B.11 is a screenshot of part 2 of the “show run” command output on S2. This part shows that DHCP snooping is enabled on VLAN 1, DNS lookup is disabled, and that STP is enabled.

Appendix B.12: S2 Running Configuration 3

```
interface GigabitEthernet0/0
  switchport mode access
  switchport port-security maximum 2
  switchport port-security mac-address sticky
  switchport port-security mac-address sticky 0800.2721.b1d0
  switchport port-security
  negotiation auto
  spanning-tree portfast edge
  spanning-tree bpduguard enable
  ip dhcp snooping trust
  !
interface GigabitEthernet0/1
  negotiation auto
  !
interface GigabitEthernet0/2
  negotiation auto
  ip dhcp snooping trust
```

Appendix B.12 is a screenshot of part 3 of the “show run” command output on S2. This part shows the configuration of the 3 active interfaces on S1 Gi0/0 – 2.

Appendix B.13: S2 Running Configuration 4

```
interface Vlan1
  ip address 192.168.10.3 255.255.255.192
  shutdown
  !
ip default-gateway 192.168.10.1
ip forward-protocol nd
  !
ip http server
  !
ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr
ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr
```

Appendix B.13 is a screenshot of part 4 of the “show run” command output on S2. This part shows the configuration of VLAN 1 including the IP address and Default gateway.

Appendix B.14: S2 Running Configuration 5

```
banner motd ^CUnauthorized Access Prohibited^C
!  
line con 0  
  password 7 112A303628010E023B3A392B220C0670100818  
  login  
line aux 0  
line vty 0 4  
  password 7 04782235303249403609171818343F563C3F3D  
  login  
!  
!  
end
```

Appendix B.14 is a screenshot of part 5 of the “show run” command output on S2. This part shows the banner that I configured and the encrypted password hashes for the console and VTY lines.

Appendix B.15: S2 IP Configurations

```
S2#sh ip int brief
Interface                IP-Address      OK? Method Status      Protocol
GigabitEthernet0/0      unassigned      YES unset    up          up
GigabitEthernet0/1      unassigned      YES unset    up          up
GigabitEthernet0/2      unassigned      YES unset    up          up
GigabitEthernet0/3      unassigned      YES unset    administratively down down
GigabitEthernet1/0      unassigned      YES unset    administratively down down
GigabitEthernet1/1      unassigned      YES unset    administratively down down
GigabitEthernet1/2      unassigned      YES unset    administratively down down
GigabitEthernet1/3      unassigned      YES unset    administratively down down
GigabitEthernet2/0      unassigned      YES unset    administratively down down
GigabitEthernet2/1      unassigned      YES unset    administratively down down
GigabitEthernet2/2      unassigned      YES unset    administratively down down
GigabitEthernet2/3      unassigned      YES unset    administratively down down
GigabitEthernet3/0      unassigned      YES unset    administratively down down
GigabitEthernet3/1      unassigned      YES unset    administratively down down
GigabitEthernet3/2      unassigned      YES unset    administratively down down
GigabitEthernet3/3      unassigned      YES unset    administratively down down
Vlan1                    192.168.10.3    YES NVRAM    administratively down down
S2#
```

Appendix B.15 is a screenshot of the IP interface information on S2. As shown all inactive ports are shut down and the IP address is assigned to VLAN 1.

Appendix B.16: S2 STP Configurations

```
S2#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
             Address    0ce7.ed0f.0000
             Cost        4
             Port        3 (GigabitEthernet0/2)
             Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec

  Bridge ID  Priority    28673 (priority 28672 sys-id-ext 1)
             Address    0cc5.1521.0000
             Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
             Aging Time  300 sec

Interface                Role Sts Cost        Prio.Nbr Type
-----
Gi0/0                    Desg FWD 4           128.1    P2p Edge
Gi0/1                    Altn BLK 4           128.2    P2p
Gi0/2                    Root FWD 4           128.3    P2p
```

Appendix B.16 is a screenshot of the STP configuration for S2. This shows the Root ID which is the information that is recorded about the Primary Root Bridge and the Bridge ID that is recorded about S2.

Appendix B.17: S2 DHCP Snooping Configurations

```
S2#sh ip dhcp snooping
Switch DHCP snooping is enabled
Switch DHCP gleaning is disabled
DHCP snooping is configured on following VLANs:
1
DHCP snooping is operational on following VLANs:
1
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is enabled
  circuit-id default format: vlan-mod-port
  remote-id: 0cc5.1521.0000 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                Trusted    Allow option    Rate limit (pps)
-----                -
GigabitEthernet0/0      yes       yes             unlimited
  Custom circuit-ids:
GigabitEthernet0/2      yes       yes             unlimited
  Custom circuit-ids:
S2#
```

Appendix B.17 shows a screenshot of the DHCP snooping Configuration for S2. This shows that DHCP snooping is enabled on the device and VLAN 1 and that verification is enabled. It also shows what interfaces are trusted on S2.

Appendix B.18: S2 Port Security Configurations

```
S2#sh port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
              (Count)      (Count)      (Count)
-----
          Gi0/0           2           0           11           Shutdown
-----
Total Addresses in System (excluding one mac per port)    : 0
Max Addresses limit in System (excluding one mac per port) : 4096
S2#
```

Appendix B.18 shows the Port Security Configuration for S2. This shows the interfaces it is configured on, the number of MAC addresses that can be recorded, the current number of addresses recorded, the violation count, and the violation action.

Appendix B.20: S3 Running Configuration 2

```
interface Vlan1
  ip address 192.168.10.4 255.255.255.192
  shutdown
!
ip default-gateway 192.168.10.1
ip forward-protocol nd
!
ip http server
!
ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr
ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr
```

Appendix B.20 is a screenshot of part 2 of the “show run” command issued on S3.

This shows the VLAN 1 configurations including IP address and default gateway.

Appendix B.21: S3 Running Configuration 3

```
banner motd ^CUnauthorized Access Prohibited^C
!
line con 0
  password 7 03277238391C244271190B0A1D2D385F072525
  login
line aux 0
line vty 0 4
  password 7 13263E21341F012414343A3C3F1D2054001718
  login
!
!
end
S3#
```

Appendix B.21 is a screenshot of part 3 of the “show run” command output on S3.

This part shows the banner that I configured and the encrypted password hashes for the console and VTY lines.

Appendix B.22: S3 IP Configurations

```
S3#sh ip int brief
Interface          IP-Address      OK? Method Status        Protocol
GigabitEthernet0/0 unassigned      YES unset  up            up
GigabitEthernet0/1 unassigned      YES unset  up            up
GigabitEthernet0/2 unassigned      YES unset  up            up
GigabitEthernet0/3 unassigned      YES unset  administratively down down
GigabitEthernet1/0 unassigned      YES unset  administratively down down
GigabitEthernet1/1 unassigned      YES unset  administratively down down
GigabitEthernet1/2 unassigned      YES unset  administratively down down
GigabitEthernet1/3 unassigned      YES unset  administratively down down
GigabitEthernet2/0 unassigned      YES unset  administratively down down
GigabitEthernet2/1 unassigned      YES unset  administratively down down
GigabitEthernet2/2 unassigned      YES unset  administratively down down
GigabitEthernet2/3 unassigned      YES unset  administratively down down
GigabitEthernet3/0 unassigned      YES unset  administratively down down
GigabitEthernet3/1 unassigned      YES unset  administratively down down
GigabitEthernet3/2 unassigned      YES unset  administratively down down
GigabitEthernet3/3 unassigned      YES unset  administratively down down
Vlan1              192.168.10.4   YES NVRAM  administratively down down
```

Appendix B.22 is a screenshot of the IP interface information on S3. As shown all inactive ports are shut down and the IP address is assigned to VLAN 1.

Appendix B.23: S3 STP Configurations

```
S3#sh spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
             Address     0ce7.ed0f.0000
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24577 (priority 24576 sys-id-ext 1)
             Address     0ce7.ed0f.0000
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface          Role Sts Cost          Prio.Nbr Type
-----
Gi0/0              Desg FWD 4           128.1    P2p
Gi0/1              Desg FWD 4           128.2    P2p
Gi0/2              Desg FWD 4           128.3    P2p
```

Appendix B.23 is a screenshot of the STP configuration for S3. This shows the Root ID which is the information that is recorded about the Primary Root Bridge

and the Bridge ID that is recorded about S3. The information is the same because S3 is the Primary Root Bridge.

Appendix B.24: S3 DHCP Snooping Configurations

```
S3#sh ip dhcp snooping
Switch DHCP snooping is enabled
Switch DHCP gleaning is disabled
DHCP snooping is configured on following VLANs:
1
DHCP snooping is operational on following VLANs:
1
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is enabled
  circuit-id default format: vlan-mod-port
  remote-id: 0ce7.ed0f.0000 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                Trusted    Allow option    Rate limit (pps)
-----                -
GigabitEthernet0/0      yes       yes             unlimited
  Custom circuit-ids:
GigabitEthernet0/1      yes       yes             unlimited
  Custom circuit-ids:
GigabitEthernet0/2      yes       yes             unlimited
Interface                Trusted    Allow option    Rate limit (pps)
-----                -
  Custom circuit-ids:
S3#
```

Appendix B.24 shows a screenshot of the DHCP snooping Configuration for S3. This shows that DHCP snooping is enabled on the device and VLAN 1 and that verification is enabled. It also shows what interfaces are trusted on S3.

Appendix C: Router Configurations

Appendix C.1: R1 Running Configuration 1

```
version 16.6
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$NGg/$3aeZCMeV/R8kijXNsByCS1
!
```

Appendix C.1 is a screenshot of part 1 of the “show run” command output on R1. This shows the encryption of passwords, the hostname, and the hash of the privilege executive password. This also shows that DHCP snooping is enabled, and that DNS lookup is disabled.

Appendix C.2: R1 Running Configuration 2

```
router ospf 10
 network 192.168.10.0 0.0.0.63 area 0
 network 192.168.10.128 0.0.0.3 area 0
 !
 !
 virtual-service csr_mgmt
 !
 ip forward-protocol nd
 ip http server
 ip http authentication local
 ip http secure-server
 !
 ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr
 ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr
 !
 !
 !
 !
 control-plane
 !
 !
 !
 !
 banner motd ^CUnauthorized Access Prohibited^C
 !
 line con 0
 password 7 123A2C242D18090A153B3627390A104204190D
 login
 stopbits 1
 line vty 0 4
 password 7 00273A353B480E0830315E4103263746041F15
 login
```

Appendix C.2 is a screenshot of part 2 of the “show run” command output on R1.

This shows the OSPF configuration, the banner I configured, and the password hashes for the console and VTY lines.

Appendix C.3: R1 IP Configurations

```
R1#sh ip int brief
Interface          IP-Address      OK? Method Status      Protocol
GigabitEthernet1  192.168.10.1    YES NVRAM  up          up
GigabitEthernet2  192.168.10.129 YES NVRAM  up          up
GigabitEthernet3  unassigned      YES NVRAM  administratively down down
GigabitEthernet4  unassigned      YES NVRAM  administratively down down
R1#
```

Appendix C.3 is a screenshot of the IP interface configuration for R1. This shows the interfaces that are active and their IP addresses.

Appendix C.4: R1 Routing Configurations

```
R1#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

    192.168.10.0/24 is variably subnetted, 6 subnets, 3 masks
C       192.168.10.0/26 is directly connected, GigabitEthernet1
L       192.168.10.1/32 is directly connected, GigabitEthernet1
O       192.168.10.64/26
        [110/3] via 192.168.10.130, 03:54:24, GigabitEthernet2
C       192.168.10.128/30 is directly connected, GigabitEthernet2
L       192.168.10.129/32 is directly connected, GigabitEthernet2
O       192.168.10.132/30
        [110/2] via 192.168.10.130, 03:54:29, GigabitEthernet2
R1#
```

Appendix C.4 shows a screenshot of the routing configuration for R1. All the routes that are labeled with an “O” are the OSPF routes that I configured. The routes labeled “C” are directly connected networks, and the routes labeled “L” are Loopback interfaces.

Appendix C.6: R2 Running Configuration 2

```
ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr
ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr
!
!
!
!
!
control-plane
!
!
!
!
!
banner motd ^CUnauthorized Access Prohibited^C
!
line con 0
 password 7 00273A353B480E0830315E4103263745110402
 login
 stopbits 1
line vty 0 4
 password 7 02252D6834150A2F735E1B160F2820591A1033
 login
!
!
!
!
!
end
R2#
```

Appendix C.6 is a screenshot of part 2 of the “show run” command output on R2. This shows the OSPF configuration, the banner I configured, and the password hashes for the console and VTY lines.

Appendix C.7: R2 IP Configurations

```
R2#sh ip int brief
Interface          IP-Address      OK? Method Status      Protocol
GigabitEthernet1  192.168.10.130 YES NVRAM    up          up
GigabitEthernet2  192.168.10.133 YES NVRAM    up          up
GigabitEthernet3  unassigned     YES NVRAM    administratively down down
GigabitEthernet4  unassigned     YES NVRAM    administratively down down
R2#
```

Appendix C.7 is a screenshot of the IP interface configuration for R2. This shows the interfaces that are active and their IP addresses.

Appendix C.8: R2 Routing Configurations

```
R2#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

  192.168.10.0/24 is variably subnetted, 6 subnets, 3 masks
O       192.168.10.0/26
        [110/2] via 192.168.10.129, 03:44:04, GigabitEthernet1
O       192.168.10.64/26
        [110/2] via 192.168.10.134, 03:44:02, GigabitEthernet2
C       192.168.10.128/30 is directly connected, GigabitEthernet1
L       192.168.10.130/32 is directly connected, GigabitEthernet1
C       192.168.10.132/30 is directly connected, GigabitEthernet2
L       192.168.10.133/32 is directly connected, GigabitEthernet2
R2#
```

Appendix C.8 shows a screenshot of the routing configuration for R2. All the routes that are labeled with an “O” are the OSPF routes that I configured. The routes labeled “C” are directly connected networks, and the routes labeled “L” are Loopback interfaces.

Appendix C.9: R2 DHCP Server Configurations

```
R2#sh ip dhcp server
% Incomplete command.

R2#sh ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/      Lease expiration    Type      State      Interface
                Hardware address/
                User name
192.168.10.11   0100.5079.6668.00    Apr 01 2024 11:56 AM    Automatic    Active      GigabitEthernet1
192.168.10.12   0108.0027.21b1.d0    Apr 01 2024 11:55 AM    Automatic    Active      GigabitEthernet1
R2#sh ip dhcp pool

Pool R1_LAN :
Utilization mark (high/low) : 100 / 0
Subnet size (first/next)     : 0 / 0
Total addresses               : 62
Leased addresses              : 2
Excluded addresses            : 10
Pending event                  : none
1 subnet is currently in the pool :
Current index      IP address range      Leased/Excluded/Total
192.168.10.12     192.168.10.1 - 192.168.10.62    2 / 10 / 62
R2#
```

Appendix C.9 is a screenshot of the DHCP Server configuration. This shows the leased IP addresses as well as the DHCP pool information. This includes the total addresses, the number of excluded addresses, the number of leased addresses, the network address, and address range.

Appendix C.10: R3 Running Configuration 1

```
version 16.6
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console serial
!
hostname R3
!
boot-start-marker
boot-end-marker
!
!
enable secret 5 $1$oDjH$pdCY2m7dhU06akrC41uN7/
!
```

Appendix C.10 is a screenshot of part 1 of the “show run” command output on R3. This shows the encryption of passwords, the hostname, and the hash of the privilege executive password. This also shows that DHCP snooping is enabled, and that DNS lookup is disabled.

Appendix C.12: R3 IP Configurations

```
R3#sh ip int brief
Interface          IP-Address      OK? Method Status          Protocol
GigabitEthernet1  192.168.10.134 YES NVRAM   up              up
GigabitEthernet2  192.168.10.65  YES NVRAM   up              up
GigabitEthernet3  unassigned      YES NVRAM   administratively down down
GigabitEthernet4  unassigned      YES NVRAM   administratively down down
R3#
```

Appendix C.12 is a screenshot of the IP interface configuration for R3. This shows the interfaces that are active and their IP addresses.

Appendix C.13: Routing Configurations

```
R3#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

  192.168.10.0/24 is variably subnetted, 6 subnets, 3 masks
O       192.168.10.0/26
        [110/3] via 192.168.10.133, 03:50:20, GigabitEthernet1
C       192.168.10.64/26 is directly connected, GigabitEthernet2
L       192.168.10.65/32 is directly connected, GigabitEthernet2
O       192.168.10.128/30
        [110/2] via 192.168.10.133, 03:50:20, GigabitEthernet1
C       192.168.10.132/30 is directly connected, GigabitEthernet1
L       192.168.10.134/32 is directly connected, GigabitEthernet1
R3#
```

Appendix C.13 shows a screenshot of the routing configuration for R3. All the routes that are labeled with an “O” are the OSPF routes that I configured. The routes labeled “C” are directly connected networks, and the routes labeled “L” are Loopback interfaces.

Appendix D: Ubuntu Server Configuration

Appendix D.1: Server IP and Firewall Configuration

```
ubuntu@ubuntu-server: /etc/ssh$ sudo iptables -N SSHATTACK
ubuntu@ubuntu-server: /etc/ssh$ sudo iptables -A SSHATTACK -j LOG --log-prefix "Possible SSH attack! " --log-level 7
ubuntu@ubuntu-server: /etc/ssh$ sudo iptables -A SSHATTACK -j DROP
ubuntu@ubuntu-server: /etc/ssh$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:21:0e:b8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.66/26 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe21:eb8/64 scope link
        valid_lft forever preferred_lft forever
ubuntu@ubuntu-server: /etc/ssh$ sudo iptables -A INPUT -i enp0s3 -p tcp -m state --dport 22 --state NEW -m recent --set
ubuntu@ubuntu-server: /etc/ssh$ sudo iptables -A INPUT -i enp0s3 -p tcp -m state --dport 22 --state NEW -m recent --update --seconds 120 --hitcount 4 -j SSHATTACK
ubuntu@ubuntu-server: /etc/ssh$ sudo
```

Appendix D.1 is a screenshot of the IP address and firewall configuration for the Ubuntu_Server. As shown the IP address is 192.168.10.66 and the firewall blocks all SSH connections for 2 minutes after 4 failed login attempts.

Appendix E.4: Brute Force

```
[kali@kali] ~$ sudo hydra -l root -P /home/kali/Desktop/Password_List 192.168.10.66 ssh
[sudo] password for kali:
Hydra v2.5 (C) 2022 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 2024-03-29 15:49:00
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -i to skip waiting)) from a previous session found, to prevent overwriting. /hydra.restore
[DATA] max 16 tasks per 1 servers overall 16 tasks, 200 login tries (1:1/p:200), ~13 tries per task
[DATA] attacking ssh://192.168.10.66:22/
[STATUS] 26/00 tries/min, 16 tries in 00:00h, 199 to do in 00:11h, 1 active
[ERROR] all children were disabled due too many connection errors.
0 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-29 15:50:15

[kali@kali] ~$
```

```
Mar 29 19:49:44 ubuntuserver kernel: [ 1663.173595] Possible SSH attack! IN=enp0s3 OUT= MAC=08:00:27:21:0e:b8:bc:fs:e2:a4:00:01:08:00 SRC=192.168.10.12 DST=192.168.10.66 LEN=60 TOS=0x00 PREC=0x00 TTL=61 ID=46231 DF PROTO=TCP SPT=42384 DPT=22 WINDOW=32120 RES=0x00 SYN URCP=0
Mar 29 19:49:45 ubuntuserver kernel: [ 1664.191862] Possible SSH attack! IN=enp0s3 OUT= MAC=08:00:27:21:0e:b8:bc:fs:e2:a4:00:01:08:00 SRC=192.168.10.12 DST=192.168.10.66 LEN=60 TOS=0x00 PREC=0x00 TTL=61 ID=46232 DF PROTO=TCP SPT=42384 DPT=22 WINDOW=32120 RES=0x00 SYN URCP=0
Mar 29 19:49:46 ubuntuserver kernel: [ 1665.229834] Possible SSH attack! IN=enp0s3 OUT= MAC=08:00:27:21:0e:b8:bc:fs:e2:a4:00:01:08:00 SRC=192.168.10.12 DST=192.168.10.66 LEN=60 TOS=0x00 PREC=0x00 TTL=61 ID=46233 DF PROTO=TCP SPT=42384 DPT=22 WINDOW=32120 RES=0x00 SYN URCP=0
Mar 29 19:49:47 ubuntuserver kernel: [ 1666.326865] Possible SSH attack! IN=enp0s3 OUT= MAC=08:00:27:21:0e:b8:bc:fs:e2:a4:00:01:08:00 SRC=192.168.10.12 DST=192.168.10.66 LEN=60 TOS=0x00 PREC=0x00 TTL=61 ID=46234 DF PROTO=TCP SPT=42384 DPT=22 WINDOW=32120 RES=0x00 SYN URCP=0
Mar 29 19:49:48 ubuntuserver kernel: [ 1667.346937] Possible SSH attack! IN=enp0s3 OUT= MAC=08:00:27:21:0e:b8:bc:fs:e2:a4:00:01:08:00 SRC=192.168.10.12 DST=192.168.10.66 LEN=60 TOS=0x00 PREC=0x00 TTL=61 ID=46235 DF PROTO=TCP SPT=42384 DPT=22 WINDOW=32120 RES=0x00 SYN URCP=0
Mar 29 19:49:50 ubuntuserver kernel: [ 1669.373488] Possible SSH attack! IN=enp0s3 OUT= MAC=08:00:27:21:0e:b8:bc:fs:e2:a4:00:01:08:00 SRC=192.168.10.12 DST=192.168.10.66 LEN=60 TOS=0x00 PREC=0x00 TTL=61 ID=46236 DF PROTO=TCP SPT=42384 DPT=22 WINDOW=32120 RES=0x00 SYN URCP=0
Mar 29 19:49:54 ubuntuserver kernel: [ 1673.474457] Possible SSH attack! IN=enp0s3 OUT= MAC=08:00:27:21:0e:b8:bc:fs:e2:a4:00:01:08:00 SRC=192.168.10.12 DST=192.168.10.66 LEN=60 TOS=0x00 PREC=0x00 TTL=61 ID=46237 DF PROTO=TCP SPT=42384 DPT=22 WINDOW=32120 RES=0x00 SYN URCP=0
Mar 29 19:50:02 ubuntuserver kernel: [ 1681.611421] Possible SSH attack! IN=enp0s3 OUT= MAC=08:00:27:21:0e:b8:bc:fs:e2:a4:00:01:08:00 SRC=192.168.10.12 DST=192.168.10.66 LEN=60 TOS=0x00 PREC=0x00 TTL=61 ID=46238 DF PROTO=TCP SPT=42384 DPT=22 WINDOW=32120 RES=0x00 SYN URCP=0
```

Appendix E.4 is a screenshot of the Brute Force attack failing because the connection is being blocked by the firewall configured on the Ubuntu_Server. The attack is also being logged in the “/var/log/syslog” file.