Abstract: This lab covers the Kali Linux Open-Source operating system

and its vast amounts of applications to help facilitate various security and penetration tasks with good intentions in mind.

Lab Report 4

CEN 3078 | Dr. Qu

KALI LINUX LAB

Nash Morrison

2024

**Task 1 - Show IP Address (Main & Clone) & Communications:**

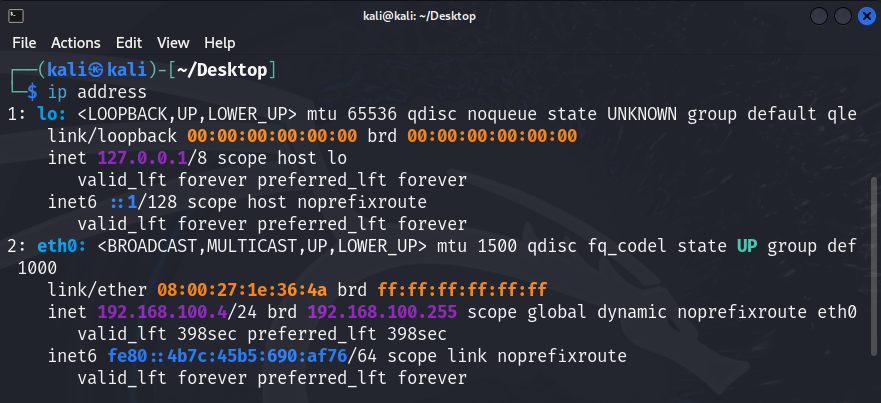


Figure A: IP Address of Main Kali Window

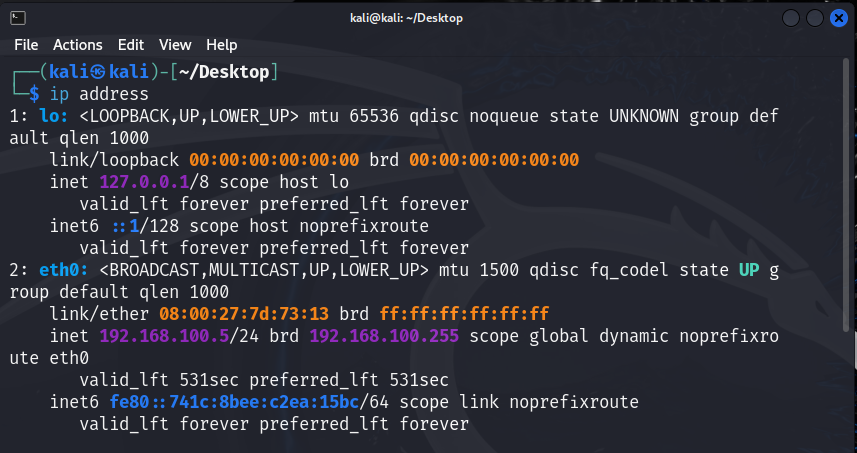
****

Figure B: IP Address of Clone Kali Window

For our first task, there must be an established network connection between our “host” and “clone” machines, respectively. Using the command “ip address” shows that we have both machines up and operational with their own IP addresses. Now let us confirm these results.

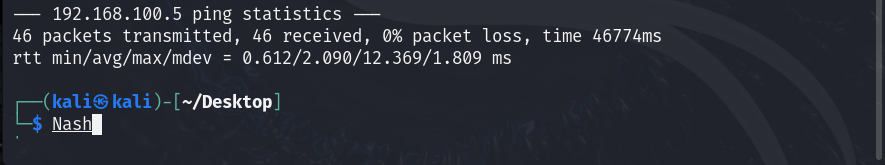
****

Figure C: Established Connection (Main to Clone)

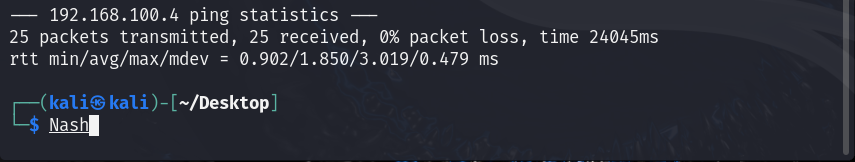
****

Figure D: Established Connection (Clone to Main)

To show that we do in fact have a steady connection between both of our machines, we are going to “**ping**” each IP address from the other machine and let it run for a moment. ^C will pull us out of the test to show that packets are both transmitting and receiving from both ends with 0% packet loss.

**Task 2 – Analysis Using Dmitry:**

A screenshot of a computer program

Description automatically generated

Figure E: Gaining Access to Root Privilege

For this task we are going to grasp a better understanding of Dmitry, which is a search and scanning tool that allows us on websites to get a wealth of information found on the backend. For this section we are going to be running in “root privilege.” We achieved this with our operating system password “kali” to allow for root access.

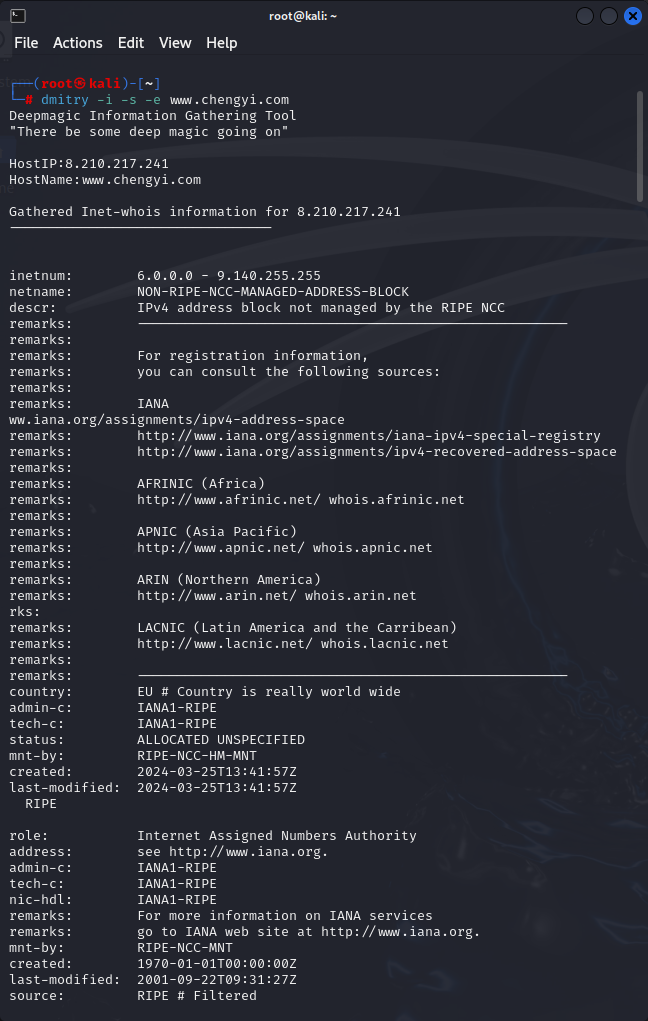
****

Figure F: Given Dmitry Command Pt.1

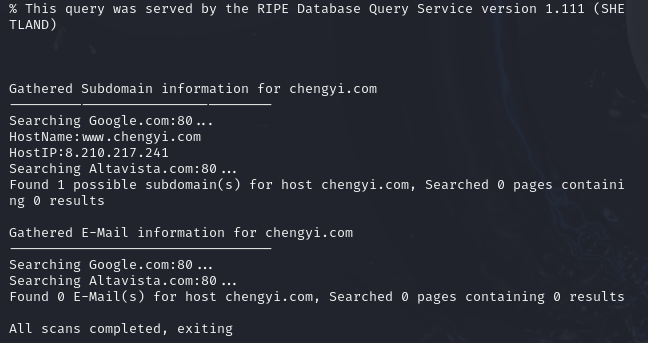


Figure G: Given Dmitry Command Pt.2

The above images show our given Dmitry command found in our PDF document:

One key piece of information that we will be using later in another tool is the actual IP address of our given website. Keeping note that the address is “**8.210.217.241**,” let us move on to our specific Dmitry command to pull our name server and open ports.

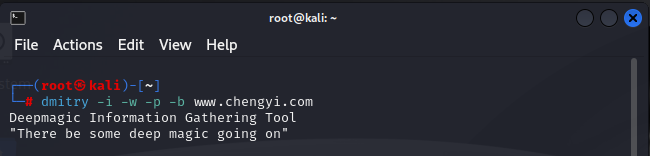


Figure H: My Dmitry Command

I reworked my command to incorporate “-w, -p, and -b” on our given website. “-w” allows for the “who is lookup” on an IP address of a specific host. “-p” performs a TCP port scan on a host, and “-b” reads any banner associated with any port that may be open when searching.

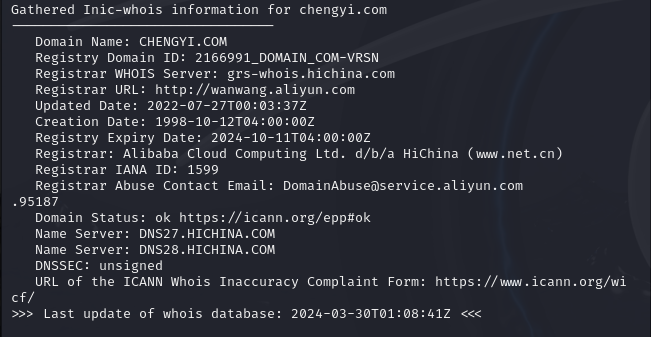


Figure I: Name Server

As we can see our **Name Server** provides us with two different options, with the only difference of the first part of the address changing, [DNS27 -> DNS28]

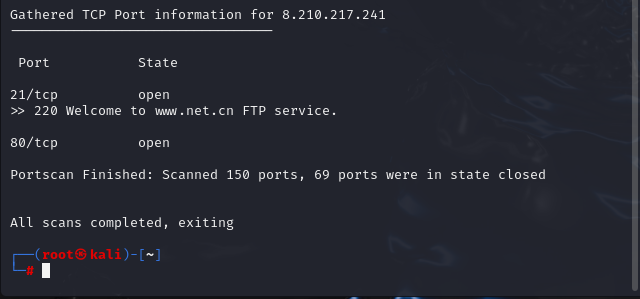


Figure J: Open Ports from IP Address

This is our “-p and -b” commands showcasing that there are two open TCP ports, **21 and 80**. We are shown a welcome banner with TCP port 21 of “220 Welcome to [www.net.cn](http://www.net.cn) FTP service” which could be handy to learn as much as you can about specific ports that stay open for web traffic.

**Task 3 – Recon-ng Testing:**

For this task, the Recon-ng program allows us to reduce time data mining from open sources while also having a plethora of modules, downloadable and used together. We are going to generate a car report using “recon-ng” and the “Hacker Target” Module to create an HTML report to view our data mining results.

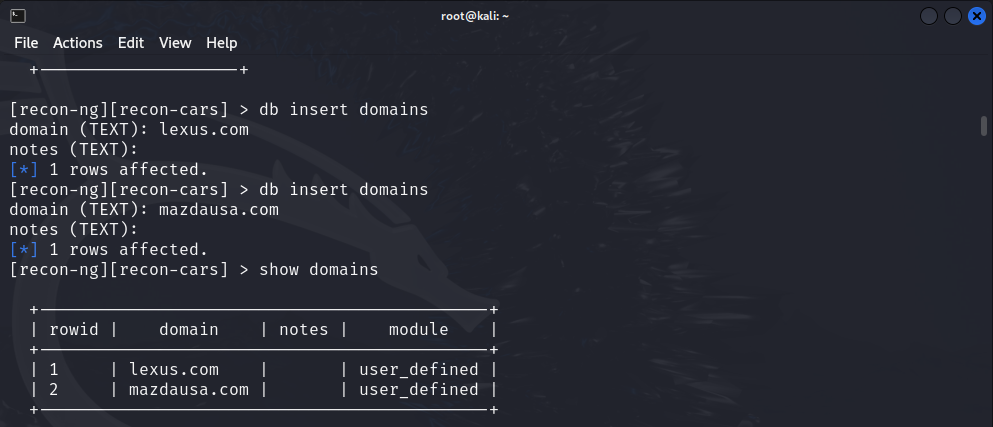
****

Figure K: Changing the Car Titles

For our car report we are going to be using “lexus.com & mazdausa.com” to differentiate from our PDF document.

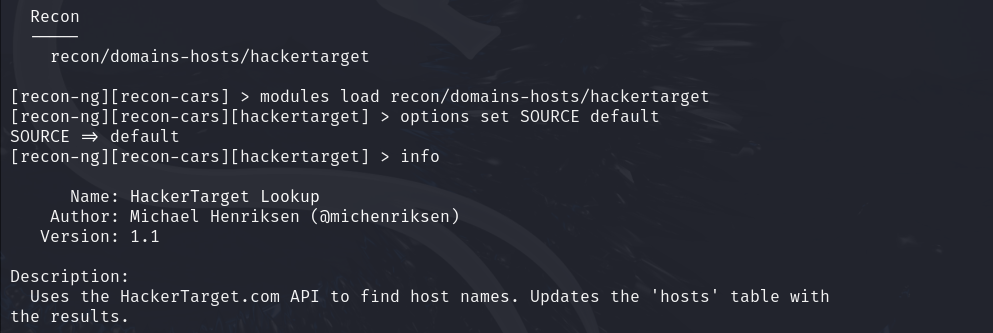


Figure L: Hacker Target Module

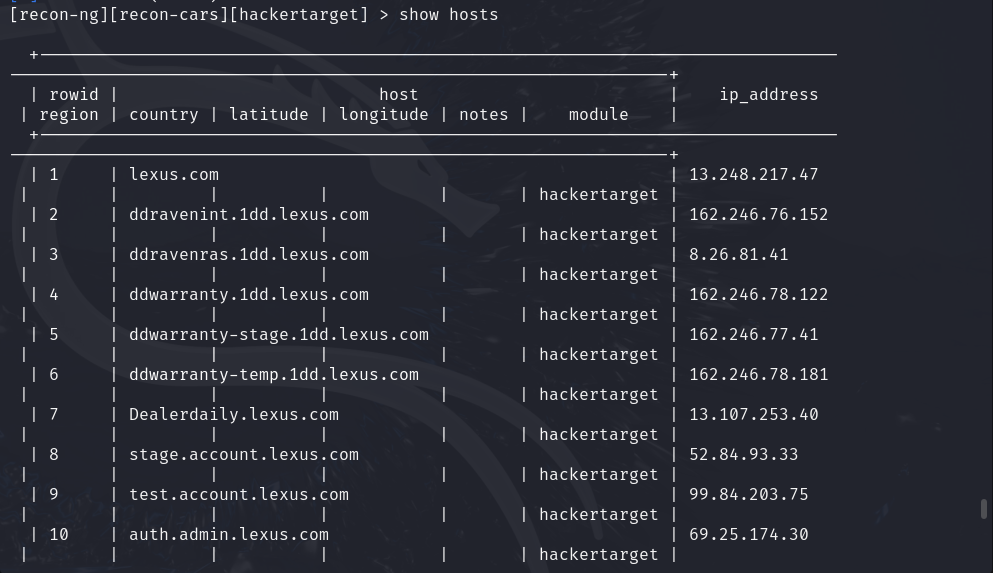


Figure M: Show Hosts Option

Hacker Target simply gathers all the host names found from our search and can compile them in a nice-looking table of values with the “show hosts” command.

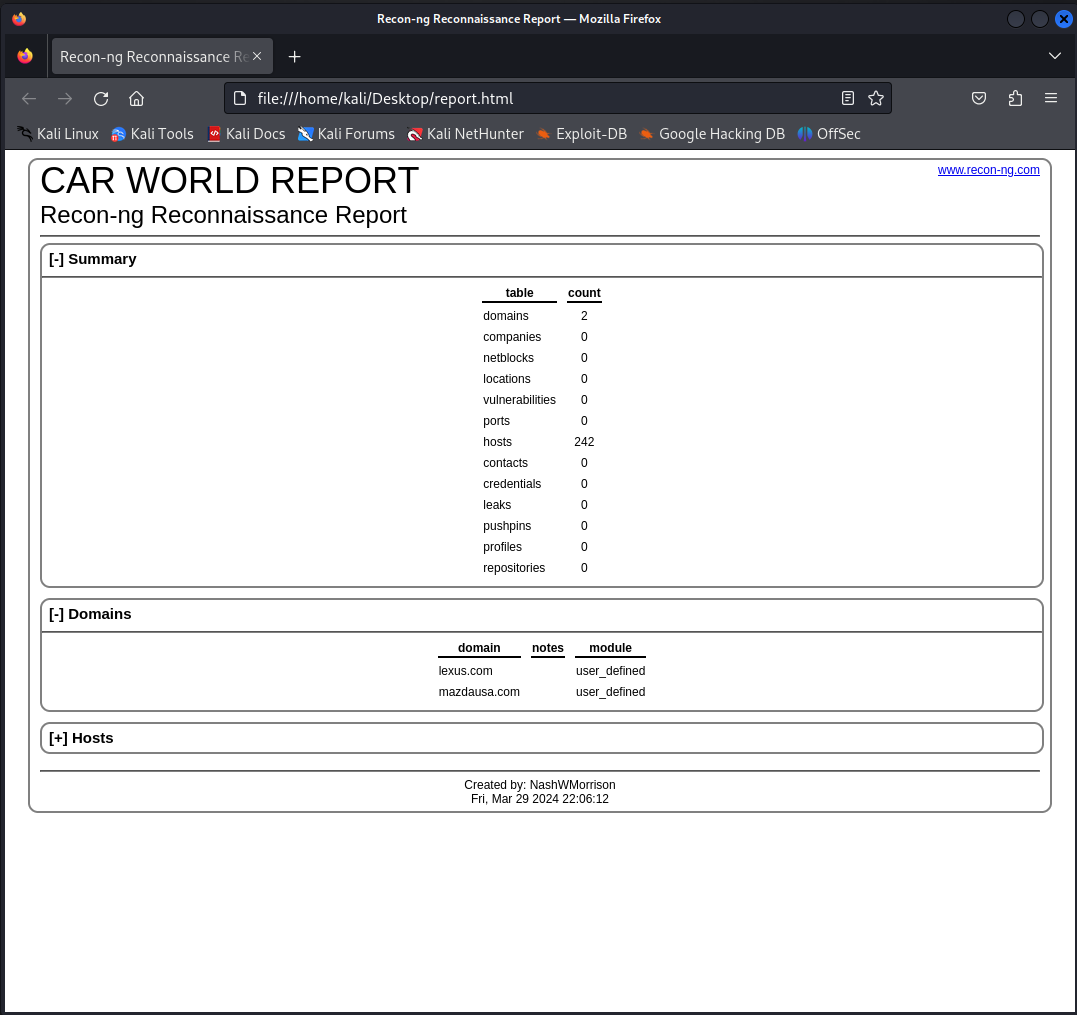


Figure N: HTML Report with Address URL

Creating an HTML file that opens in a web browser provides a nicer looking template for your searching. Provided is a summary of our findings, the domains given to the program, and all the hosts in a listed format.

**Task 4 – Learn The Basics of Metasploit:**

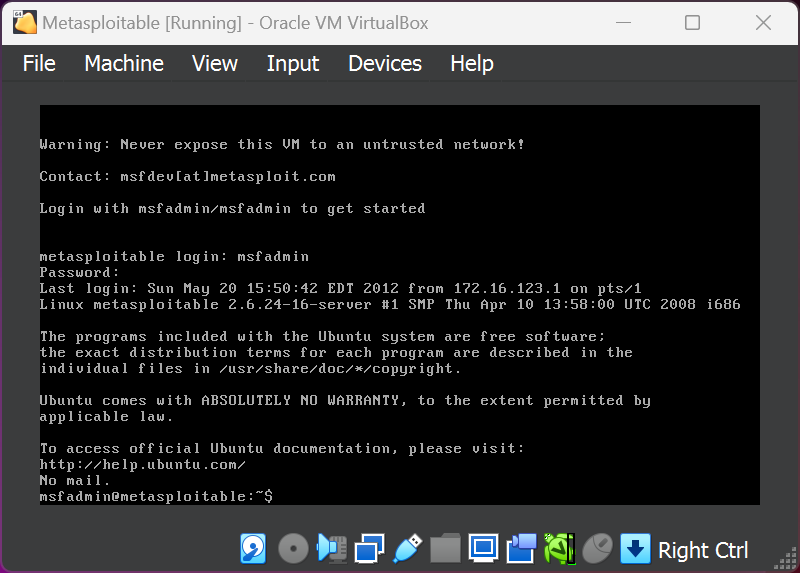


Figure O: Logging into Metasploitable

Our next task has us incorporating Metasploitable2 into the mix, which prides itself on its vulnerabilities and ease of access for aspiring hackers to learn and gather new knowledge and skills.

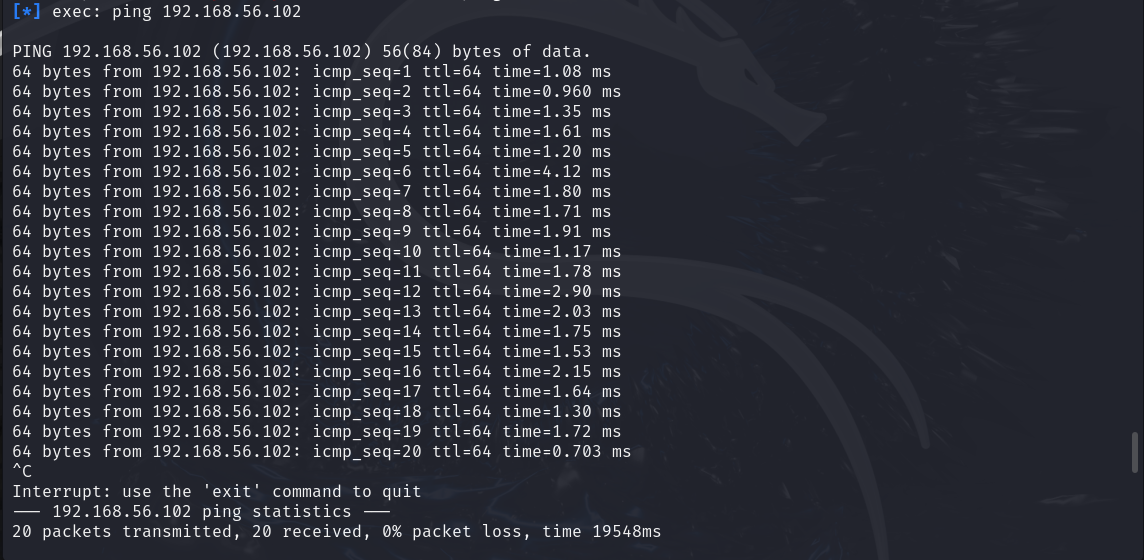
****

Figure P:Ping Test (Linux to Metasploitable)

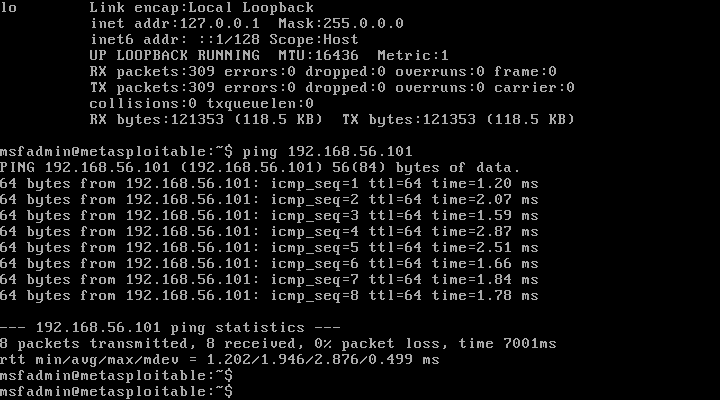


Figure Q:Ping Test (Metasploitable to Linux)

The pictures above showcase that our “Ping” test between both our Linux and Metasploitable client are up and operational between one another.

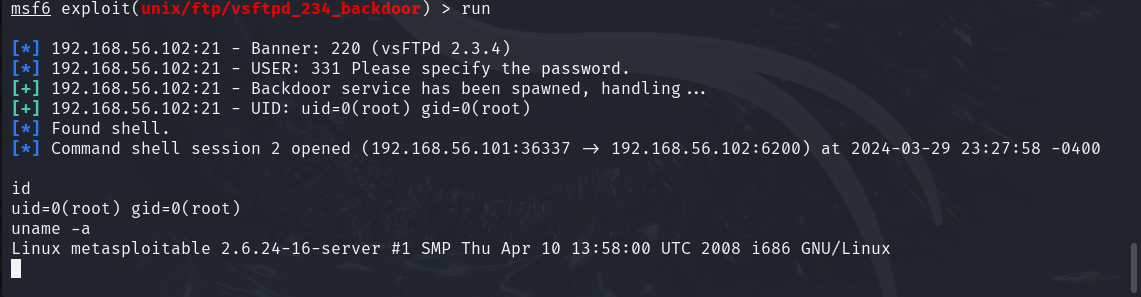
****

Figure R: Shell Hack into Metasploitable

Our main task is to hack into our Metasploitable shell through our Linux command line. This known exploit titled, “ftp/vsftpd\_234\_backdoor” which sends a specific sequence of bytes to port twenty-one which upon successful execution will open a backdoor on port 6200 of the system. The command line becomes blank which is a good thing as far as the Linux client is concerned. Let us double check to see if this backdoor worked.

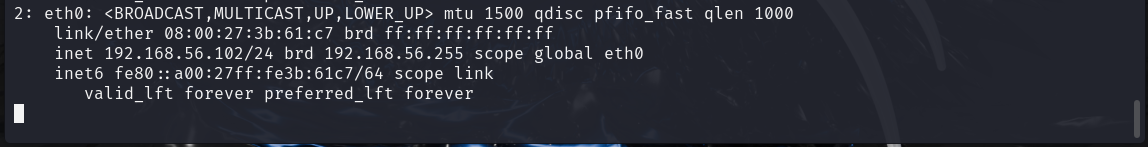
****

Figure S: Confirmation of IP Address Switch

Checking our IP Address does confirm that it looks like we are on the Metasploitable command line from within our Linux Client (Original was 192.168.56.101/24).

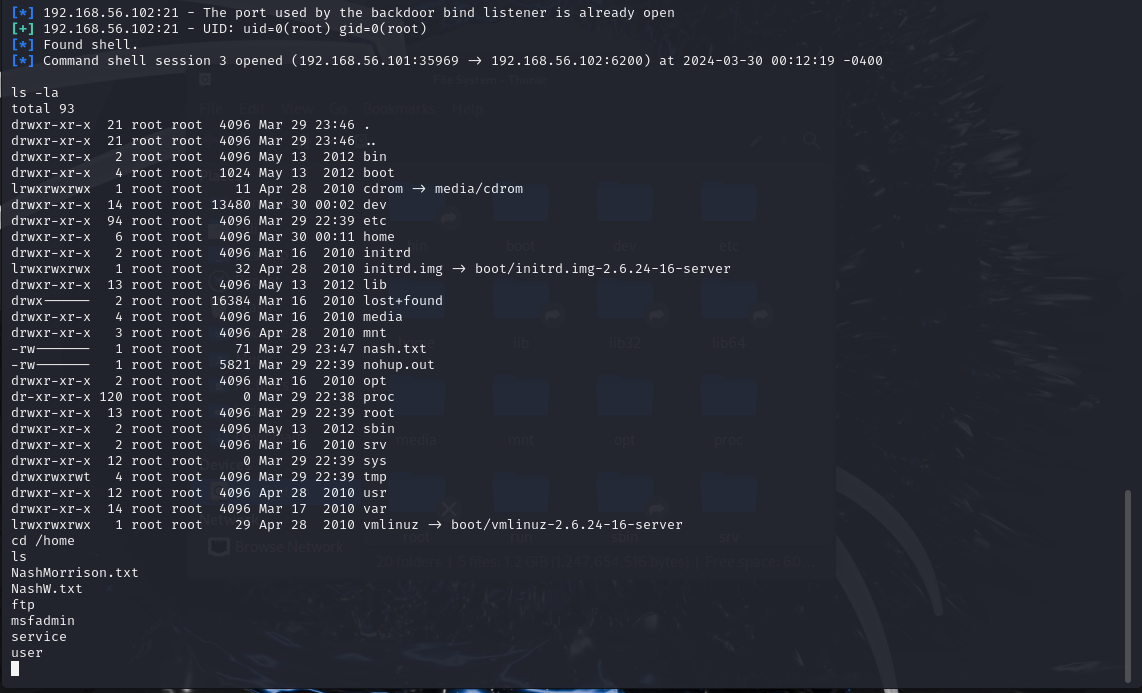
****

Figure T: Text File in Linux VM

The second portion of this task involves creating “.txt” files from within our Linux Client and to show that they are in our Metasploitable2 client. I wrote two files, marked above because I was initially having issues getting them to show in the Metasploitable2 window. However, we can see below from our other image that they do in fact show up.

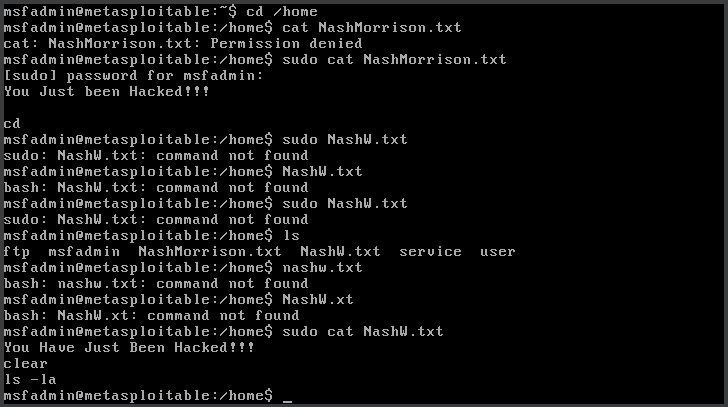
****

Figure U: Text File in Metasploitable VM

Going back over to our Metasploitable2 VM using the command “sudo cat” while inputting our password “msfadmin” will print out the “.txt” files to console.

**Task 4.1 – Extra Tool: [auxiliary/scanner/http/http\_version]**

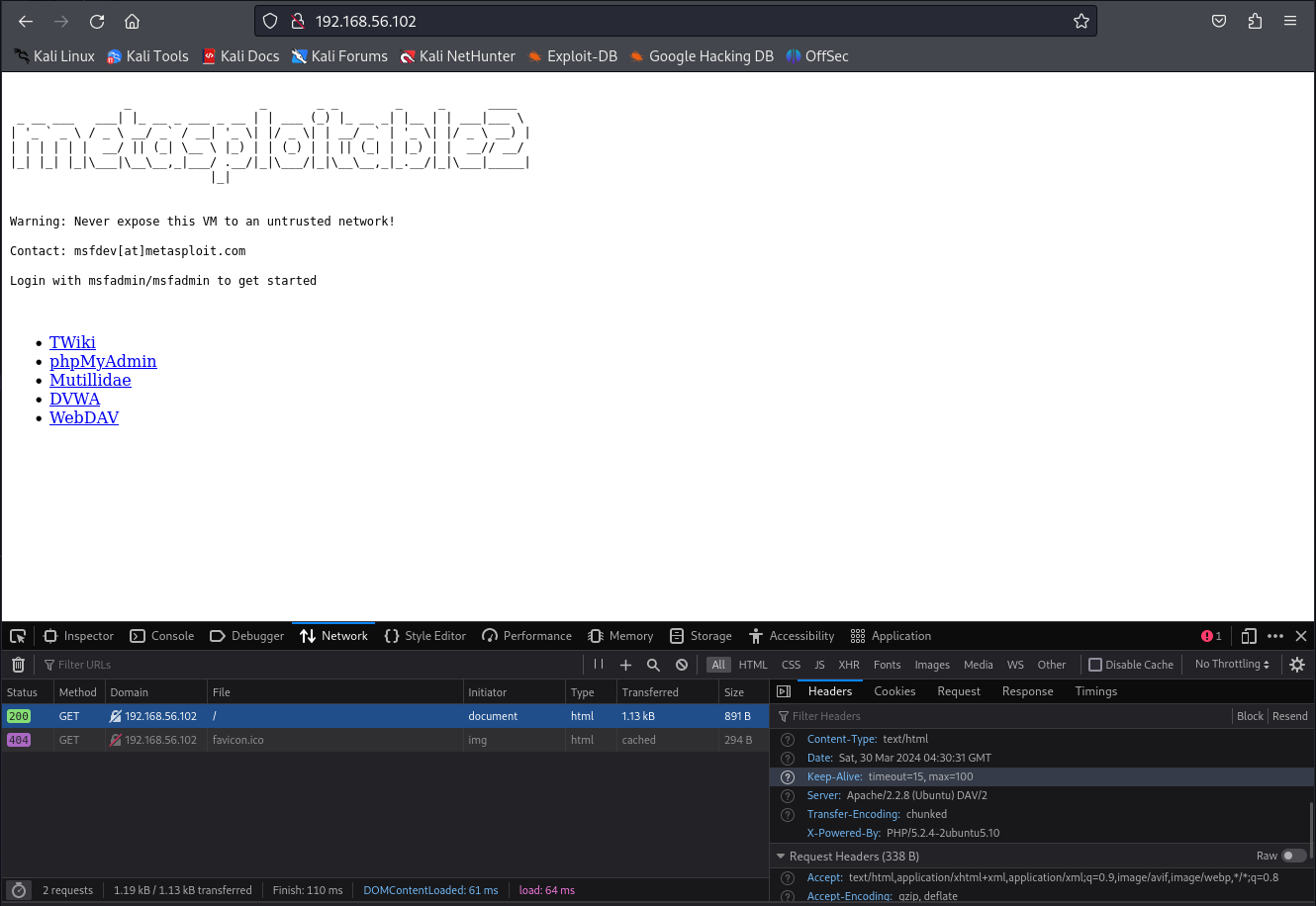
****

Figure V: Developer Options (Network)

Our extra tool has us looking at the ability to lookup the HTTP version of any given website if we have access to its IP address. It is hard to see, but hosts will hide their server and “powered-by” markers because a hacker can utilize potential vulnerabilities in those respective clients. Here we are using the Metasploitable2 webpage. The server is “Apache/2.2.8 (Ubuntu) DAV/2”, powered by “PHP/5.2.4-2ubuntu5.10”. Usually hidden but let us run our extra tool as if we had no idea what those markers were.

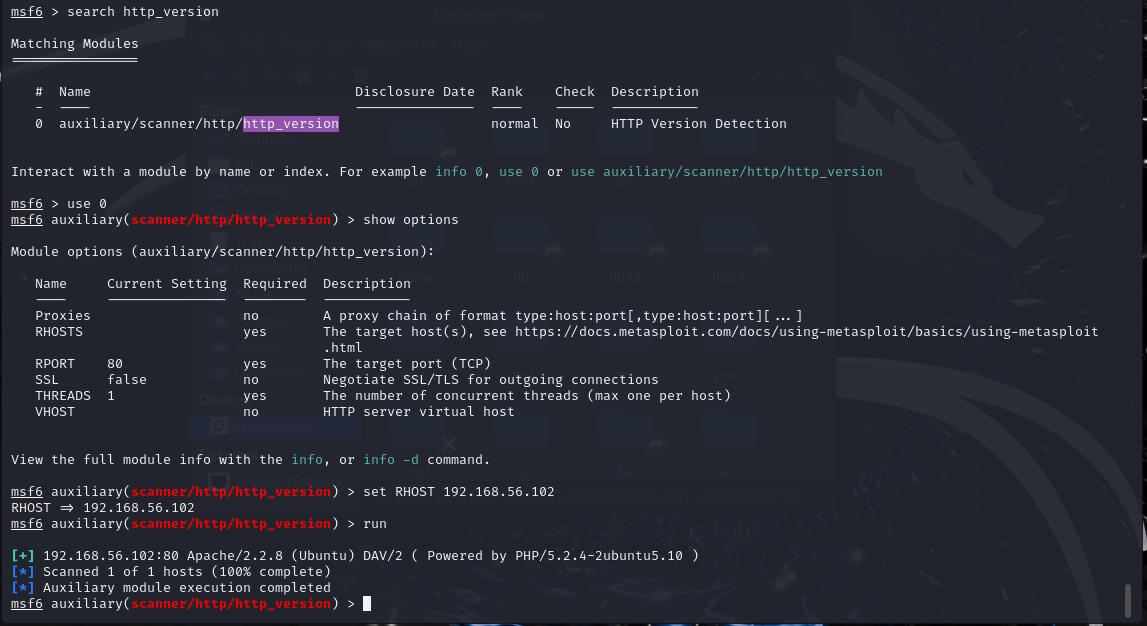
****

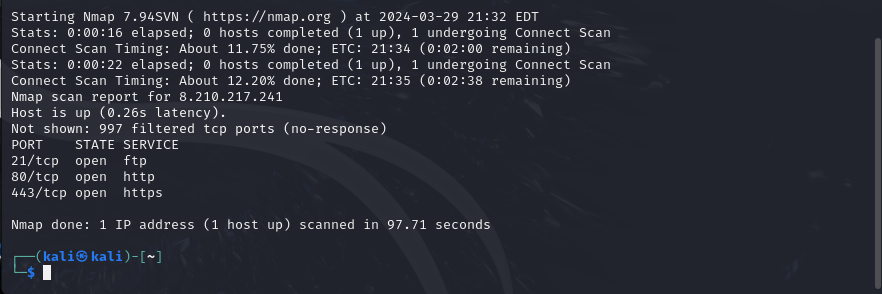
Figure W: HTTP\_Version Module

Setting the RHOST to the IP address provides us with the HTTP version of the website, which diving a hacker can dive deeper to see if there are any vulnerabilities associated with these results.

**EXTRA CREDIT BELOW**

**Extra Credit - Additional Tools:**

**Tool 1 – NMAP:**

****

The most widely used port scanner in the world is Nmap.(“Kali Linux Web Penetration Testing Cookbook - Packt Subscription”) Its ability to scan your perimeter network devices and servers from an external perspective makes this an incredibly powerful network security tool. I wanted to use this tool with our Dmitry task to see if there were any more open ports on the network. Dmitry only seemed to scan up to 150 ports, so I wanted to push the limits a little bit more. Taking our IP address that we took note of earlier in our report, we use our program to see that ports 21 and 80 are open which confirms our results from earlier, but that it also has an open port of 443 which is TCP and provides HTTPS services to users.

**Tool 2 – MacChanger:**

For our second tool, Macchanger allows us to spoof a MAC address so that it makes it much harder to track down your actual machine when completing malicious activities. Our Linux Machine has a starting MAC address of [08:00:27:1e:36:4a]. To start the change in addresses, we must first bring down our Eth0 network for a moment with the “eth0 down” command. Next “-r eth0” to randomize the MAC address that we are spoofing, which now gives us the address of [e6:d3:86:6e:52:75]. Bringing back up the eth0 network and checking our results confirms that we did in fact spoof our MAC address. Our last image highlights that we killed our spoofed address and changed it back to our original.

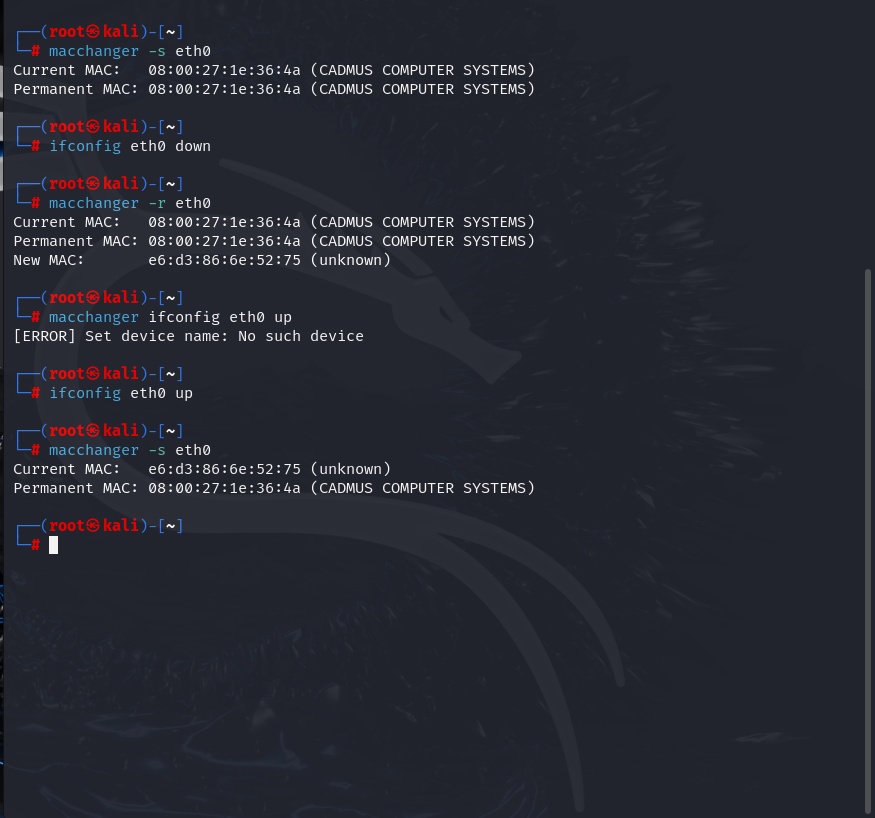
****

Figure X: Changing Old MAC Address to New Address

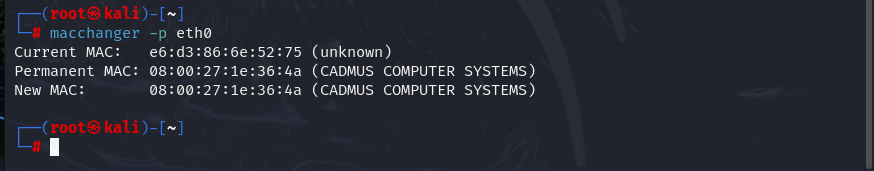


Figure Y: Resetting to Old MAC Address