

University of Hertfordshire
School of Computer Science
BSc Computer Science

Module: Computer Systems Security

Coursework 3

System Security Project Report

Aaron Mascarenhas

Level 6

Academic Year 2020 – 21

Abstract

The goal of this project was to conduct a penetration test on a target computer system, to find vulnerabilities and exploit them using tools that performed penetrating testing on Linux machines. The project consisted of several tasks, which aimed at testing a computer system on the target machine according to a pre-prepared plan where a server was set up by the university to perform a penetration test on the target machine. and finally, write the penetration testing report.

The results of the vulnerability scan on the target showed that there was a vulnerability using the DIRB scanner which was used as a dictionary-based attack against a web server to find all the hidden directories. The result of the vulnerability scan showed a list of directories that are linked to this server.

Four vulnerabilities were chosen to be exploited using the Metasploit Framework and other methods. The result of the exploits and the mitigation for each of them were, the apache web server had had a nonsecure webpage, giving hackers access to accounts and passwords. This demonstrates that a hacker may easily acquire access to a web server's list of safe usernames and passwords and that the hacker will be able to access and discover them even if the passwords and usernames are updated and altered to mitigate this the directory listing should be disabled, the apache software should be hidden from all but the administrator, the apache software should be updated regularly, and the usernames and passwords should be changed and encrypted ONLY after all of this have been done. The second vulnerability was a Default Credential Exploit where the default username and password were being used and unmodified on the login page to PhpMyAdmin, to mitigate this Admin/IT Security personnel can prevent root user access, alter the default PhpMyAdmin access URL, set a secure password, make frequent backups, and maintain software up to date to safeguard the website. The third vulnerability showed the account leaks of usernames and passwords of users who are linked with the network, was found for the first vulnerability for the Apache server. This exploit according to my reach is related to a Data Breach, in order to mitigate this issue, the password needs to be changed in order to maintain security within the server and avoid attackers from logging into the network. The fourth exploit was performed on the system had been rendered inaccessible as a result of a DOS assault. Users were unable to access the system, resulting in downtime and, in the case of a business, earnings loss owing to the inability to use the system.

The conclusions that could be drawn from this penetration project were that the system that was the subject of this penetration test has revealed that it has several exploits, the majority of which can render the system useless or non-existent if attackers use the exploits described. Due to the lack of encryption of usernames and passwords, as well as PhpMyAdmin's default user credentials, the system was extremely easy to access. what actions should be taken, and so forth.

Table of Context

1.0 Introduction

2.0 Attack Narrative

2.1 Information Gathering

2.2 Scanning and Enumeration

2.3 Vulnerability Identification

2.4 Vulnerability analysis

3.0 Vulnerability Exploitation using tools

4.0 Vulnerability Mitigation

5.0 Conclusions

6.0 Overall Conclusions and Reflections

7.0 References

8.0 Appendix

1.0 Introduction

This penetration testing project was completed as part of the Computer System Security module, to consider security from the vulnerabilities and exploitations point of view. The project

involved the attack on a Linux system, following the same steps as prescribed by penetration testing methodologies, such as PTES. The main purpose of penetration tests is to discover vulnerabilities on systems that can be exploited by malicious attackers. Therefore, an important part of the work is to report on these vulnerabilities and the mitigation methods

This report will go through the exploits which were found while performing penetrating testing, which will be discussed in the Attack Narrative section, and will go through a brief explanation of the vulnerabilities that were found from the risks and mitigation standpoint.

2.0 Attack Narrative

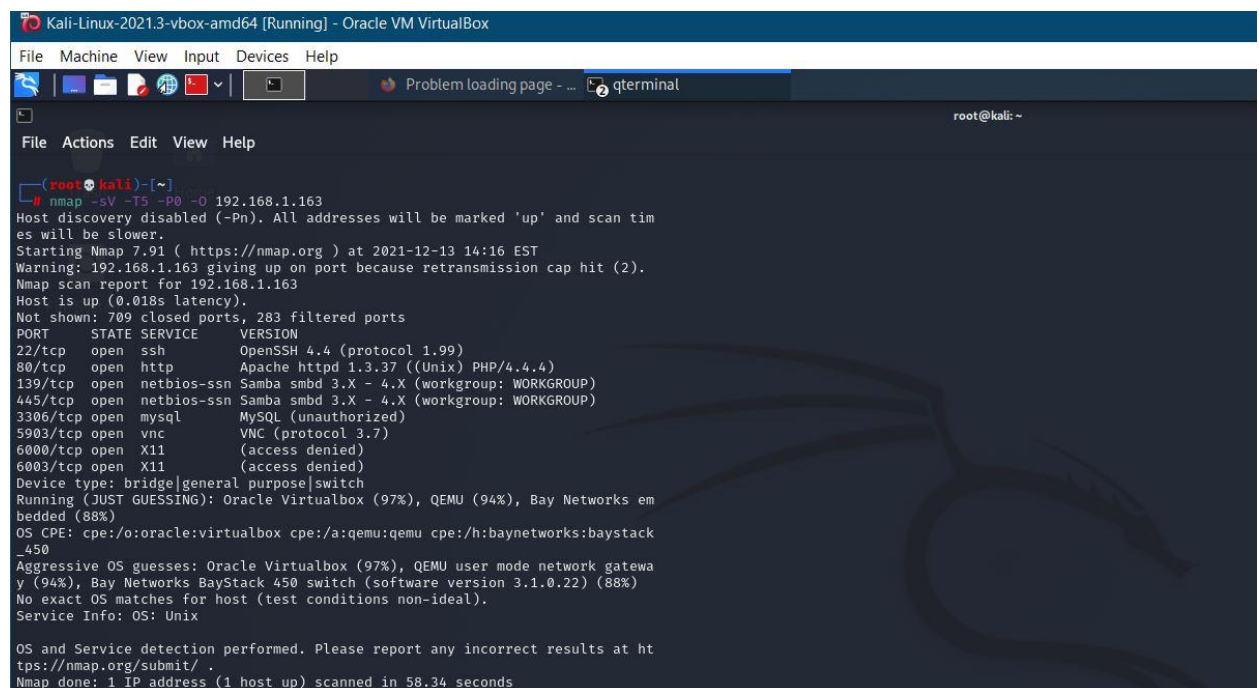
This section will describe how the attack was performed on the targeted machine which the help of tools that were used in this test. Below is a step-by-step process of how the penetrations testing was conducted.

2.1 Information Gathering

There was not much information that was gathered as the target's IP address was already provided to perform the test. In any case, if the IP address of the target was not provided, there are still ways to find out using tools such as Command prompt or a terminal in a Window or a Linux machine to trace the target address by identifying the Domain Name of the targeted server and running commands to locate their Address.

2.2 Scanning and Enumeration

In this section, to scan for any open port on the server a scanning tool was used which was called **Nmap** (Network mapper) to find any open ports on the 192.168.1.163 server. The scan results showed that the target system was running services such as Port 22-SSH, port 80-HTTP, port 139/445 NetBIOS-SSN, port 3306-MySQL, port 5903-VNC, port 6000/6003 X11 which all had open ports, Port 80 had a hyperlink to an HTTP server for **Apache**. These services were analyzed using the software version, and research was performed to find out any vulnerabilities that were present in the software version before a vulnerability scan could be run on the target. This process was conducted in advance to prepare for analysis further in the test.



```
Kali-Linux-2021.3-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Problem loading page - ... qterminal
root@kali: ~

root@kali:~# nmap -sV -T5 -P0 -O 192.168.1.163
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-13 14:16 EST
Warning: 192.168.1.163 giving up on port because retransmission cap hit (2).
Nmap scan report for 192.168.1.163
Host is up (0.018s latency).
Not shown: 709 closed ports, 283 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.4 (protocol 1.99)
80/tcp    open  http      Apache httpd 1.3.37 ((Unix) PHP/4.4.4)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
3306/tcp  open  mysql     MySQL (unauthorized)
5903/tcp  open  vnc       VNC (protocol 3.7)
6000/tcp  open  X11       (access denied)
6003/tcp  open  X11       (access denied)
Device type: bridge|general purpose|switch
Running (JUST GUESSING): Oracle Virtualbox (97%), QEMU (94%), Bay Networks em
bedded (88%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:qemu:qemu cpe:/h:baynetworks:baystack
_450
Aggressive OS guesses: Oracle Virtualbox (97%), QEMU user mode network gatewa
y (94%), Bay Networks BayStack 450 switch (software version 3.1.0.22) (88%)
No exact OS matches for host (test conditions non-ideal).
Service Info: OS: Unix

OS and Service detection performed. Please report any incorrect results at ht
tps://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 58.34 seconds
```

Fig 1: Nmap Scan

2.3 Vulnerability Identification

In fig 2.0, a vulnerability scanning was performed to analyze the vulnerabilities inside the targeted systems' network, and the Nessus Scanner program was used to complete the scan. This stage is critical before the real penetration test since it will, and has, disclosed issues that the pen test will confirm are vulnerable, thus the name "exploits". Unfortunately, OpenVAS AND Nessus could not be used since they just did not work.

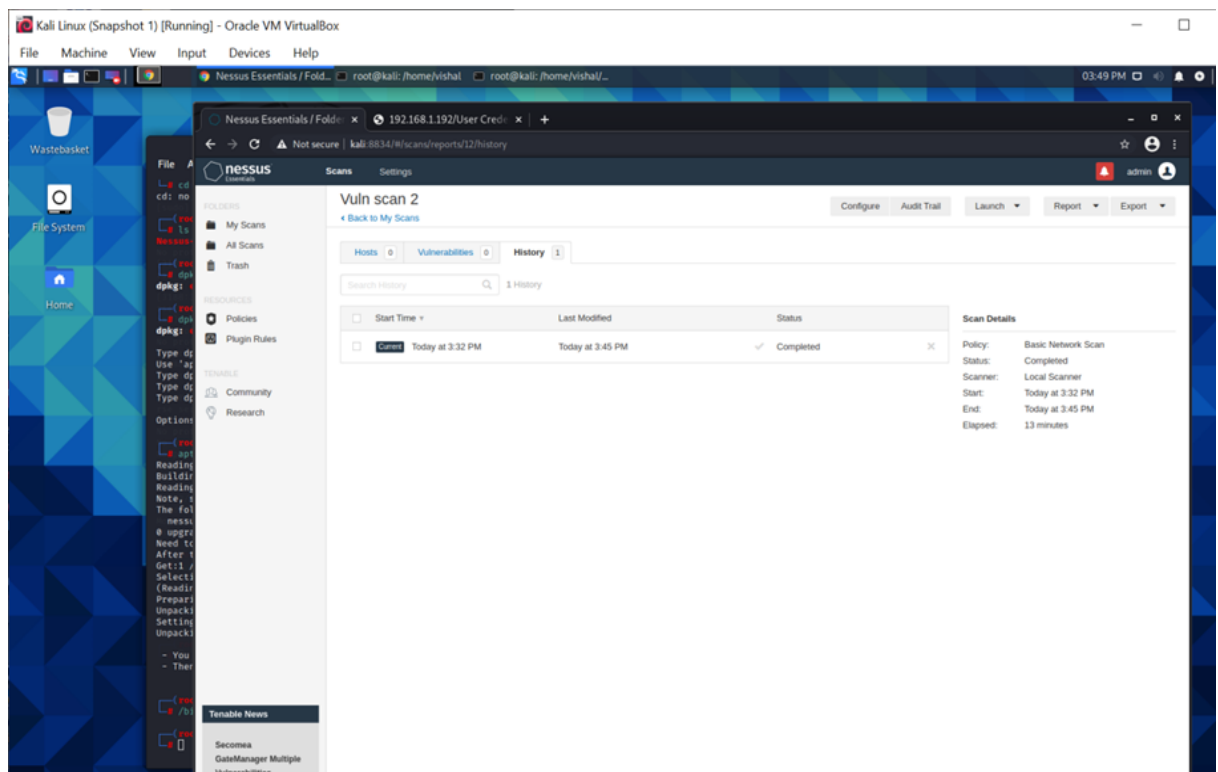
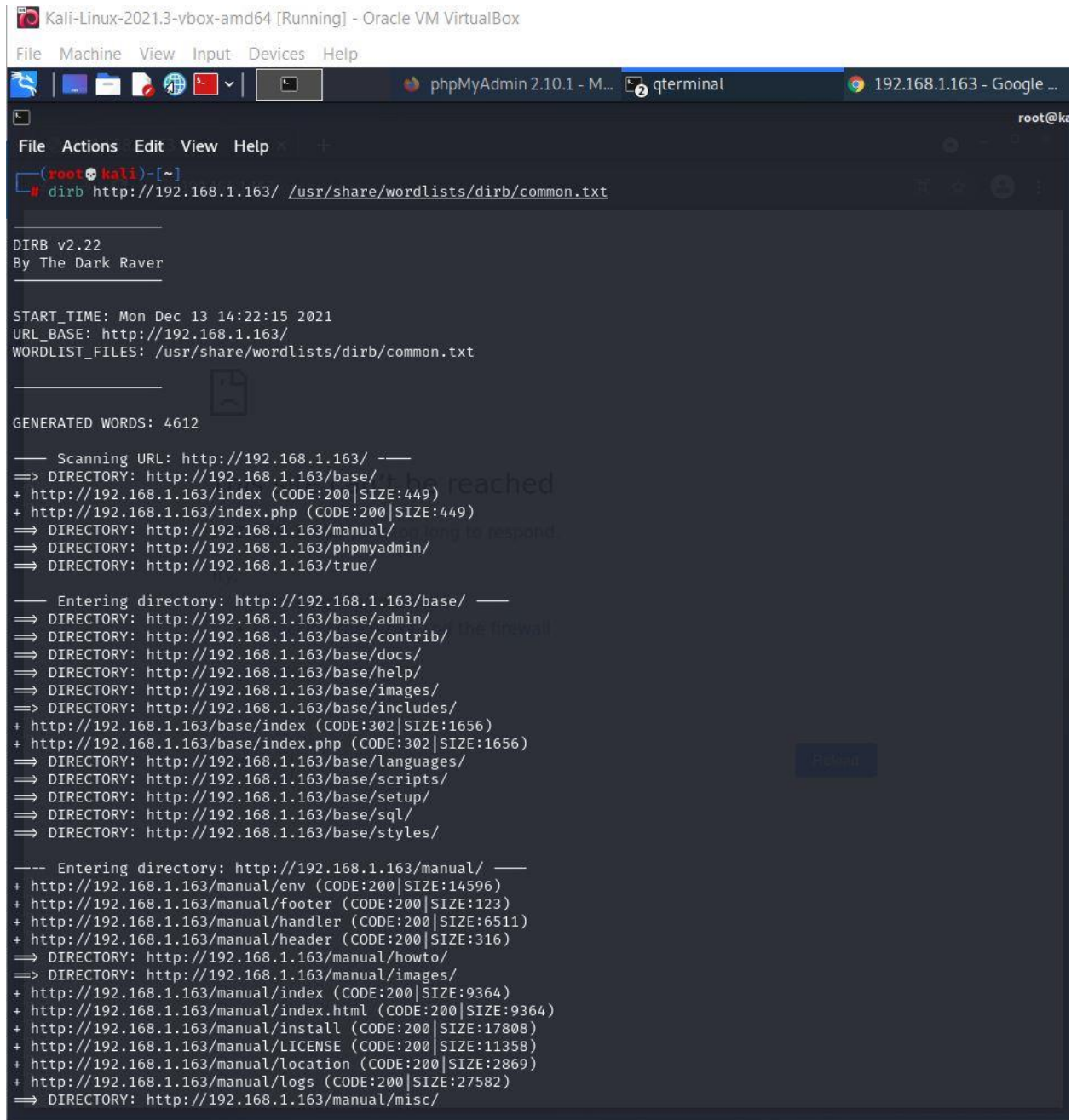


Fig 2.0: Nessus Scan

In fig 3.0, As part of the vulnerability scanning, the DIRB scanner was used as a dictionary-based attack against a web server to find all the hidden directories. The result of the vulnerability scan showed a list of directories that are linked to this server which include a directory for a Manual, images, scripts, etc. but the interesting directory is the phpMyAdmin which we will get into later in the test.

Name: Aaron Mascarenhas
Module: Computer Systems Security



```
Kali-Linux-2021.3-vbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
phpMyAdmin 2.10.1 - M... qterminal 192.168.1.163 - Google ...
root@kali

File Actions Edit View Help
(root@kali)~
# dirb http://192.168.1.163/ /usr/share/wordlists/dirb/common.txt

DIRB v2.22
By The Dark Raver

START_TIME: Mon Dec 13 14:22:15 2021
URL_BASE: http://192.168.1.163/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt

GENERATED WORDS: 4612

--- Scanning URL: http://192.168.1.163/ ---
=> DIRECTORY: http://192.168.1.163/base/
+ http://192.168.1.163/index (CODE:200|SIZE:449)
+ http://192.168.1.163/index.php (CODE:200|SIZE:449)
=> DIRECTORY: http://192.168.1.163/manual/
=> DIRECTORY: http://192.168.1.163/phpmyadmin/
=> DIRECTORY: http://192.168.1.163/true/

--- Entering directory: http://192.168.1.163/base/ ---
=> DIRECTORY: http://192.168.1.163/base/admin/
=> DIRECTORY: http://192.168.1.163/base/contrib/
=> DIRECTORY: http://192.168.1.163/base/docs/
=> DIRECTORY: http://192.168.1.163/base/help/
=> DIRECTORY: http://192.168.1.163/base/images/
=> DIRECTORY: http://192.168.1.163/base/includes/
+ http://192.168.1.163/base/index (CODE:302|SIZE:1656)
+ http://192.168.1.163/base/index.php (CODE:302|SIZE:1656)
=> DIRECTORY: http://192.168.1.163/base/languages/
=> DIRECTORY: http://192.168.1.163/base/scripts/
=> DIRECTORY: http://192.168.1.163/base/setup/
=> DIRECTORY: http://192.168.1.163/base/sql/
=> DIRECTORY: http://192.168.1.163/base/styles/

--- Entering directory: http://192.168.1.163/manual/ ---
+ http://192.168.1.163/manual/env (CODE:200|SIZE:14596)
+ http://192.168.1.163/manual/footer (CODE:200|SIZE:123)
+ http://192.168.1.163/manual/handler (CODE:200|SIZE:6511)
+ http://192.168.1.163/manual/header (CODE:200|SIZE:316)
=> DIRECTORY: http://192.168.1.163/manual/howto/
=> DIRECTORY: http://192.168.1.163/manual/images/
+ http://192.168.1.163/manual/index (CODE:200|SIZE:9364)
+ http://192.168.1.163/manual/index.html (CODE:200|SIZE:9364)
+ http://192.168.1.163/manual/install (CODE:200|SIZE:17808)
+ http://192.168.1.163/manual/LICENSE (CODE:200|SIZE:11358)
+ http://192.168.1.163/manual/location (CODE:200|SIZE:2869)
+ http://192.168.1.163/manual/logs (CODE:200|SIZE:27582)
=> DIRECTORY: http://192.168.1.163/manual/misc/
```

Fig 3.0: DIRB Scan

2.4.0 Vulnerability analysis

The vulnerability exploitation step involves putting vulnerabilities discovered during the earlier identification phase through penetration testing to see whether they can be exploited. When a hacker can use a vulnerability to carry out hostile action, it is called an exploit. The pen test revealed four exploits, which are explained in further detail in the following section.

2.4.1 – Browser Exploitation (Apache Web Server 1.3.37)

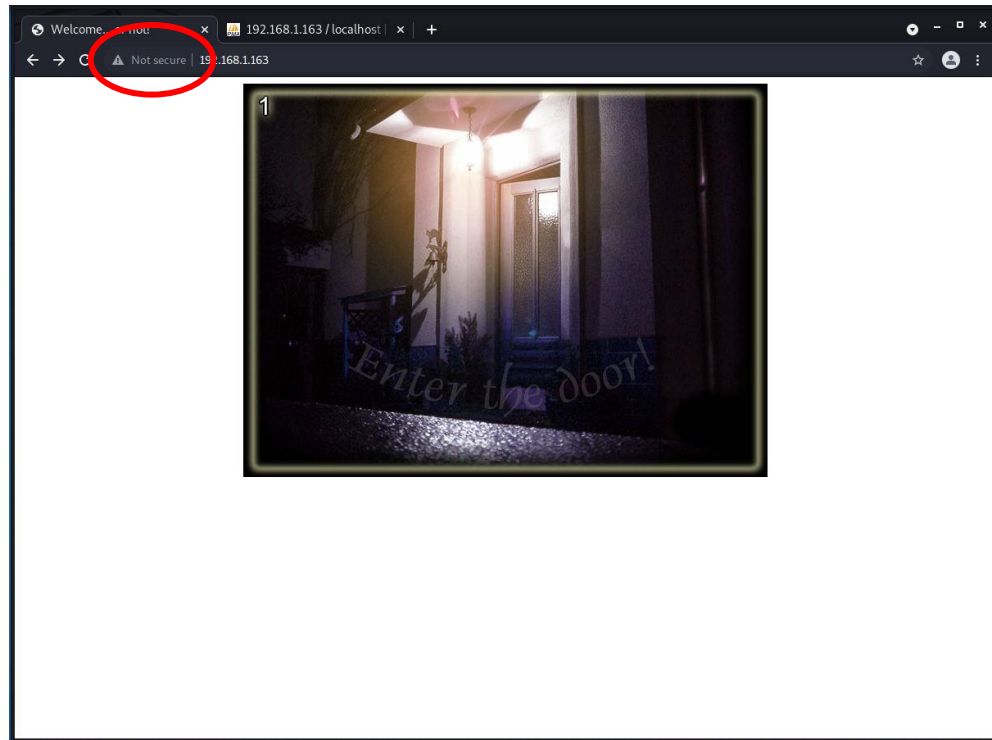


Fig 3.0: Apache Webpage hyperlinked image

Google Chrome was installed on Kali Linux to attack the Apache web server, and the target IP (192.168.1.163) was scanned with the DIRB tool (as mentioned in section 3.0) and browsed to. An interactive, hyperlinked door was presented on the webpage. The above fig 3.0 shows a webpage for the targets after entering their IP address which displays an image with the message “Enter the Door”. Upon further inspection, the webpage seems to be non-secure without any security protocol such as “HTTPS”, this leads to cyber threats that include malware and cyberattacks. Looking at the image displayed on the page, and upon clicking on it, it redirects to a second page.

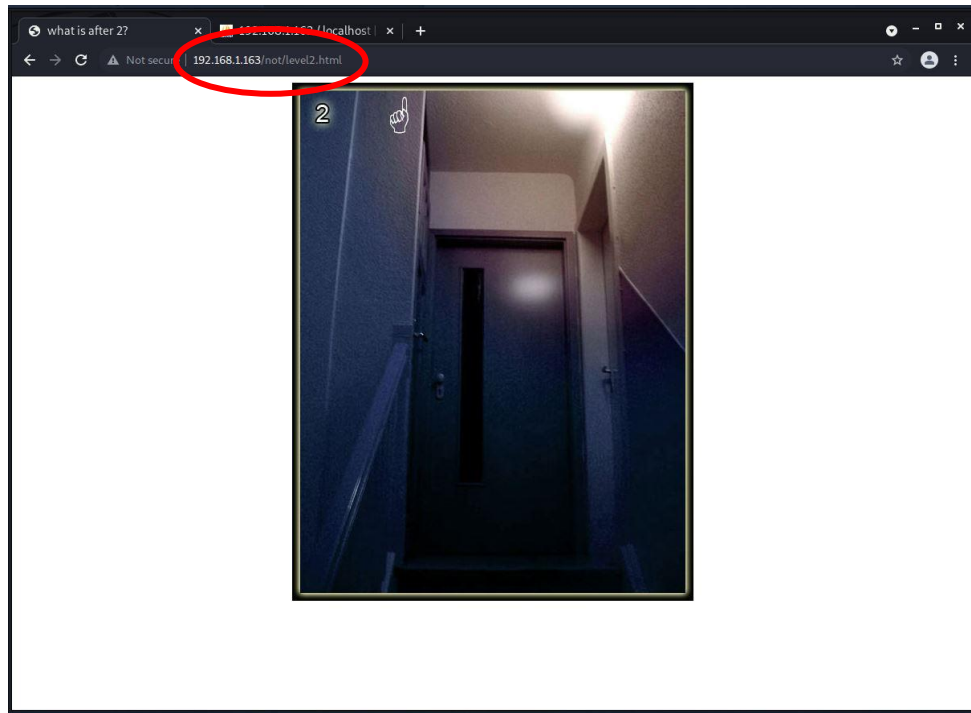


Fig 3.1: Apache Page 2

After clicking on the image, the above fig 3.1 takes us to a webpage with the URL named level 2, and an image is displayed in the center. If we look at the URL it says it's not secure and has a not level 2 directory list. Here, if you notice, the URL is set to “**/not/level2.html**”.

Here, the risk of an attacker changing the URL to find directories within the website to find directory indexes which may include usernames and passwords linked to the website.

Name: Aaron Mascarenhas
Module: Computer Systems Security

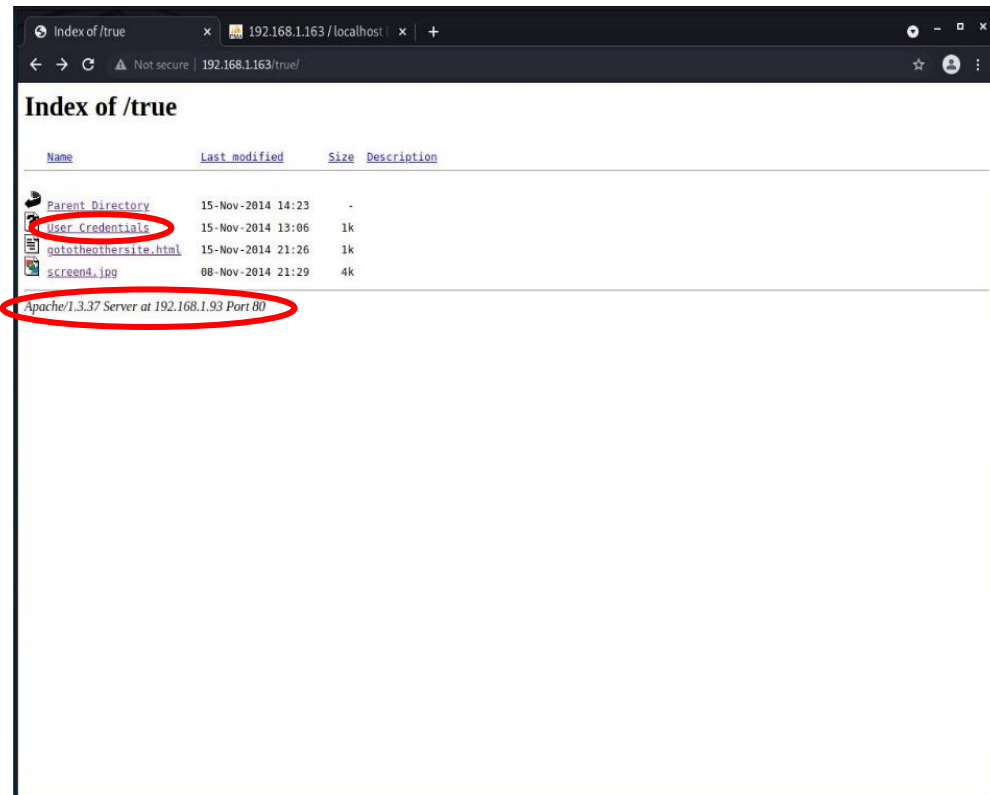


Fig 3.2 Client web Index Page

Fig 3.2 shows if the URL is changed to **“/true”** it takes us to another page with the title **“Index of /true”** where it displays a list of directories that include the **user’s credentials**, and other documents. Here, the vulnerability is the version of Apache version the server is running which is **1.3.37**, and the server address which is **192.168.1.93** and Port **80**, with this information the attacker can find out if there are exploits by checking if the server is running an older version of the software which may have bugs and were patched in the later updates.

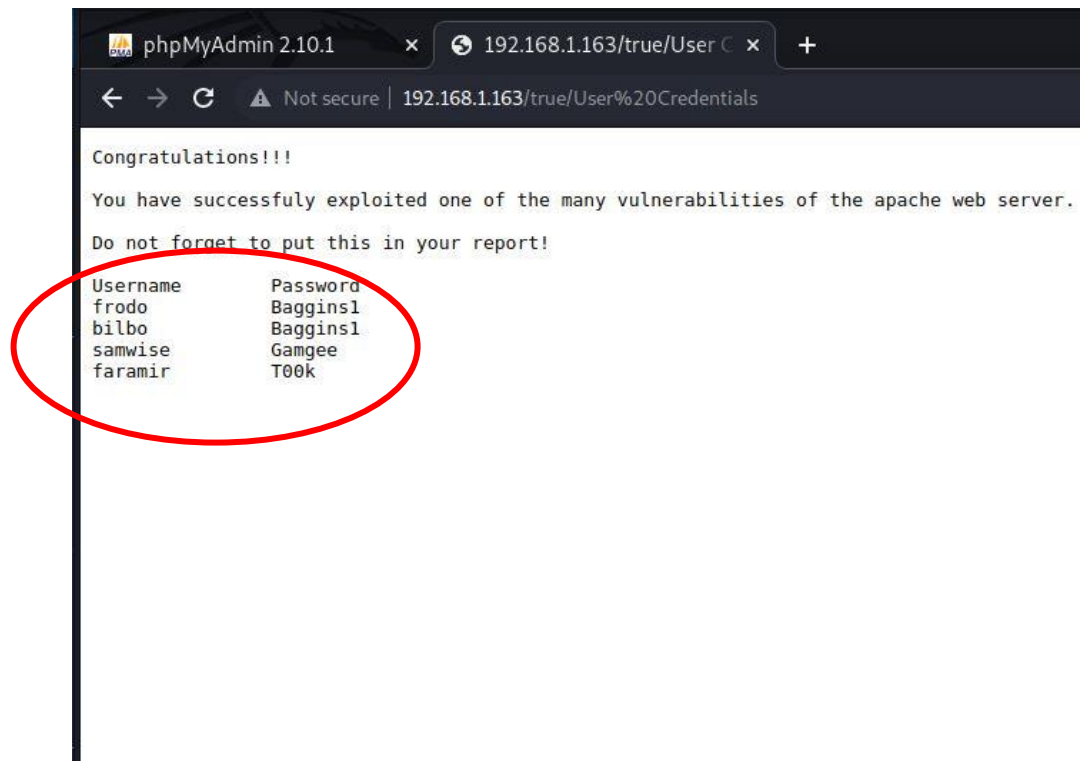


Fig 3.3: Apache user credentials Vulnerability

Fig 3.3, is an exploit that shows account leaks of usernames and passwords of users who are linked with the network, was found for the first vulnerability for the Apache server. This exploit according to my reach is related to a **Data Breach**.

2.4.2 – PhpMyAdmin – Default Credential Exploit

1 - Open phpMyAdmin Directory found by DIRB using chrome

In the target system, PhpMyAdmin was being utilized for administrative purposes. This information was discovered during the DIRB scan, which revealed the appropriate indexes. (Figure 4.0).

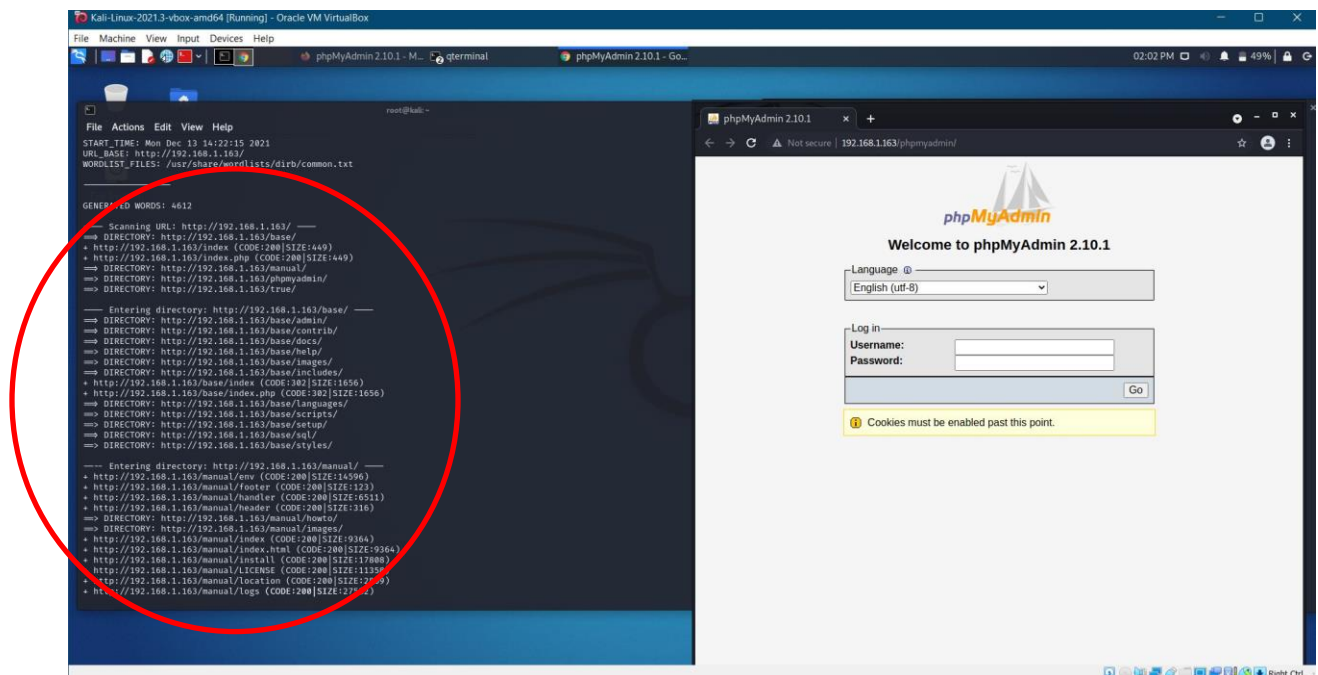


Fig 4.0: phpMyAdmin Directory link

On the PHPMYAdmin directory, the "default credentials" attack has been utilized. In this hack, research was undertaken to identify what the default username and password for the PHPMYAdmin directory should be, which in this case is "root" and the password box is left blank, however, it is changed to password in certain versions."

Name: Aaron Mascarenhas
Module: Computer Systems Security

PhpMyAdmin Client Home page of Apache webserver

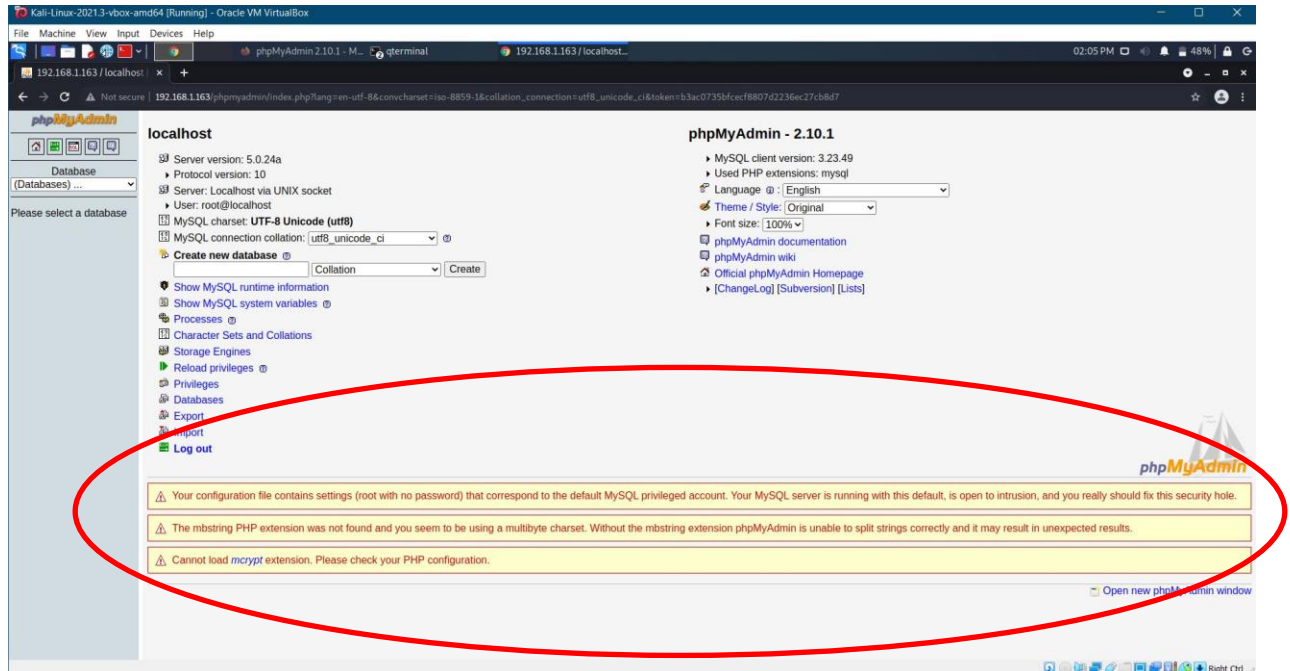


Fig 5.0: Logging into phpMyAdmin Client Details

When the e107_user table was accessed after the default user credentials had been entered and the login page had been successfully finished, an encrypted password was displayed. (Fig 6.0).

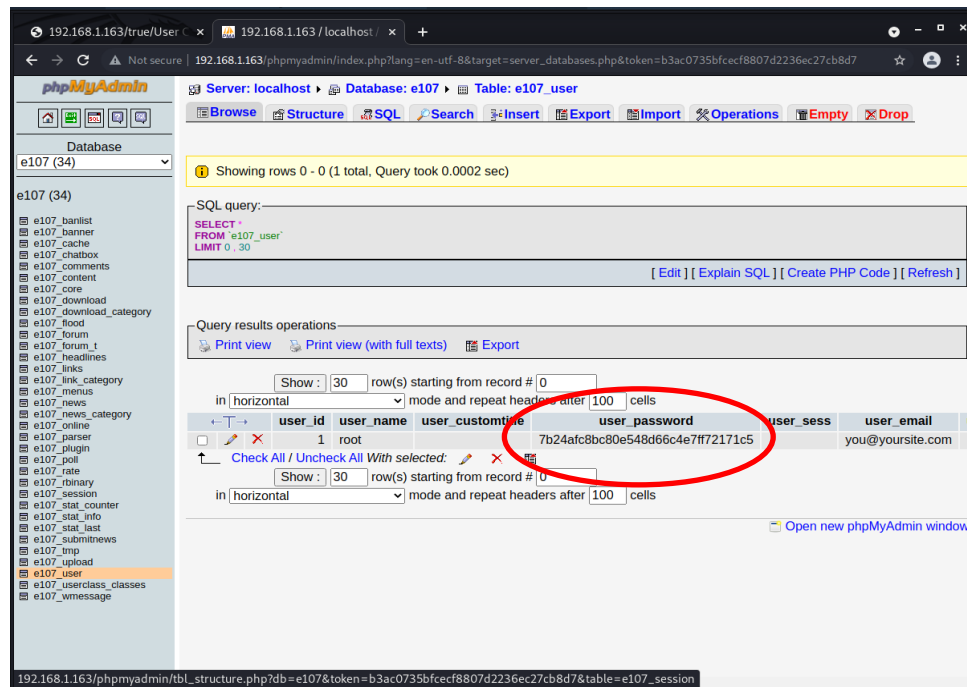


Fig 6.0: e107_user password details phpMyAdmin

Name: Aaron Mascarenhas
Module: Computer Systems Security

3.0 Vulnerability Exploitation using tools

To exploit these vulnerabilities, Exploitation tools were used to carry out means to bring down the server and to find and collect information that could be used for malicious intent. These are shown below with the help of screenshots to document the results.

3.0.1 John the Ripper Tool

The "John the Ripper" program had been used to decode the encrypted password obtained from the previous vulnerability. The encrypted password, "toor," was shown when the instrument had completed its decoding (Fig 7).

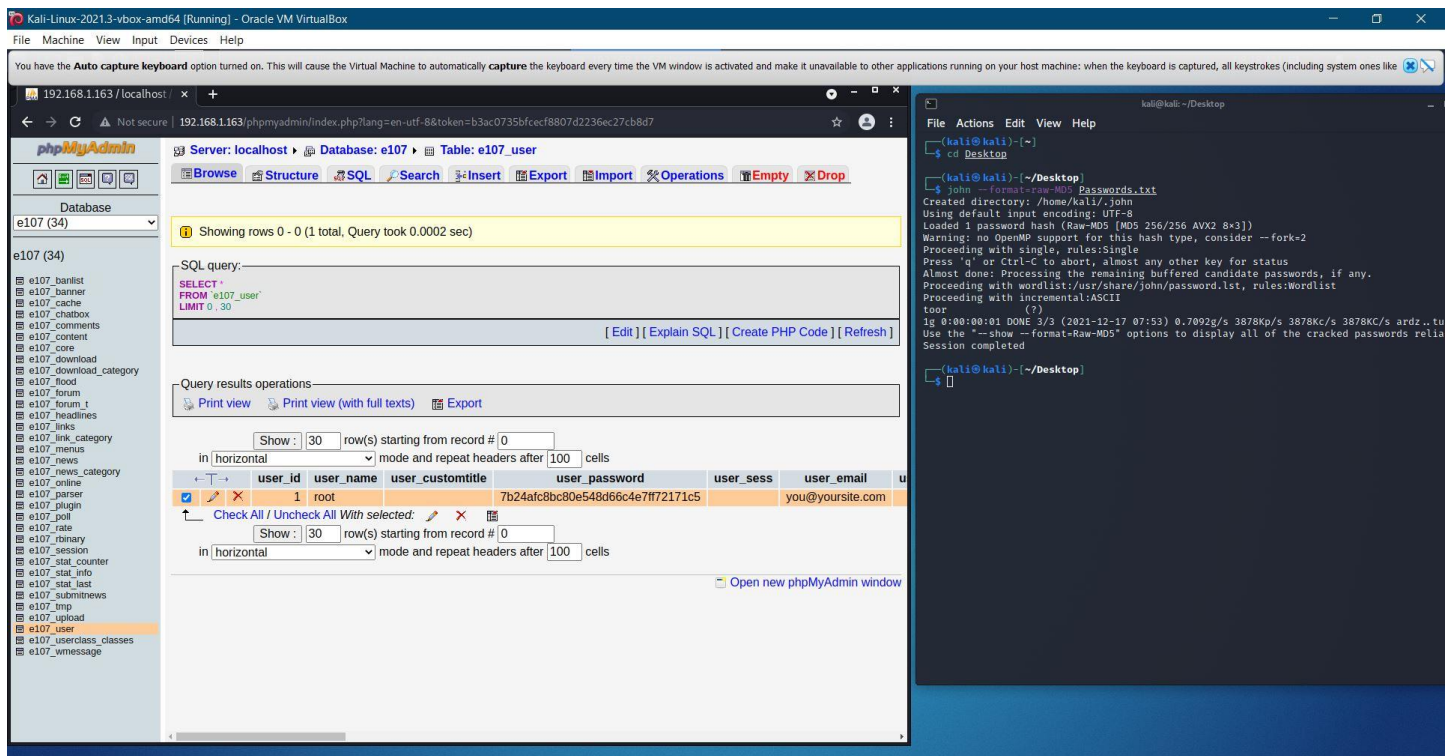


Fig 7.0: John the Ripper Password descriptor tool

Name: Aaron Mascarenhas
Module: Computer Systems Security

3.0.2 DOS Attack

As there were no further vulnerabilities to exploit within the system, the system was deemed to be compromised enough. So, the final exploit to test was a DOS attack. To carry out a DOS attack, an attack tool had to be utilized, thus the Metasploit tool was downloaded. The DOS attack command was given and initialized attacking the given IP address. As you can see in figure 8, the webpage was opened up alongside the attack to capture the success/failure of the attack.

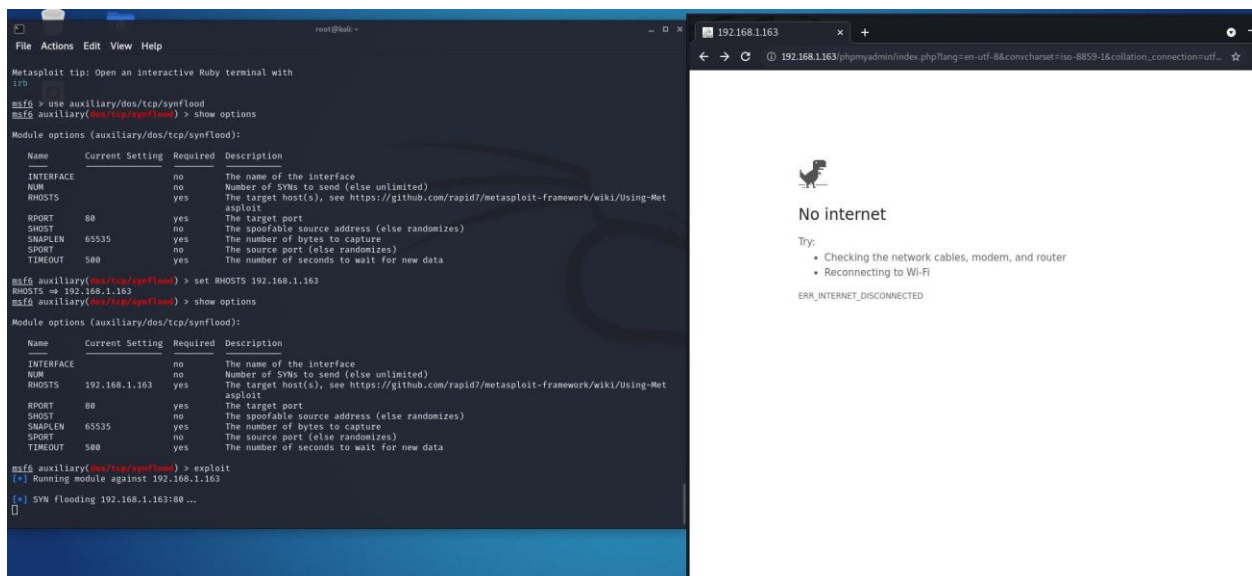


Fig 8.0: Dos attack using Metasploit to bring down webserver

3.0.3 Directory traversal attack using PuTTY

To access the targets directory, I've used PuTTY to Telnet to the Target's IP address, which is "192.168.1.163", the port is set to 22, and the connection type is set to SSH, this information was found using Nmap which is shown in figure 1.

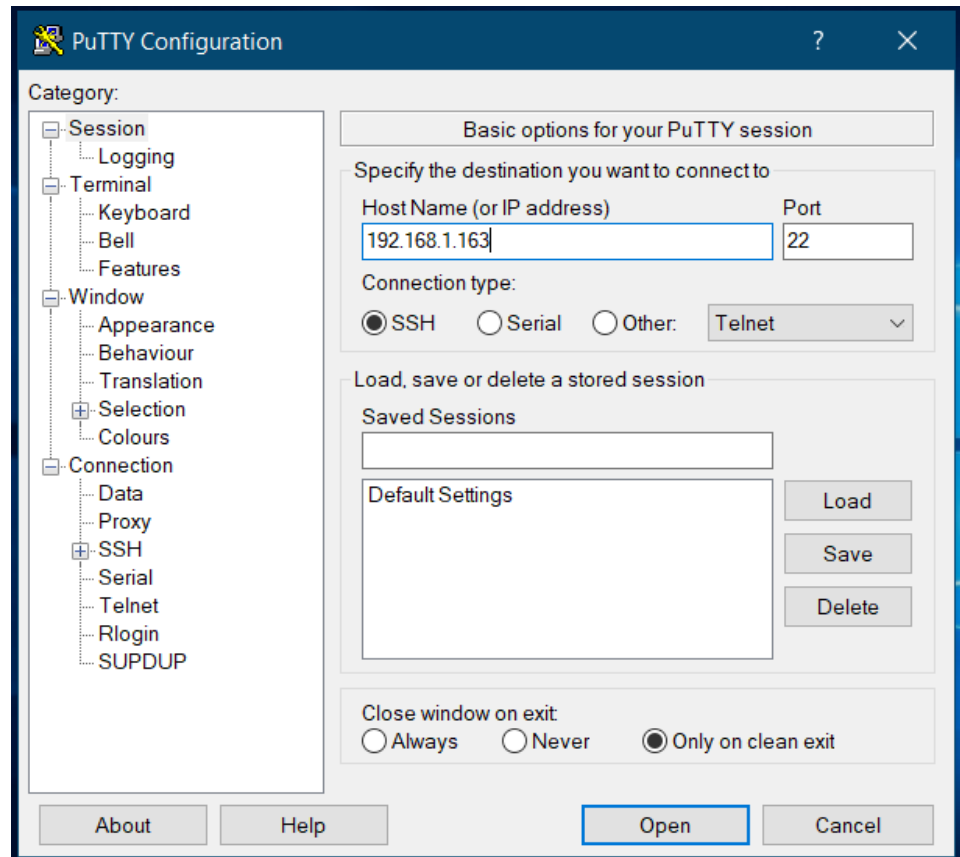


Fig 9: PuTTY Telnet connection to Target IP address

After entering all the configuration settings into PuTTY it connects remotely to the target's IP address and displays a terminal with asks to the login using the client's information which has been discovered in the "**Browser Exploitation**" section where the client's credentials were leaked.

Name: Aaron Mascarenhas
Module: Computer Systems Security

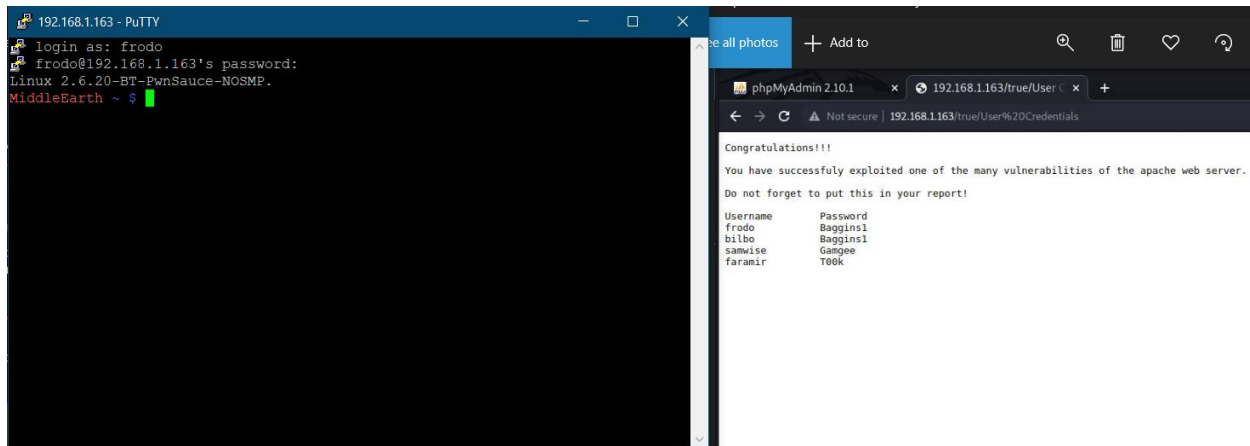


Fig 10: PuTTY Login using User credentials

The terminal prompts us to enter the client's credentials to log in and upon entering the username and password that was found in the browser directory. After logging into the client's account we gain access to all the libraries including the home directories, root directory, system files, etc., and have access to other clients that are related to the network.

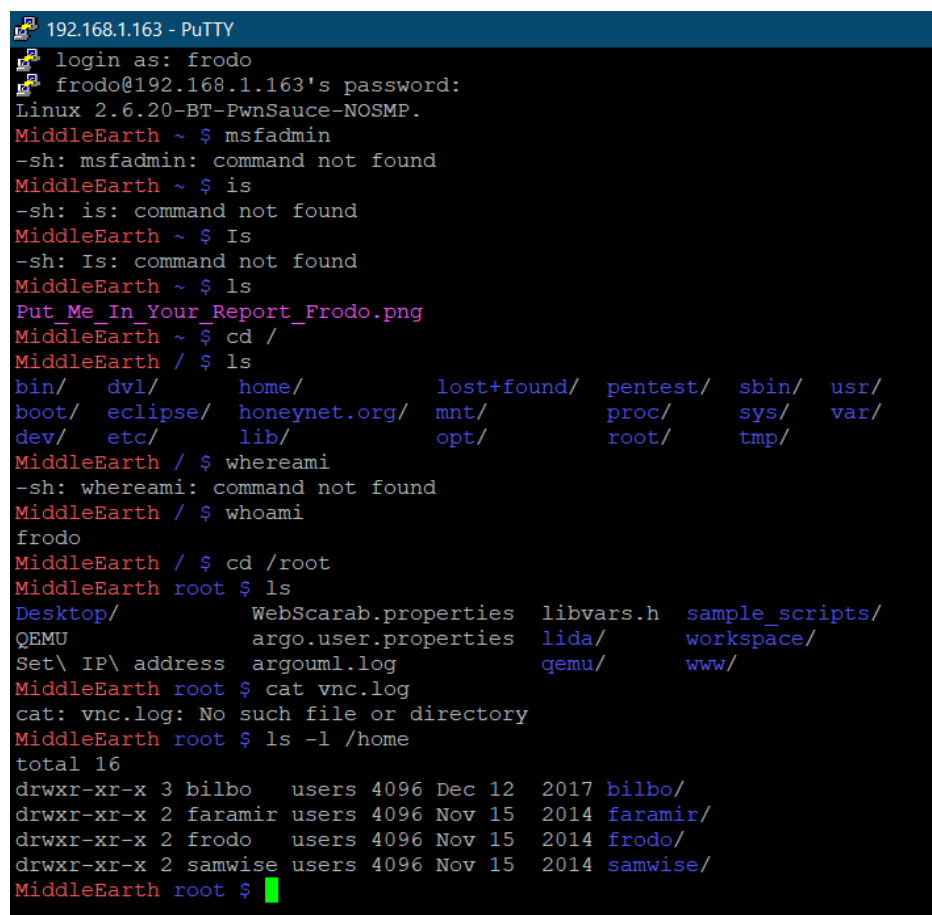


Fig 11: List of users linked with the with root directories with different paths

4.0 Vulnerability Mitigation

4.0.1 Browser Exploitation – Apache Web Server

Risk Rating: **High**

Description –

The apache web server had been hacked, giving hackers access to accounts and passwords. This demonstrates that a hacker may easily acquire access to a web server's list of safe usernames and passwords and that the hacker will be able to access and discover them even if the passwords and usernames are updated and altered.

Mitigation –

To address this, the directory listing should be disabled, the apache software should be hidden from all but the administrator, the apache software should be updated regularly, and the usernames and passwords should be changed and encrypted ONLY after all of this have been done.

4.0.2 Default Credentials for PHPMyAdmin login

Risk Rating: **High**

Description -

The default username and password are being used and unmodified on the login page to PHPMyAdmin. This can result in any hacker, who has information that the system is utilizing PHPMyAdmin to “have a go” and employ the default credentials and access the database. The database would provide the hacker with very sensitive information, such as encrypted passwords, all domain name accounts, server passwords, and the capacity to delete all material within the databases, as well as the ability to demand ransom for the release of this data to the administrator.

Mitigation –

Admin/IT Security personnel can prevent root user access, alter the default PHPMyAdmin access URL, set a secure password, make frequent backups, and maintain software up-to-date to safeguard the website.

4.0.3 DOS Attack

Risk Rating: **High**

Description -

The system had been rendered inaccessible as a result of a DOS assault. Users were unable to access the system, resulting in downtime and, in the case of a business, earnings loss owing to the inability to use the system.

Mitigation –

Set up firewalls and routers and look at black hole routing.

4.0.4 PuTTY - Directory traversal attack

Risk Rating: **High**

Description -

The network was remotely connected to gain access to the server directories using the client's login details to find valuable data that was stored in the server webpage folders which contained client system files, lost + found files, temporary files, root files, user files, developer files, library files, etc. with could get leaked if it falls into the wrong hands.

Mitigation –

Set up network firewalls and encrypt data files that scramble data into unreadable text format to avoid data breaches.

5.0 Conclusions

The system that was the subject of this penetration test has revealed that it has several exploits, the majority of which can render the system useless or non-existent if attackers use the exploits described. Due to the lack of encryption of usernames and passwords, as well as PHPMyAdmin's default user credentials, the system was extremely easy to access. what actions should be taken, and so forth.

6.0 Overall Conclusions and Reflections

I've learned a variety of abilities and approaches that a pen tester, as well as a computer scientist and a student of computer science, should have. I was able to gain knowledge and understanding (as defined by the University of Hertfordshire School of Physics, Engineering, and Computer Science) of a variety of current computer security techniques, as well as how the principles of systems security methods are embodied within them; essential facts, concepts, and principles of systems requirements for secure operations and practices, as well as computer systems risks, vulnerabilities, threats analysis, and software security (Nashi, G, 2021)

7.0 References

Herts.instructure.com. 2022. *CSS Guide*. [online] Available at:
<https://herts.instructure.com/courses/90977/pages/assignment-3-guidance?module_item_id=1696572> [Accessed 6 January 2022].

Openwall.com. 2022. *John the Ripper password cracker*. [online] Available at:
<<https://www.openwall.com/john/>> [Accessed 6 January 2022].

Palo Alto Networks. 2022. *What is a denial of service attack (DoS) ?*. [online] Available at:
<[https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos#:~:text=A%20Denial%2Dof%2DService%20\(information%20that%20triggers%20a%20crash.>](https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos#:~:text=A%20Denial%2Dof%2DService%20(information%20that%20triggers%20a%20crash.>)> [Accessed 6 January 2022].

En.wikipedia.org. 2022. *Browser exploit - Wikipedia*. [online] Available at:
<https://en.wikipedia.org/wiki/Browser_exploit> [Accessed 6 January 2022].

Group, D., 2022. *Apache HTTP Server 1.3 vulnerabilities - The Apache HTTP Server Project*. [online] Httpd.apache.org. Available at: <https://httpd.apache.org/security/vulnerabilities_13.html> [Accessed 6 January 2022].

Pretag. 2022. *Phpmyadmin default password*. [online] Available at:
<<https://pretagteam.com/question/phpmyadmin-default-password>> [Accessed 6 January 2022].

8.0 Appendix

8.0.1 Kali Update Download

Kali Linux was updated before proceeding with the penetration testes, using the terminal provided in kali I was able to run commands to download the update files on kali to the latest version.

```
(root@kali)~# sudo apt update
Get:1 http://kali.download/kali kali-rolling InRelease [30.6 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [17.8 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [39.6 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [114 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [152 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [209 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [963 kB]
Fetched 58.9 MB in 19s (3,064 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
956 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Fig 12: Updating Kali to the latest version

8.0.2 Kali Update Install

After the download for the update files were completed, the next step was to install these update files on kali machine.

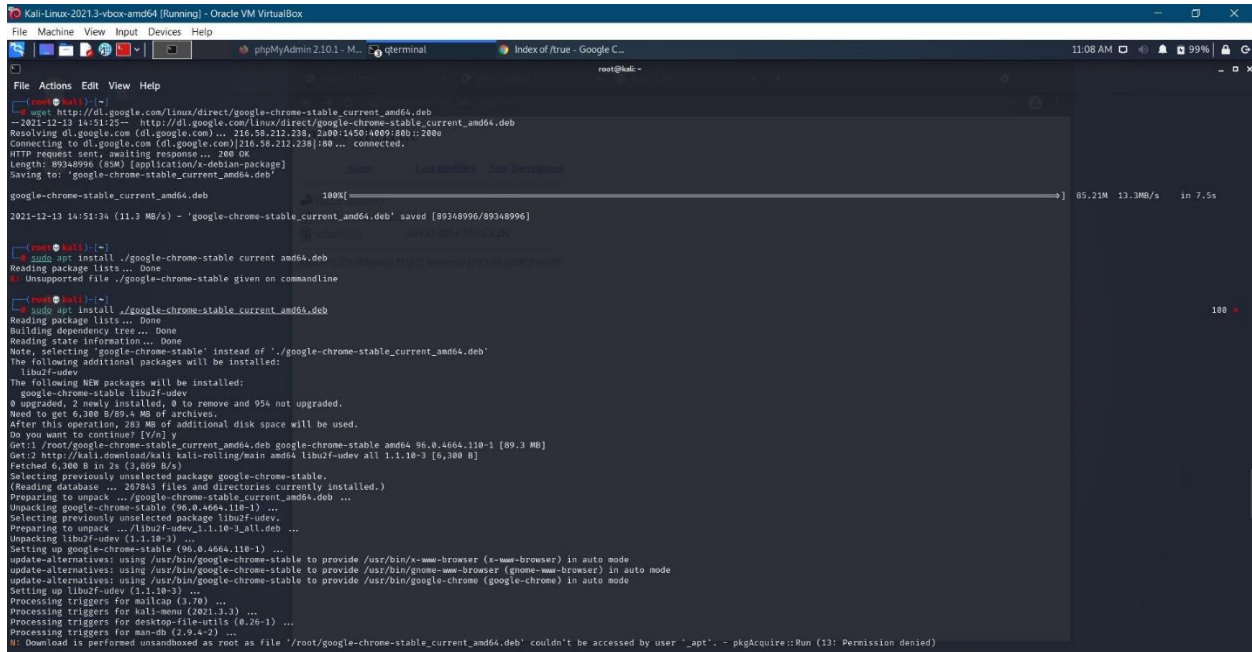
```
(root@kali)~# sudo apt -y install wget
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgnutls30
Suggested packages:
  gnutls-bin
The following packages will be upgraded:
  libgnutls30 wget
2 upgraded, 0 newly installed, 0 to remove and 954 not upgraded.
Need to get 2,326 kB of archives.
After this operation, 119 kB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 libgnutls30 amd64 3.7.2-2 [1,350 kB]
Get:2 http://http.kali.org/kali kali-rolling/main amd64 wget amd64 1.21.2-2+b1 [976 kB]
Fetched 2,326 kB in 2s (1,223 kB/s)
(Reading database ... 267842 files and directories currently installed.)
Preparing to unpack .../libgnutls30_3.7.2-2_amd64.deb ...
Unpacking libgnutls30:amd64 (3.7.2-2) over (3.7.1-5) ...
Setting up libgnutls30:amd64 (3.7.2-2) ...
(Reading database ... 267842 files and directories currently installed.)
Preparing to unpack .../wget_1.21.2-2+b1_amd64.deb ...
Unpacking wget (1.21.2-2+b1) over (1.21-1+b1) ...
Setting up wget (1.21.2-2+b1) ...
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for kali-menu (2021.3.3) ...
Processing triggers for libc-bin (2.31-13) ...
```

Fig 13: Installing update files for Kali

Name: Aaron Mascarenhas
Module: Computer Systems Security

8.0.3 Installing Chrome Browser on Kali

In order to test the network and open links to webpages as well as running Nessus vulnerability scan on the server, chrome browser was required to be setup and installed on the kali machine,



```
root@kali:~# wget http://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
--2021-12-13 14:51:22-- http://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
Resolving dl.google.com (dl.google.com)... 216.58.212.238, 2001:480:800::2000
Connecting to dl.google.com (dl.google.com)|216.58.212.238|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 89348996 (85M) [application/x-debian-package]
Saving to: 'google-chrome-stable_current_amd64.deb'

google-chrome-stable_current_amd64.deb
100%[=====] 85.21M 13.3MB/s in 7.5s

2021-12-13 14:51:34 (11.3 MB/s) - 'google-chrome-stable_current_amd64.deb' saved [89348996/89348996]

root@kali:~# curl -O http://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 85.2M  100 85.2M    0     0  13.3M      0  0:00:07  0:00:07 --:--:-- 13.3M

root@kali:~# sudo apt install ./google-chrome-stable_current_amd64.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'google-chrome-stable' instead of './google-chrome-stable_current_amd64.deb'
The following additional packages will be installed:
  libu2f-dev
The following NEW packages will be installed:
  google-chrome-stable libu2f-dev
0 upgraded, 2 newly installed, 0 to remove and 954 not upgraded.
Need to get 6,380 B/89.4 MB of archives.
After this operation, 281 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 /root/google-chrome-stable_current_amd64.deb google-chrome-stable amd64 96.0.4664.110-1 [89.3 MB]
Get:2 http://kali.download/kali kali-rolling/main amd64 libu2f-dev all 1.1.10-3 [6,380 B]
Fetched 6,380 B in 2s (3,859 B/s)
Selecting previously unselected package google-chrome-stable.
(Reading database ... 267843 files and directories currently installed.)
Preparing to unpack .../google-chrome-stable_current_amd64.deb ...
Unpacking google-chrome-stable (96.0.4664.110-1) ...
Selecting previously unselected package libu2f-dev.
Preparing to unpack .../libu2f-dev_1.1.10-3_all.deb ...
Unpacking libu2f-dev (1.1.10-3) ...
Setting up google-chrome-stable (96.0.4664.110-1) ...
update-alternatives: using /usr/bin/google-chrome-stable to provide /usr/bin/x-www-browser (x-www-browser) in auto mode
update-alternatives: using /usr/bin/google-chrome-stable to provide /usr/bin/gnome-www-browser (gnome-www-browser) in auto mode
Setting up libu2f-dev (1.1.10-3) ...
Processing triggers for mailcap (3.70) ...
Processing triggers for kali-menu (2021.3.3) ...
Processing triggers for desktop-file-utils (0.26-1) ...
Processing triggers for man-db (2.9.4-2) ...
N: Download is performed unsandboxed as root as file '/root/google-chrome-stable_current_amd64.deb' couldn't be accessed by user '_apt'. - pkgAcquire::Run (13: Permission denied)
```

Fig 14: Downloading Chrome on kali