Defensa1_antes_educerecer_cerrar_NFS

February 4, 2026

1 Endurecimiento del laboratorio (Hardening guiado y didáctico)

1.1 Objetivo del endurecimiento

Convertir el laboratorio en un escenario **más realista**, donde: - No sea trivial escalar a root - No se pueda leer todo el sistema - Siga siendo posible **exfiltrar una carpeta de un usuario** - El atacante tenga que **pensar y elegir técnicas**

1.2 1 Endurecimiento de NFS (infraestructura crítica)

1.2.1 Situación anterior

- NFS exportando `/`
- Sin `root_squash`
- Escritura completa desde el cliente

Esto permitía: - Leer `/etc/passwd` y `/etc/shadow` - Crear usuarios root - Modificar binarios SUID

1.2.2 Medida defensiva

Exportar **solo una carpeta concreta del usuario objetivo** y limitar permisos.

Archivo: `/etc/exports` “‘bash /home/msfadmin/share 192.168.56.0/24(ro,root_squash,no_subtree_check)

2 Endurecimiento de NFS — Paso a paso (defensivo)

2.1 Contexto

NFS estaba permitiendo: - Montar `/` completo - Acceder como root remoto - Leer y modificar archivos críticos

Objetivo: - Evitar montaje de `/` - Evitar root remoto - Permitir **solo** acceso controlado a una carpeta concreta

---

2.2 PASO 1 — Identificar qué se está exportando ahora mismo

2.2.1 Qué hacemos

Comprobamos **qué directorios NFS está exportando el servidor** y con qué opciones. Esto nos dice **por qué el ataque funcionaba**.

2.2.2 Comando (en la víctima)

“‘bash cat /etc/exports Dentro vemos la clave / *(rw,sync,no_root_squash) 1. / Todo el sistema 2.* Cualquier IP 3. rw lectura y esctritura 4 no_root_squash=root remoto

2.3 PASO 2 — Corregir la exportación NFS (cerrar la brecha)

2.3.1 Qué vamos a hacer

Vamos a **eliminar la exportación insegura** y reemplazarla por una **exportación mínima y controlada**.

Objetivo: - No exportar `/` - No permitir root remoto - Exportar solo una carpeta concreta - Solo a una IP concreta (Kali)

2.3.2 2.1 Editar `/etc/exports`

Archivo: “‘bash nano /etc/exports

Línea insegura actual (ELIMINAR o COMENTAR) Copiar código Text / *(rw,sync,no_root_squash) 2.2 Crear una exportación segura Ejemplo seguro: Copiar código Text /home/msfadmin/shared 192.168.56.102(rw,sync,root_squash,no_subtree_check) Qué significa cada opción /home/msfadmin/shared → solo esa carpeta 192.168.56.102 → solo tu Kali rw → lectura/escritura (opcional) root_squash → root remoto pasa a nobody no_subtree_check → evita errores NFS comunes 2.3 Crear la carpeta compartida (si no existe) Copiar código Bash mkdir -p /home/msfadmin/shared chown msfadmin:msfadmin /home/msfadmin/shared chmod 750 /home/msfadmin/shared

Confirmamos comabios con sudo exportfs -ra y comprobamos:

showmount -e localhost
Export list for localhost:
/home/msfadmin/192.168.56.102

2.4 Paso 2 — Verificar que el endurecimiento NFS funciona de verdad

Ahora vamos a **pensar como atacantes** y comprobar que lo que acabamos de hacer **rompe el ataque anterior**.
Esto es clave para que el laboratorio sea didáctico.

2.4.1 1 Volver al atacante (Kali)

Nos situamos en Kali, **como si no supiéramos nada**:

“‘bash showmount -e 192.168.56.101

3 Mapa de Endurecimiento: Filtrado por IP y por Firmware

Este mapa sirve como **guía defensiva** para limitar el acceso a servicios (especialmente NFS/RPC) filtrando **por IP** y **por tipo/versión de firmware o sistema**.
La idea es reducir superficie de ataque incluso aunque un servicio siga activo.

3.1 1 Filtrado por IP (primera línea de defensa)

3.1.1 Objetivo

Permitir acceso **solo a hosts legítimos** y bloquear todo lo demás.

3.1.2 Nivel 1: Firewall del sistema (iptables / nftables)

Concepto - Solo IPs autorizadas pueden hablar con el servicio - Todo lo demás se descarta silenciosamente

Ejemplo conceptual (iptables) “‘bash # Permitir NFS solo desde red interna iptables -A INPUT -p tcp -s 192.168.56.0/24 –dport 2049 -j ACCEPT iptables -A INPUT -p udp -s 192.168.56.0/24 –dport 2049 -j ACCEPT

4 Bloquear resto

iptables -A INPUT -p tcp –dport 2049 -j DROP iptables -A INPUT -p udp –dport 2049 -j DROP

4.1 Ejemplo REAL aplicado a NUESTRO CASO (laboratorio)

4.1.1 Contexto del laboratorio

- **Víctima (servidor NFS / Metasploitable):** `192.168.56.101`
- **Host legítimo (cliente autorizado):** `192.168.56.102`
- **Atacante (Kali):** cualquier otra IP de la red (`192.168.56.0/24`)

Objetivo:
Que **SOLO** `192.168.56.102` pueda acceder a NFS/RPC
Que **Kali NO vea ni enumere nada**, aunque NFS esté activo

4.2 Filtrado por IP con iptables (caso real)

4.2.1 1 Política base (cerrar todo)

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
```

Demasiado agresiva en general, cierra todos los puertos con todos los protoclos ### Permitir loopback

```
iptables -A INPUT -i lo -j ACCEPT
```

Necesario para que el sistema funcione internamente. ### Permitir conexiones ya establecidas

```
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

4.2.2 Permitir RPC (111) SOLO desde el host autorizado

```
iptables -A INPUT -p tcp -s 192.168.56.102 --dport 111 -j ACCEPT
iptables -A INPUT -p udp -s 192.168.56.102 --dport 111 -j ACCEPT
```

tanto tcp como udp. RCS necesario para NFS pero solo para el cliente válido

4.2.3 Permitir NFS (2049, ver ataque 1) SOLO desde el host autorizado

```
iptables -A INPUT -p tcp -s 192.168.56.102 --dport 2049 -j ACCEPT
iptables -A INPUT -p udp -s 192.168.56.102 --dport 2049 -j ACCEPT
```

4.2.4 Bloqueo explícito para el resto

```
iptables -A INPUT -p tcp --dport 111 -j DROP
iptables -A INPUT -p udp --dport 111 -j DROP
iptables -A INPUT -p tcp --dport 2049 -j DROP
iptables -A INPUT -p udp --dport 2049 -j DROP
```

Esto oculta el servicio aunque esté activo TODO CON SUDO. Hacemos **sudo iptables -L -n** para ver las reglas

4.2.5 Salida desde atacante

```
  (kali kali)-[~]
 $ sudo nmap -sV 192.168.56.101
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-04 05:52 EST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid se
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --s
Nmap scan report for 192.168.56.101
Host is up (0.00022s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE    SERVICE     VERSION
21/tcp   open     ftp         vsftpd 2.3.4
22/tcp   open     ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open     telnet      Linux telnetd
```

```
25/tcp   open     smtp        Postfix smtpd
53/tcp   open     domain      ISC BIND 9.4.2
80/tcp   open     http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  filtered rpcbind
139/tcp  open     netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open     netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open     exec        netkit-rsh rexecd
513/tcp  open     login       OpenBSD or Solaris rlogind
514/tcp  open     shell       Netkit rshd
1099/tcp open     java-rmi    GNU Classpath grmiregistry
1524/tcp open     bindshell   Metasploitable root shell
2049/tcp filtered nfs
2121/tcp open     ftp         ProFTPD 1.3.1
3306/tcp open     mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open     postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open     vnc         VNC (protocol 3.3)
6000/tcp open     X11         (access denied)
6667/tcp open     irc         UnrealIRCd
8009/tcp open     ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open     http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:EC:73:E2 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.81 seconds
```

LOS PUERTOS SELECCIONADOS ESTÁN FILTRADOS.

5 Tabla resumen de iptables — comandos, usos y tips clave

Esta tabla recopila **todas las opciones que hemos usado** para **ver**, **añadir**, **borrar**, **ordenar** y **entender** reglas de `iptables`, con notas prácticas para no romper el sistema.

5.1 Visualización y auditoría de reglas

Comando	Para qué sirve	Tip importante
`iptables -L`	Lista reglas (vista básica)	No muestra números ni contadores
`iptables -L INPUT`	Lista solo la cadena INPUT	Útil para auditar tráfico entrante
`iptables -L -n`	Evita resolución DNS	Más rápido y claro
`iptables -L -v`	Muestra contadores (pkts/bytes)	Confirma si la regla se está usando
`iptables -L --line-numbers`	Añade números de regla	Imprescindible para borrar por índice

Comando	Para qué sirve	Tip importante
`iptables -S`	Muestra reglas “crudas”	Ideal para documentación
`iptables-save`	Muestra reglas persistentes	Verifica lo que sobrevivirá a reboot

---

5.2 Añadir reglas (ACCEPT / DROP)

Comando	Uso	Tip crítico
`iptables -A INPUT ... -j ACCEPT`	Añade regla al final	El orden importa
`iptables -I INPUT 1 ... -j ACCEPT`	Inserta en la posición 1	Útil para loopback
`-p tcp / -p udp`	Filtra por protocolo	RPC/NFS usan ambos
`--dport 111`	Filtra puerto rpcbind	RPC = vector de enumeración
`--dport 2049`	Filtra puerto NFS	Montaje de filesystem
`-s 192.168.56.102`	Limita por IP origen	Endurecimiento selectivo
`-i lo`	Interfaz loopback	**Nunca bloquear localhost**
`-j DROP`	Descarta silenciosamente	El atacante no recibe feedback

5.3 Borrado de reglas (limpieza segura)

Comando	Para qué sirve	Cuándo usarlo
`iptables -D INPUT 3`	Borra regla nº 3	Método más seguro
`iptables -D INPUT ...`	Borra por coincidencia exacta	Solo si conoces la regla
`iptables -F INPUT`	Vacía toda la cadena INPUT	Limpieza rápida
`iptables -F`	Borra todas las reglas	Reset completo
`iptables -X`	Borra cadenas custom	Tras `-F`

5.4 Políticas por defecto (usar con cuidado)

Comando	Efecto	Advertencia
`iptables -P INPUT ACCEPT`	Todo permitido por defecto	Modelo permisivo
`iptables -P INPUT DROP`	Todo bloqueado por defecto	Muy agresivo
`iptables -P OUTPUT ACCEPT`	Salidas permitidas	Evita romper el sistema
`iptables -P FORWARD DROP`	No enrutar tráfico	Recomendado

5.5 Tips fundamentales (MUY importantes)

Regla mental	Explicación
El orden importa	iptables evalúa de arriba a abajo
DROP antes de ACCEPT	La ACCEPT nunca se ejecuta
ACCEPT antes de DROP	Flujo correcto
Localhost pasa por INPUT	Si lo bloqueas, rompes servicios
Usa contadores	Confirma que la regla funciona
Prueba local y remota	Servicio vivo accesible
Guarda cambios	Si no, se pierden al reiniciar

5.6 Ejemplo de orden CORRECTO (mental)

1. ACCEPT loopback
2. ACCEPT IP legítima
3. DROP puerto sensible

Si inviertes 2 y 3 → **rompes el acceso legítimo**

5.7 Intento de acceso remoto con mount

```
[ ]:
```