Ataque_2_Bindshell

February 10, 2026

0.0.1 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 ser
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --sy
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
```

El siguiente paso es atacar el 1524 (luego cerramos cuando hayamos terminado con el ataque) que es un rootshell abierto—> Acceso directo

1 Comparativa: nc (Netcat) vs telnet — mentalidad de atacante

Esta tabla compara **nc** y **telnet** desde el punto de vista de un atacante, pensando en **sigilo**, **flexibilidad** y **utilidad real** en escenarios como el puerto 1524.

1.1 Tabla comparativa

Característica	nc (netcat)	telnet
Propósito original	Herramienta de red multipropósito	Cliente de acceso remoto
Flexibilidad	Muy alta	Baja
Protocolos soportados	TCP y UDP	Solo TCP
Uso típico del atacante	Conexión cruda, shells, exfiltración, pivoting	Acceso interactivo simple
Nivel de ruido	Bajo	Medio
Dependencia de servicio	Ninguna	Requiere servicio telnet
Manejo de datos binarios	Sí	No
Redirecciones y pipes	Sí	No
Automatización / scripting	Fácil	Difícil
Presencia en sistemas antiguos	Habitual	Muy habitual
Uso con bindshell	Sí	Sí, pero limitado
Uso con reverse shell	Sí	No
Cifrado	No	No
Facilidad de detección (IDS)	Menor	Mayor
Huella en logs	Menor	Mayor
Uso defensivo (debug)	Frecuente	Poco común
Estado actual	Herramienta estándar	Obsoleta

1.2 Lectura atacante (interpretación)

- **nc (netcat)** es la herramienta preferida:
  - Permite interactuar con cualquier puerto sin asumir protocolo
  - Sirve tanto para acceso como para exfiltración
  - Facilita automatización y ataques sigilosos
- **telnet** es útil solo para:
  - Comprobaciones rápidas
  - Sistemas muy antiguos
  - Accesos manuales simples

En un puerto como **1524**, un atacante real usaría **nc** primero y telnet solo como respaldo.

1.3 Implicación para los siguientes pasos

- Si un servicio responde correctamente a `nc`, la superficie de ataque es mayor.
- Defender contra `nc` implica:
  - cerrar el puerto
  - eliminar el servicio que lo lanza
  - no solo filtrar clientes telnet

Con esto claro, el siguiente paso lógico es decidir **cómo mitigar 1524**: - cerrar el servicio - eliminar el binario - o filtrar por firewall para estudiar el impacto

1.4 Ataque con nc a la máquina victima, intentamos directamente conexion al shell abierta

```
┌──(kali㉿kali)-[~]
└─$ nc 192.168.56.101 1524

root@metasploitable:/# whoami
root
```

Accedemos al puerto y la shell como root directamente. Ya podemos exfiltrar si queremos

2 HARDERING DEL PUERTO 1524 (BINDSHELL)

Vamos a suponer que es un servicio que tiene que estar levantado, en vez de filtrar por IP, como hemos hecho en el caso anterior, vamos a que podríamos hacerlo para endurecer aún más, vamos a eliminar el acceso remoto sin autenticación. Como primer paso, para mitigar mientras, añadimos regla al firewall (desde víctima)

```
iptables -A INPUT -p tcp --dport 1524 -j DROP
```

Así impedimos conexión tcp a ese puerto mientras corregimos. La shell está ahora bloqueada

```
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 filtered ingreslock
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)
```

Vamos a usar socketstatistic para entender qué o quién a abierto dicho proceso # Resumen del comando `ss` — uso y significado (para JupyterLab)

Esta tabla resume **qué hace `ss`**, por qué se usa y qué información aporta, en el contexto de análisis de puertos y hardening (ej. puerto 1524).

2.1 Tabla resumen

Elemento	Significado	Por qué se usa aquí
`ss`	Socket Statistics	Muestra el estado real de los sockets del sistema
`-l`	Listening	Filtra solo puertos en escucha (servicios activos)
`-n`	Numeric	Evita DNS / resolución de servicios, salida exacta
`-t`	TCP	El puerto 1524 usa TCP
`-u`	UDP	Incluido por completitud (evita repetir comando)
`-p`	Process	Muestra PID y binario que abrió el puerto
`|`	Pipe	Redirige salida a otro comando
`grep 1524`	Filtro de texto	Aísla el puerto sospechoso
Resultado	Puerto + proceso	Identifica la causa real del servicio

---

2.2 Qué información crítica revela

Dato obtenido	Utilidad
Puerto exacto	Confirma superficie de ataque
Dirección (0.0.0.0 / ::)	Indica exposición externa
PID	Permite matar o investigar el proceso
Binario	Identifica el servicio real
Usuario	Confirma privilegios (root / no root)

2.3 Ejemplo típico de salida

```
LISTEN 0 128 0.0.0.0:1524 0.0.0.0:* users:(("sh",pid=1234,fd=3))
```

También, en este caso mejor, con **netsat -tulpn**

El proceso es 4485 que con

```
ps -fp 4485
```

Nos da toda la información (es de root el proceso, algo que ya sabiamos). El PPID es 1 por lo que fue lanzado por un init, en este caso **xinetd** Vamos a revisar los servicioes de xinetd.d, dentro de etc uno por uno hasta encontrar un servicio que esté habilitado y corra como root. (Nota: esto podríamos automatizarlo aunque estamos corriendo en una máquina vieja). El servicio ha resultado ser vsftpd versión 2.3.4 que por lo que es puerta trasera reconocida. Desactivamos este servicio de xinetd (disabled=yes) y restablecemos el servicio /etc/init.d/xinetd restart.
Comprubeo desde kali, **sin desactivar la regla de firewall**

```
PORT     STATE    SERVICE     VERSION
22/tcp   open     ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
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)
1099/tcp open     java-rmi    GNU Classpath grmiregistry
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)
```

Desactivo el firewall, esa regla en concreto ya que ha desaparecido el puerto, está desactivado

[ ]: