



UNIVERSIDAD DEL ISTMO

UNISTMO TEHUANTEPEC

DOCENTE:

ING. CARLOS MIJANGOS.

ASIGNATURA:

REDES DE COMPUTADORAS II.

TEMA:

ANÁLISIS DE ESCANEOS Y MEDIDAS DEFENSIVAS

ALUMNOS:

JEOVANI PACHECO RUEDA

CARRERA:

INGENIERÍA EN COMPUTACIÓN.

GRUPO:

704.

SANTO DOMINGO TEHUANTEPEC, OAXACA; 20 DE OCTUBRE DE 2025.

Índice

Introducción	1
Creación de Sandbox	2
Escaneos con NMAP	3
Medidas de seguridad	6
¿Por qué usar Wireshark?	10
Conclusiones	12
Referencias bibliográficas	13

1. Introducción

En el campo de la ciberseguridad, comprender cómo se comunican los sistemas y cómo se exponen sus servicios es esencial para proteger cualquier infraestructura digital. Antes de poder defender una red, es necesario conocerla a fondo: qué equipos la integran, qué servicios ofrecen y qué puertas de entrada pueden representar un riesgo. Este principio guía las auditorías de seguridad, donde las herramientas de escaneo y monitoreo se convierten en los ojos del analista.

El propósito de esta práctica fue explorar el funcionamiento y la eficacia de dos herramientas fundamentales en la auditoría de redes: Nmap (Network Mapper) [1] y Wireshark [2], aplicadas tanto desde una perspectiva ofensiva como defensiva. El ejercicio se desarrolló dentro de un entorno controlado o sandbox, lo que permitió simular ataques de reconocimiento de forma segura y luego observar sus efectos en tiempo real.

Nmap es una herramienta de código abierto ampliamente utilizada para el descubrimiento de hosts, servicios y puertos dentro de una red. Su capacidad para identificar sistemas operativos, versiones de software y configuraciones expuestas la convierte en un recurso esencial para evaluar la superficie de ataque y localizar posibles vulnerabilidades.

Por otro lado, Wireshark actúa como un microscopio del tráfico de red: un analizador de protocolos capaz de capturar, decodificar y examinar cada paquete que circula por una interfaz. Gracias a su nivel de detalle y a la potencia de sus filtros, permite detectar comportamientos anómalos, rastrear escaneos y correlacionar eventos sospechosos con su origen.

Para el correcto desarrollo de la práctica se requieren permisos administrativos, conectividad local y la configuración de la tarjeta de red en modo promiscuo, permitiendo la captura completa del tráfico entre los equipos del entorno de prueba. Con estas condiciones establecidas, el alumno pudo observar de forma práctica cómo las acciones de reconocimiento se reflejan en el tráfico de red y cómo una correcta monitorización puede servir como defensa temprana ante intentos de intrusión.

2. Creación del Sandbox

Para realizar la práctica de forma segura, se configuró un entorno controlado (sandbox) [3] dentro de una red local. Este espacio aislado permitió ejecutar escaneos sin afectar sistemas reales ni vulnerar políticas de red. Se utilizaron dos equipos: uno actuó como atacante con Nmap, y el otro como víctima y monitor de tráfico mediante Wireshark, ambos conectados bajo la misma subred.

IP ATACANTE: 192.168.8.57

```
jeovani@jeovani-PB: ~  
64 bytes from 192.168.8.48: icmp_seq=3 ttl=64 time=0.282 ms  
64 bytes from 192.168.8.48: icmp_seq=4 ttl=64 time=0.423 ms  
--- 192.168.8.48 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3039ms  
rtt min/avg/max/mdev = 0.282/2.997/10.991/4.615 ms  
jeovani@jeovani-PB: ~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000  
    link/ether 0c:9a:3c:55:f7:8f brd ff:ff:ff:ff:ff:ff  
    altname wlp0s20f3  
    inet 192.168.8.57/24 brd 192.168.8.255 scope global dynamic noprefixroute wlo1  
        valid_lft 82558sec preferred_lft 82558sec  
    inet6 fe80::ab1d:45db:50fe:1e0e/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
jeovani@jeovani-PB: ~$
```

IP VICTIMA: 192.168.8.48

```
Victima (SistemaOperativoVictima) [Corriendo] - Oracle VM VirtualBox  
Archivo Máquina Ver Entrada Dispositivos Ayuda  
To access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
No mail.  
msfadmin@metasploitable:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000  
    link/ether 08:00:27:67:8c:f4 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.8.48/24 brd 192.168.8.255 scope global eth0  
    inet6 fe80::a00:27ff:fe67:8cf4/64 scope link  
        valid_lft forever preferred_lft forever  
msfadmin@metasploitable:~$ ping -c 4 192.168.8.57  
PING 192.168.8.57 (192.168.8.57) 56(84) bytes of data.  
64 bytes from 192.168.8.57: icmp_seq=1 ttl=64 time=0.334 ms  
64 bytes from 192.168.8.57: icmp_seq=2 ttl=64 time=0.379 ms  
64 bytes from 192.168.8.57: icmp_seq=3 ttl=64 time=0.246 ms  
64 bytes from 192.168.8.57: icmp_seq=4 ttl=64 time=0.291 ms  
--- 192.168.8.57 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 2998ms  
rtt min/avg/max/mdev = 0.246/0.312/0.379/0.052 ms  
msfadmin@metasploitable:~$
```

Comprobamos que ambos equipos están conectados, y permiten hacer ping entre ellos.

3. Escaneos con Nmap (Máquina Atacante).

- **Ping scan:** Envía sondeo para ver qué hosts responden en la subred. No escanea puertos. Útil para saber qué máquinas están encendidas.

cmd: nmap -sn 192.168.8.48

```
jeovani@jeovani-PB:~$ nmap -sn 192.168.8.48
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-11 11:09 CST
Nmap scan report for 192.168.8.48
Host is up (0.00053s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.01 seconds
jeovani@jeovani-PB:~$
```

- **Escaneo TCP (Protocolo de Control de Transmisión):** Intenta completar la conexión TCP usando las llamadas normales del sistema operativo . Es más ruidoso que SYN [4] pero funciona sin privilegios y evita necesitar sudo.

nmap -sT -p 22,80,443 192.168.8.48

```
jeovani@jeovani-PB:~$ nmap -sT -p 22,80,443 192.168.8.48
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-11 11:18 CST
Nmap scan report for 192.168.8.48
Host is up (0.00052s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp    closed https

Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
jeovani@jeovani-PB:~$
```

- **Escaneo SYN:** Envía SYN y analiza respuestas sin completar el handshake [5]. Muy usado para descubrimiento rápido de puertos. Necesita permisos de root.

sudo nmap -sS -p 1-1000 192.168.8.48

```

jeovani@jeovani-PB:~$ sudo nmap -sS -p 1-1000 192.168.8.48
[sudo] contraseña para jeovani:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-11 11:22 CST
Nmap scan report for 192.168.8.48
Host is up (0.00045s latency).
Not shown: 988 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
MAC Address: 08:00:27:67:8C:F4 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
jeovani@jeovani-PB:~$

```

- **Detección de versión de servicio:** Intenta identificar software y versiones que corren en los puertos encontrados. Útil para inventario y para evaluar si hay servicios desactualizados.

sudo nmap -sV -p 22,80 192.168.8.48

```

jeovani@jeovani-PB:~$ sudo nmap -sV -p 22,80 192.168.8.48
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-11 11:23 CST
Nmap scan report for 192.168.8.48
Host is up (0.00047s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.2.8 ((Ubuntu) DAV/2)
MAC Address: 08:00:27:67:8C:F4 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.43 seconds
jeovani@jeovani-PB:~$

```

- **Detección de sistema operativo:** Recolecta huellas del stack TCP/IP para inferir el sistema operativo. No siempre es 100% fiable. Requiere permisos y genera tráfico que puede ser conspicuo.

sudo nmap -O 192.168.8.48

```

Nmap done: 1 IP address (1 host up) scanned in 7.43 seconds
jeovani@jeovani-PB:~$ sudo nmap -O 192.168.8.48
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-11 11:25 CST
Nmap scan report for 192.168.8.48
Host is up (0.00036s 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: 08:00:27:67:8C:F4 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

```

- **Escaneo UDP:** Prueba puertos UDP (DNS, DHCP, NTP). UDP es inherentemente más lento por retransmisiones y timeouts.

sudo nmap -sU -p 53,67,123 192.168.8.48

```

jeovani@jeovani-PB:~$ sudo nmap -sU -p 53,67,123 192.168.8.48
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-11 11:27 CST
Nmap scan report for 192.168.8.48
Host is up (0.00051s latency).

PORT      STATE SERVICE
53/udp    open  domain
67/udp    closed dhcpc
123/udp   closed ntp
MAC Address: 08:00:27:67:8C:F4 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
jeovani@jeovani-PB:~$ 

```

- **Escaneo “agresivo”:** Combina detección de OS, versión, scripts y traceroute. Muy informativo pero ruidoso, es sólo usado en laboratorio o con permiso explícito.

sudo nmap -A 192.168.8.48

```

jiovani@jiovani-PR: ~$ sudo nmap -A 192.168.8.48
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-11-11 11:29 CST
Nmap scan report for 192.168.8.48
Host is up (0.00056s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ ftp-syst:
|_ STAT:
|_ FTP server status:
|_   Connected to 192.168.8.57
|_   Logged in as ftp
|_   TYPE: ASCII
|_   No session bandwidth limit
|_   Session timeout in seconds is 300
|_   Control connection is plain text
|_   Data connections will be plain text
|_   vsFTPd 2.3.4 - secure, fast, stable
|_ End of status
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
|_ ssh-hostkey:
|_   1024 60:0f:cf:1e:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
|_ ssl-date: 2025-11-11T17:29:55+00:00: 0s from scanner time.
|_ smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITIME, DSN
|_ sslv2:
|_   SSLv2 supported
|_   ciphers:
|_     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|_     SSL2_DES_64_CBC_WITH_MD5
|_     SSL2_RC4_128_WITH_MD5
|_     SSL2_RC2_128_CBC_WITH_MD5
|_     SSL2_DES_192_EDE3_CBC_WITH_MD5
|_     SSL2_RC4_128_EXPORT40_WITH_MD5
53/tcp    open  domain       ISC BIND 9.4.2
|_ dns-nsid:
|_   bind.version: 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_ http-title: Metasploitable2 - Linux
|_ http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
111/tcp   open  rpcbind      2 (RPC #100000)

```

```

|_ servers: 0
|_ server: irc.Metasploitable.LAN
|_ version: Unreal3.2.8.1. irc.Metasploitable.LAN
|_ uptime: 0 days, 0:26:38
|_ source ident: nmap
|_ source host: FF6BEDD0.49132F3E.FFFA6049.IP
|_ error: Closing Link: uuczfvpwc[192.168.8.57] (Quit: uuczfvpwc)
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
|_ _ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
|_ http-title: Apache Tomcat/5.5
|_ http-favicon: Apache Tomcat
|_ http-server-header: Apache-Coyote/1.1
MAC Address: 08:00:27:67:8C:F4 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ smb-security-mode:
|_   account_used: <blank>
|_   authentication_level: user
|_   challenge_response: supported
|_   message_signing: disabled (dangerous, but default)
|_ nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_ smb-os-discovery:
|_   OS: Unix (Samba 3.0.20-Debian)
|_   Computer name: metasploitable
|_   NetBIOS computer name:
|_   Domain name: localdomain
|_   FQDN: metasploitable.localdomain
|_   System time: 2025-11-11T12:29:46-05:00
|_ clock-skew: mean: 1h15m00s, deviation: 2h30m00s, median: 0s
|_ smb2-time: Protocol negotiation failed (SMB2)

TRACEROUTE
HOP RTT ADDRESS
1 0.56 ms 192.168.8.48

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 23.26 seconds
jiovani@jiovani-PR: ~$

```

4. Medidas de seguridad (Máquina Víctima).

Configurar un firewall local: Un firewall local actúa como una barrera entre tu equipo y el resto de la red. Su función es controlar el tráfico entrante y saliente según reglas definidas: qué puertos, direcciones IP o protocolos están permitidos o bloqueados.

Al configurarlo correctamente, puedes bloquear intentos de conexión no solicitados, es decir, evitar que alguien desde fuera escanee o intente conectarse a tus servicios sin autorización. Solo el tráfico que tú declares explícitamente como permitido podrá pasar.


```
# instalar
sudo apt update
sudo apt install ufw -y

# política por defecto: bloquear entradas, permitir salidas
sudo ufw default deny incoming
sudo ufw default allow outgoing

# permitir SSH con rate limit (evita brute force)
sudo ufw limit ssh

# permitir sólo puertos necesarios
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp

# activar
sudo ufw enable

# ver reglas
sudo ufw status verbose
```

Cortar scans rápidos con reglas de tasa: Otra forma de defensa es aplicar reglas de limitación de tasa, que permiten conexiones normales pero bloquean o descartan aquellas que generan demasiadas solicitudes en poco tiempo.

Durante un escaneo rápido, el atacante envía una gran cantidad de paquetes SYN por segundo hacia distintos puertos. Este comportamiento no es típico del uso legítimo de un servicio, por lo que se puede filtrar fácilmente.

```
sudo apt install nftables -y
sudo systemctl enable --now nftables

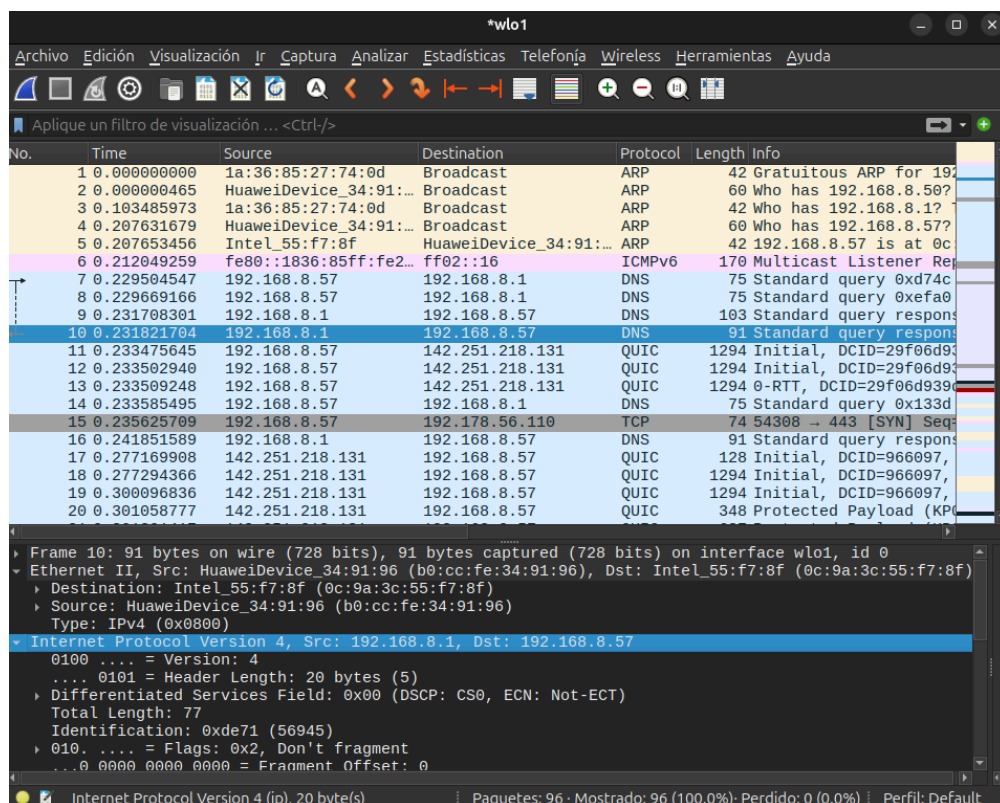
sudo nft add table inet filter
sudo nft 'add chain inet filter input { type filter hook input priority 0; }'
sudo nft 'add rule inet filter input ct state established,related accept'
sudo nft 'add rule inet filter input tcp flags syn ct count over 20/second drop'
```

Detección de escaneos en registros firewall: Además de bloquear conexiones sospechosas, un buen firewall también puede detectar patrones de escaneo al analizar sus propios registros (logs). Cuando un atacante realiza un escaneo puerto a puerto, el sistema recibe múltiples intentos de conexión seguidos a diferentes puertos desde una misma dirección IP. Estos eventos, registrados en los logs del firewall, pueden interpretarse como una señal de reconocimiento o ataque.

Mediante herramientas de monitoreo o scripts de análisis, se pueden establecer alertas automáticas que avisen al administrador si una IP genera demasiadas solicitudes en poco tiempo o intenta acceder a puertos no autorizados.

```
sudo apt install psad -y
sudo psad --sig-update
sudo psad --Status
```

WireShark: Esta herramienta captura y muestra cada pieza de información que viaja por la red. Cada fila que ves en la lista es un "paquete" de datos individual. Piensa en cada paquete como una pequeña carta digital.



Wireshark te permite ver "detrás de cámaras" cómo se comunican tus dispositivos con Internet. Las columnas te dicen:

- Source (Fuente): Qué dispositivo envió la "carta".
- Destination (Destino): Para quién era la "carta".

- Protocol (Protocolo): Qué "idioma" o conjunto de reglas se usó para escribir la carta.
- Info (Información): Un resumen de lo que dice la carta.

IP (Internet Protocol):

- Es el "sistema de correo" de internet. Su trabajo es ponerle una dirección (la dirección IP) a cada paquete de datos para que los routers sepan a dónde enviarlo. No garantiza que llegue, solo pone la dirección y lo envía.

TCP (Transmission Control Protocol):

- Es el "servicio de mensajería con acuse de recibo". Trabaja junto con IP y se asegura de que todos los datos lleguen, que lleguen en el orden correcto y sin errores. Si un paquete se pierde, TCP lo pide de nuevo.

UDP (User Datagram Protocol):

- Es el "servicio de mensajería rápido". También trabaja con IP, pero su prioridad es la velocidad, no la fiabilidad. Envía los paquetes y no le importa si algunos se pierden o llegan en desorden.

HTTP (Hypertext Transfer Protocol):

- Es el idioma que usa tu navegador para pedir y recibir páginas web. Cuando escribes una dirección "http://", estás usando este protocolo.

HTTPS (HTTP Secure):

- Es la versión segura de HTTP. Hace lo mismo (pedir páginas web), pero toda la conversación entre tu navegador y el servidor está cifrada. Es el "candado" que ves en la barra de direcciones y garantiza que nadie pueda espiar lo que envías o recibes.

DNS (Domain Name System):

- Es la "agenda telefónica" de internet. Los humanos usamos nombres fáciles de recordar, pero las computadoras usan números. DNS es el servicio que traduce esos nombres a sus IPs correspondientes.

FTP (File Transfer Protocol):

- Un protocolo diseñado específicamente para transferir archivos (subir y bajar) desde y hacia un servidor.

5. ¿Por qué usar Wireshark?

Una de las principales fortalezas de Wireshark es su capacidad para usar filtros de captura y de visualización, lo que permite enfocar el análisis en paquetes específicos dentro del tráfico de red. Gracias a esta función, es posible identificar patrones típicos de escaneo o comportamientos anómalos que podrían indicar un intento de reconocimiento o ataque.

Como medida de protección, estos filtros se convierten en una herramienta valiosa: permiten detectar escaneos en tiempo real, registrar evidencias y generar alertas tempranas.

Escaneo SYN (half-open): Muestra los paquetes TCP SYN , sin el flag ACK. Un escaneo tipo SYN, como el que usa nmap -sS, envía solo ese paquete SYN para ver si el puerto responde con un SYN-ACK (abierto) o un RST (cerrado). No completa la conexión, y así evita dejar registros evidentes en el sistema destino.

tcp.flags.syn == 1 and tcp.flags.ack == 0

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

tcp.flags.syn == 1 and tcp.flags.ack == 0

No.	Time	Source	Destination	Protocol	Length	Info
1082	70.260195151	192.168.8.57	212.102.40.163	TCP	74	60816 → 443 [SYN] Seq=0 Win=64240
1235	73.857334465	192.168.8.57	212.102.40.166	TCP	74	48904 → 443 [SYN] Seq=0 Win=64240
3730	271.058725878	192.168.8.57	185.125.190.17	TCP	74	39196 → 80 [SYN] Seq=0 Win=64240
4022	282.943976671	192.168.8.57	34.120.208.123	TCP	74	52986 → 443 [SYN] Seq=0 Win=64240
4028	283.007716632	192.168.8.57	34.36.137.203	TCP	74	58134 → 443 [SYN] Seq=0 Win=64240
4034	283.050303228	192.168.8.57	34.36.54.80	TCP	74	38564 → 443 [SYN] Seq=0 Win=64240
4058	283.194671938	192.168.8.57	34.120.208.123	TCP	74	52990 → 443 [SYN] Seq=0 Win=64240
4285	285.015844145	192.168.8.57	140.82.112.4	TCP	74	35442 → 443 [SYN] Seq=0 Win=64240
4298	285.092346146	192.168.8.57	192.178.52.174	TCP	74	37080 → 443 [SYN] Seq=0 Win=64240
4787	286.324402811	192.168.8.57	172.217.2.138	TCP	74	49110 → 443 [SYN] Seq=0 Win=64240
4788	286.324512379	192.168.8.57	172.217.2.138	TCP	74	49126 → 443 [SYN] Seq=0 Win=64240
5217	287.023172895	192.168.8.57	172.217.2.142	TCP	74	50994 → 443 [SYN] Seq=0 Win=64240
5942	295.849886914	192.168.8.57	192.178.56.46	TCP	74	57020 → 443 [SYN] Seq=0 Win=64240
6159	305.327236585	192.168.8.57	172.64.148.235	TCP	74	56108 → 443 [SYN] Seq=0 Win=64240
6175	305.469842640	192.168.8.57	172.64.148.235	TCP	74	56112 → 443 [SYN] Seq=0 Win=64240
6238	310.646997921	192.168.8.57	192.178.56.234	TCP	74	51244 → 443 [SYN] Seq=0 Win=64240
6586	317.599859839	192.168.8.57	34.107.243.93	TCP	74	49242 → 443 [SYN] Seq=0 Win=64240
6605	317.718716868	192.168.8.57	34.107.243.93	TCP	74	49248 → 443 [SYN] Seq=0 Win=64240
6624	317.992751603	192.168.8.57	151.101.1.91	TCP	74	46746 → 443 [SYN] Seq=0 Win=64240
7716	370.411333794	192.168.8.57	212.102.40.166	TCP	74	38134 → 443 [SYN] Seq=0 Win=64240

Frame 1082: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface wlo1, id 0
 Ethernet II, Src: Intel_55:f7:8f (0c:9a:3c:55:f7:8f), Dst: HuaweiDevice_34:91:96 (b0:cc:fe:34:91:96)
 Internet Protocol Version 4, Src: 192.168.8.57, Dst: 212.102.40.163
 Transmission Control Protocol, Src Port: 60816, Dst Port: 443, Seq: 0, Len: 0

- **Conexiones con RST masivas:** Si ves muchos RST consecutivos provenientes del mismo destino, suele significar que alguien está probando varios puertos a la vez, lo que es típico de un escaneo SYN o Connect.

tcp.flags.reset == 1

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

tcp.flags.reset == 1

No.	Time	Source	Destination	Protocol	Length	Info
56	2.770080606	192.168.8.57	52.108.248.0	TCP	54	53198 → 443 [RST] Seq=40 Win=0 Len=0
57	2.770107867	192.168.8.57	52.108.248.0	TCP	54	53198 → 443 [RST] Seq=40 Win=0 Len=0
58	2.775061782	52.108.248.0	192.168.8.57	TCP	60	443 → 53198 [RST, ACK] Seq=41 Ack=
68	3.783983407	192.168.8.57	52.111.230.2	TCP	54	42554 → 443 [RST] Seq=40 Win=0 Len=0
70	3.785008141	192.168.8.57	52.111.230.2	TCP	54	42554 → 443 [RST] Seq=40 Win=0 Len=0
71	3.787363960	52.111.230.2	192.168.8.57	TCP	60	443 → 42554 [RST, ACK] Seq=41 Ack=
1027	68.817698478	212.102.40.163	192.168.8.57	TCP	60	443 → 37806 [RST] Seq=2 Win=0 Len=0
3988	282.321814111	192.178.56.46	192.168.8.57	TCP	60	443 → 37210 [RST] Seq=4555 Win=0 Len=0
3989	282.322844749	192.178.56.46	192.168.8.57	TCP	60	443 → 37210 [RST] Seq=4555 Win=0 Len=0
3990	282.323852175	192.178.56.46	192.168.8.57	TCP	60	443 → 37210 [RST] Seq=4555 Win=0 Len=0
3991	282.331675346	192.178.56.46	192.168.8.57	TCP	60	443 → 37210 [RST] Seq=4555 Win=0 Len=0
5845	290.626314680	192.168.8.57	140.82.112.4	TCP	54	35442 → 443 [RST] Seq=2147 Win=0 Len=0
5847	290.627360427	192.168.8.57	140.82.112.4	TCP	54	35442 → 443 [RST] Seq=2147 Win=0 Len=0
6130	302.824654838	172.64.148.235	192.168.8.57	TCP	60	443 → 43740 [RST] Seq=531 Win=0 Len=0
6536	312.523986821	34.107.243.93	192.168.8.57	TCP	60	443 → 36788 [RST] Seq=1 Win=0 Len=0
7702	368.853919276	212.102.40.166	192.168.8.57	TCP	60	443 → 48904 [RST] Seq=1669 Win=0 Len=0
8523	488.805437695	192.168.8.57	151.101.1.91	TCP	54	46746 → 443 [RST] Seq=1164 Win=0 Len=0
8525	488.805674096	192.168.8.57	151.101.1.91	TCP	54	46746 → 443 [RST] Seq=1164 Win=0 Len=0
9756	613.178527519	172.217.2.138	192.168.8.57	TCP	60	443 → 49110 [RST] Seq=8357 Win=0 Len=0
10054	639.125602949	192.178.56.46	192.168.8.57	TCP	60	443 → 57020 [RST] Seq=13042 Win=0 Len=0

Frame 1027: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface wlo1, id 0
 Ethernet II, Src: HuaweiDevice_34:91:96 (b0:cc:fe:34:91:96), Dst: Intel_55:f7:8f (0c:9a:3c:55:f7:8f)
 Internet Protocol Version 4, Src: 212.102.40.163, Dst: 192.168.8.57
 Transmission Control Protocol, Src Port: 443, Dst Port: 37806, Seq: 2, Len: 0

- **Escaneo UDP:** Muestra todas las respuestas ICMP que indican puertos UDP cerrados, revelando que alguien podría estar probando puertos UDP en tu máquina.

icmp.type == 3 and icmp.code == 3

- **Ping sweep / hosts que responden a ICMP:** Este filtro te deja ver quién está enviando o recibiendo pings ICMP, algo típico en un *ping sweep*, que es una técnica usada para descubrir qué equipos están activos dentro de una red.

icmp.type == 0 or icmp.type == 8

- **Tráfico a muchos puertos desde una misma IP:** Este filtro mostrará todo el tráfico TCP proveniente de una IP concreta. Si ves que esa IP envía paquetes a muchos puertos distintos de forma continua, eso suele indicar un escaneo de puertos TCP.

ip.src == 10.0.0.5 and tc

Conclusiones:

Al realizar tanto los escaneos con Nmap como la monitorización con Wireshark, comprendí de forma práctica cómo interactúan el ataque y la defensa dentro de una red. Nmap permitió descubrir los servicios expuestos y entender la superficie de ataque, mientras que Wireshark evidenció las trazas que esos escaneos dejan en el tráfico, revelando los patrones que pueden alertar sobre una actividad sospechosa.

Esta experiencia conjunta me ayudó a conectar la teoría con la práctica: interpretar paquetes, detectar anomalías y analizar los resultados desde ambos lados del proceso. En conjunto, ambas herramientas demostraron su valor para una auditoría de seguridad completa, que no solo busca vulnerabilidades, sino que también enseña a reconocerlas y mitigarlas mediante observación, análisis y registro técnico riguroso.

Referencias bibliográficas

[1] *Nmap: The network mapper - Free Security Scanner*. (n.d.). Nmap.org. Retrieved November 12, 2025, from <https://nmap.org/>

[2] *Wireshark • go deep*. (n.d.). Wireshark. Retrieved November 12, 2025, from <https://www.wireshark.org/>

[3] *Oracle VirtualBox*. (n.d.). Virtualbox.org. Retrieved November 12, 2025, from <https://www.virtualbox.org/>

[4] (N.d.-b). Akamai.com. Retrieved November 12, 2025, from <https://www.akamai.com/es/glossary/what-are-syn-flood-ddos-attacks>

[5] (N.d.). Ninjaone.com. Retrieved November 12, 2025, from <https://www.ninjaone.com/es/it-hub/endpoint-management/handshake/>

E. (n.d.). *NMAP 6: Listado de comandos*. Gva.Es. Retrieved November 12, 2025, from <https://csirtcv.gva.es/wp-content/uploads/2020/05/NMAP-6 -Listado-de-comandos.pdf>

(N.d.). Ninjaone.com. Retrieved November 12, 2025, from <https://www.ninjaone.com/es/blog/utilizar-nmap-guia-completa/>

Conceptos Básicos de Redes. (n.d.). FIRST Robotics Competition Documentation. Retrieved November 12, 2025, from <https://docs.wpilib.org/es/latest/docs/networking/networking-introduction/networking-basics.html>

Support. (2007, May 4). Cisco. <https://www.cisco.com/en/US/docs/security/vpn5000/manager/reference/guide/appA.html>

(N.d.-b). Labex.Io. Retrieved November 12, 2025, from <https://labex.io/es/tutorials/nmap-how-to-ensure-the-security-of-nmap-scan-data-4155>
18

Wireshark-filter(4). (n.d.). Wireshark.org. Retrieved November 12, 2025, from <https://www.wireshark.org/docs/man-pages/wireshark-filter.html>

Luz, S. (2020, July 4). *Cómo usar Wireshark para capturar y analizar el tráfico de red*. RedesZone.
<https://www.redeszone.net/tutoriales/redes-cable/wireshark-capturar-analizar-trafico-red/>