REDES II

ACTIVIDAD

Monitoreo y escaneo de vulnerabilidades con las herramientas nmap y wireshark

JAIRO IVAN HIPOLITO MORALES

17 de Noviembre del 2025

Índice

1. Creación del Sandbox (Entorno Controlado)

A continuación, se detallan los pasos realizados para la puesta en marcha del laboratorio.

1.1. Instalación de Virtual Box

Se utilizó Oracle VirtualBox como plataforma de virtualización debido a su capacidad para gestionar redes virtuales y su compatibilidad con las imágenes ISO seleccionadas. Se procedió a la instalación y configuración inicial del software en el sistema operativo anfitrión.

1.2. Despliegue de Máquinas Virtuales (VM)

Se crearon dos instancias virtuales independientes:

1. Atacante (Kali Linux): Se configuró una VM basada en Debian utilizando la imagen ISO oficial de Kali Linux.
2. Víctima (Ubuntu): Se configuró una VM con Ubuntu Desktop. Esta máquina actuará como el objetivo del escaneo y ejecutará las herramientas de monitoreo.

1.3. Verificación de Conectividad

Una vez iniciados ambos sistemas operativos, se procedió a validar la conexión y obtener las direcciones IP asignadas.

- Identificación de IPs:
    - IP Víctima (Ubuntu): 192.168.0.19
    - IP Atacante (Kali): 192.168.0.20
- Prueba de Ping: Se realizó una prueba de conectividad ICMP (Ping) desde la máquina atacante hacia la víctima para confirmar que el canal de comunicación está activo y listo para las pruebas de escaneo.

2. Introducción

En esta práctica se realiza un ejercicio de monitoreo y escaneo de vulnerabilidades dentro de un entorno seguro, utilizando dos herramientas fundamentales en el área de redes y ciberseguridad: Nmap y Wireshark. El propósito es comprender cómo se lleva a cabo un análisis básico de una red desde la perspectiva ofensiva (ataque) y defensiva (monitoreo), aplicando técnicas de reconocimiento, identificación de servicios y análisis de tráfico.

Para ello se utiliza un entorno controlado tipo sandbox, compuesto por dos máquinas virtuales aisladas entre sí y de la red real. Una máquina funciona como atacante, ejecutando escaneos con Nmap, mientras que la otra actúa como víctima, capturando y analizando el tráfico con Wireshark. Este entorno permite experimentar de manera segura sin comprometer otros equipos o servicios externos.

2.1. Conceptos

2.1.1. Sandbox

Es un entorno seguro y aislado donde se pueden ejecutar pruebas sin afectar la red real. En esta práctica se crea un sandbox con dos máquinas virtuales interconectadas únicamente entre ellas.

2.1.2. Escaneo de puertos

Técnica utilizada para identificar qué puertos están abiertos en un equipo y qué servicios se están ejecutando (SSH, HTTP, FTP, etc.).

2.1.3. Escaneo de Vulnerabilidades (Vulnerability Scanning)

El escaneo de vulnerabilidades es un proceso automatizado que identifica, clasifica y prioriza las debilidades de seguridad (vulnerabilidades) en sistemas informáticos, aplicaciones y la infraestructura de red. Su objetivo es proporcionar una visión clara de los riesgos de seguridad antes de que puedan ser explotados por atacantes.

2.1.4. Monitoreo de Red (Network Monitoring)

El monitoreo de red es la práctica de observar y analizar el tráfico de datos en una red en tiempo real. Permite a los administradores detectar y responder a incidentes de seguridad, identificar cuellos de botella en el rendimiento y rastrear actividades anómalas o sospechosas que puedan indicar un ataque o una brecha de seguridad.

2.1.5. Vulnerabilidad

Un fallo o debilidad en el diseño, implementación o configuración de un sistema que podría ser explotado por un atacante para comprometer la seguridad.

2.2. Detalle de Herramientas

2.2.1. Nmap

Nmap es una utilidad de código abierto para la exploración de red y auditorías de seguridad. Está diseñado para escanear redes grandes de forma rápida, aunque funciona bien contra hosts individuales. Permite determinar qué hosts están disponibles en la red, qué servicios (nombre de aplicación y versión) ofrecen, qué sistemas operativos (y versiones) están ejecutando, qué tipo de filtros de paquetes/cortafuegos están en uso, y docenas de otras características.

2.2.2. Wireshark

Wireshark es un analizador de tráfico que permite capturar, visualizar y examinar los paquetes que circulan por una interfaz de red en tiempo real. Ofrece la posibilidad de inspeccionar protocolos, direcciones IP, puertos, flags y contenidos específicos dentro de cada paquete, lo que facilita la identificación de comportamientos anómalos, actividades sospechosas o patrones generados por ataques como escaneos de puertos. Su interfaz gráfica y su capacidad de aplicar filtros lo convierten en una herramienta indispensable para monitorear, analizar y comprender el funcionamiento interno de una red.

2.3. Requisitos del Entorno

Para la correcta ejecución del monitoreo y escaneo de vulnerabilidades, se ha establecido la siguiente infraestructura de laboratorio virtualizado:

2.3.1. Equipo Anfitrión (Host)

- Computadora física con recursos suficientes para ejecutar dos máquinas virtuales simultáneamente.
- Software de virtualización instalado (en este caso, VirtualBox).

2.3.2. Máquina Virtual Atacante (Kali Linux)

- Sistema Operativo: Kali Linux (versión actual).
- Rol: Nodo ofensivo encargado de realizar los escaneos.

2.3.3. Máquina Virtual Víctima (Ubuntu)

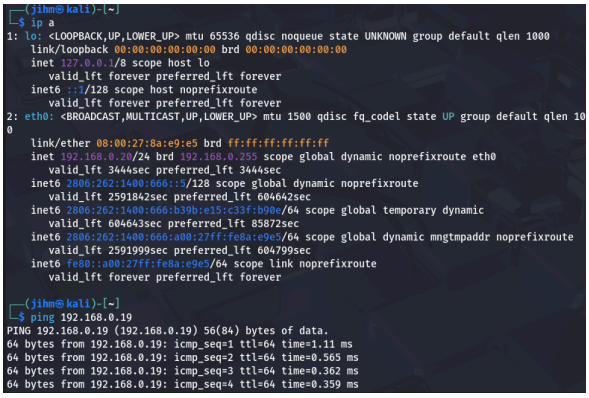
- Sistema Operativo: Ubuntu (Desktop o Server).
- Rol: Nodo objetivo y de monitoreo defensivo.

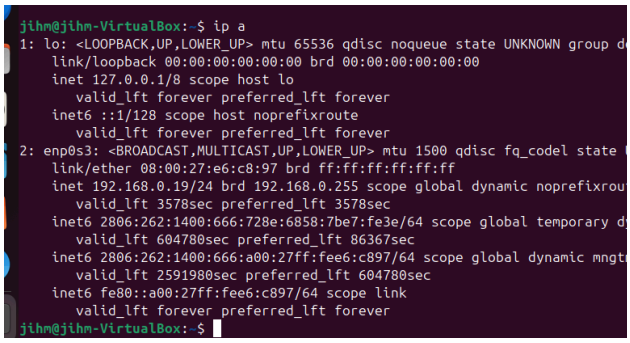
3. Desarrollo: Documentación y Descripción de la Práctica

3.1 Preparar el sandbox

Ambas máquinas virtuales fueron configuradas con un adaptador de red en modo “Red Interna”, lo que garantiza que solo puedan comunicarse entre sí y queden completamente aisladas del exterior.
Una vez encendidas las máquinas, se verificó su conectividad mediante comandos básicos de red, principalmente ip a para identificar la dirección IP asignada a cada máquina y ping para comprobar la comunicación entre atacante y víctima. Con esta configuración se confirmó que el sandbox estaba correctamente establecido y listo para ejecutar los escaneos y pruebas de monitoreo correspondientes.

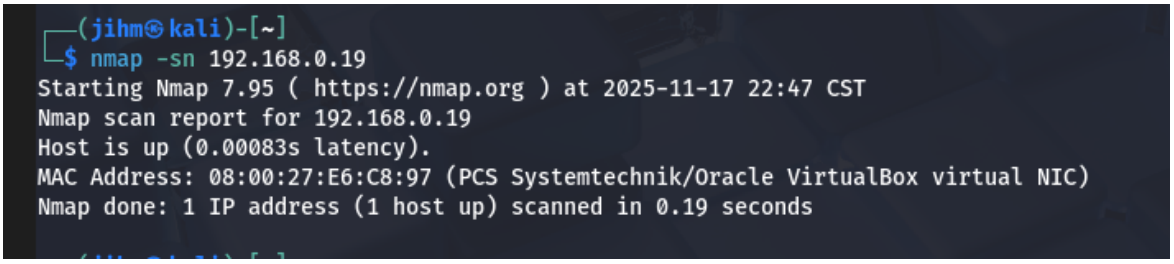
Primeramente, en ambas máquinas verificar la IP en la terminal con **ip a**, y en Kali Linux verificar si existe la conexión con la víctima con **ping <IP>**.





3.2. Práctica de Ataque: Escaneo con Nmap

En la terminal del atacante ejecutamos lo siguiente:
3.2.1 Primer escaneo: Escaneo de descubrimiento para verificar que la víctima está activa.
Comando: nmap -sn 192.168.0.19.



3.2.2 Escaneo rápido de puertos comunes: Nos ayuda a saber que puertos están abiertos sin tardar tanto. Comando: nmap -F 192.168.0.19.

```
┌──(jihm㉿kali)-[~]
└─$ nmap -F 192.168.0.19
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-17 22:49 CST
Nmap scan report for 192.168.0.19
Host is up (0.0015s latency).
All 100 scanned ports on 192.168.0.19 are in ignored states.
Not shown: 100 closed tcp ports (reset)
MAC Address: 08:00:27:E6:C8:97 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.48 seconds
```

Resultado del escaneo rápido con nmap -F, donde se observa que la máquina víctima no tiene puertos comunes abiertos.

3.2.3 Escaneo completo y agresivo: Nos ayuda a comprobar la capacidad de Nmap para identificar servicios y sistema operativo. Comando: nmap -A 192.168.0.19.

```
└─$ nmap -A 192.168.0.19
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-17 22:53 CST
Nmap scan report for 192.168.0.19
Host is up (0.0013s latency).
All 1000 scanned ports on 192.168.0.19 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 08:00:27:E6:C8:97 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   1.28 ms 192.168.0.19

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 2.26 seconds
```

3.2.4 Escaneo sigiloso (SYN scan): Es un escaneo “más seguro” para el atacante, que es más difícil de detectar. Comando: nmap -sS 192.168.0.19

```
┌──(jihm㉿kali)-[~]
└─$ nmap -sS 192.168.0.19
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-17 22:56 CST
Nmap scan report for 192.168.0.19
Host is up (0.00019s latency).
All 1000 scanned ports on 192.168.0.19 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 08:00:27:E6:C8:97 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
```

Escaneo sigiloso tipo SYN Scan (nmap -sS), donde se observa que la máquina víctima no presenta puertos abiertos.

3.3. Monitoreo con Wireshark (defensa)

3.3.1 Instalar Wireshark

Primeramente, instalamos Wireshark desde la terminal de nuestra víctima (Ubuntu) con los siguientes comandos:

sudo apt update

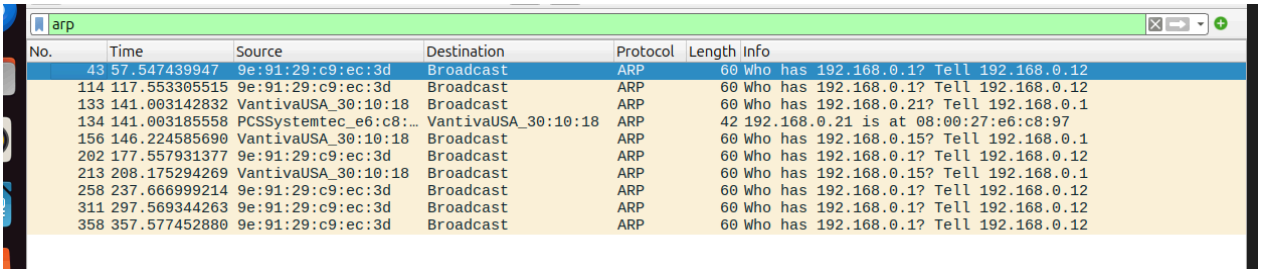
sudo apt install wireshark

Luego lo abrimos y seleccionamos la interfaz de red de Ubuntu. Y después seleccionamos el boton de Start Capturing, que significa que ya se encuentra monitoreando la red.

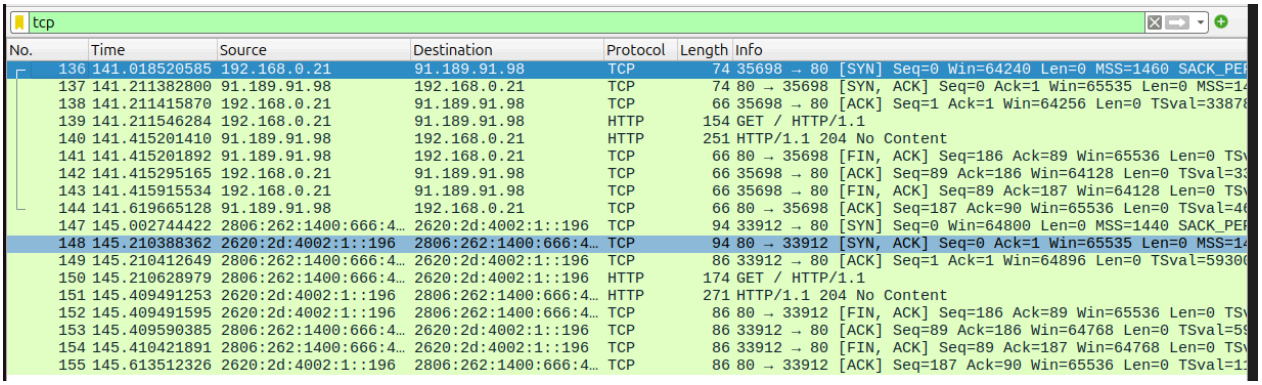
3.3.2 Ejecutar escaneos en Kali Linux y observar como aparece tráfico en Wireshark.

3.3.3 Aplicar filtros para detectar actividad sospechosa

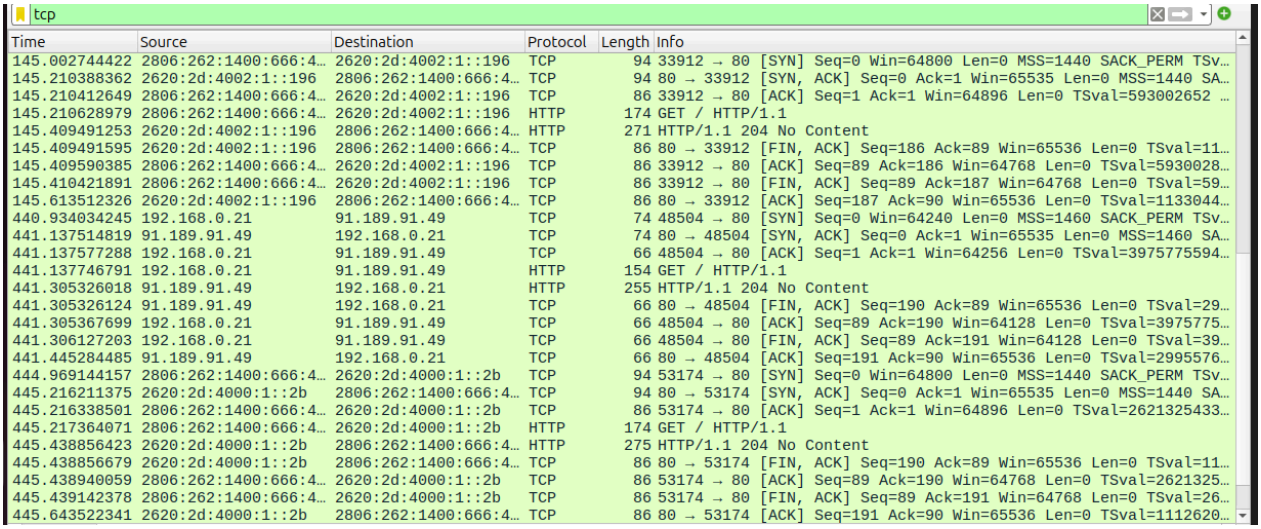
- Captura del Escaneo de Descubrimiento (nmap -sn)
  Filtro en wireshark: arp



- Captura del Escaneo Rápido de Puertos (nmap -F)
  Filtro en wireshark: tcp

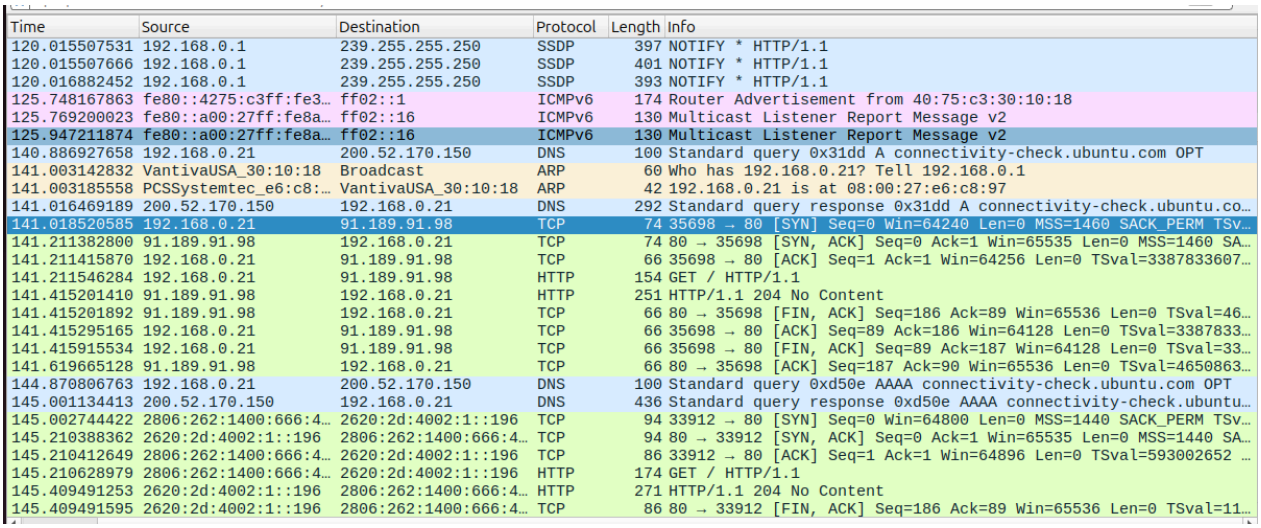


- Captura del Escaneo Agresivo (nmap -A)
  Filtro en wireshark: tcp



Time	Source	Destination	Protocol	Length	Info
145.002744422	2806:262:1400:666:4…	2620:2d:4002:1::196	TCP	94	33912 → 80 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 SACK_PERM TSv…
145.210388362	2620:2d:4002:1::196	2806:262:1400:666:4…	TCP	94	80 → 33912 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SA…
145.210412649	2806:262:1400:666:4…	2620:2d:4002:1::196	TCP	86	33912 → 80 [ACK] Seq=1 Ack=1 Win=64896 Len=0 TSval=593002652 …
145.210628979	2806:262:1400:666:4…	2620:2d:4002:1::196	HTTP	174	GET / HTTP/1.1
145.409491253	2620:2d:4002:1::196	2806:262:1400:666:4…	HTTP	271	HTTP/1.1 204 No Content
145.409491595	2620:2d:4002:1::196	2806:262:1400:666:4…	TCP	86	80 → 33912 [FIN, ACK] Seq=186 Ack=89 Win=65536 Len=0 TSval=11…
145.409590385	2806:262:1400:666:4…	2620:2d:4002:1::196	TCP	86	33912 → 80 [ACK] Seq=89 Ack=186 Win=64768 Len=0 TSval=5930028…
145.410421891	2806:262:1400:666:4…	2620:2d:4002:1::196	TCP	86	33912 → 80 [FIN, ACK] Seq=89 Ack=187 Win=64768 Len=0 TSval=59…
145.613512326	2620:2d:4002:1::196	2806:262:1400:666:4…	TCP	86	80 → 33912 [ACK] Seq=187 Ack=90 Win=65536 Len=0 TSval=1133044…
440.934034245	192.168.0.21	91.189.91.49	TCP	74	48504 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSv…
441.137514819	91.189.91.49	192.168.0.21	TCP	74	80 → 48504 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SA…
441.137577288	192.168.0.21	91.189.91.49	TCP	66	48504 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3975775594…
441.137746791	192.168.0.21	91.189.91.49	HTTP	154	GET / HTTP/1.1
441.305326018	91.189.91.49	192.168.0.21	HTTP	255	HTTP/1.1 204 No Content
441.305326124	91.189.91.49	192.168.0.21	TCP	66	80 → 48504 [FIN, ACK] Seq=190 Ack=89 Win=65536 Len=0 TSval=29…
441.305367699	192.168.0.21	91.189.91.49	TCP	66	48504 → 80 [ACK] Seq=89 Ack=190 Win=64128 Len=0 TSval=3975775…
441.306127203	192.168.0.21	91.189.91.49	TCP	66	48504 → 80 [FIN, ACK] Seq=89 Ack=191 Win=64128 Len=0 TSval=39…
441.445284485	91.189.91.49	192.168.0.21	TCP	66	80 → 48504 [ACK] Seq=191 Ack=90 Win=65536 Len=0 TSval=2995576…
444.969144157	2806:262:1400:666:4…	2620:2d:4000:1::2b	TCP	94	53174 → 80 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 SACK_PERM TSv…
445.216211375	2620:2d:4000:1::2b	2806:262:1400:666:4…	TCP	94	80 → 53174 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SA…
445.216338501	2806:262:1400:666:4…	2620:2d:4000:1::2b	TCP	86	53174 → 80 [ACK] Seq=1 Ack=1 Win=64896 Len=0 TSval=2621325433…
445.217364071	2806:262:1400:666:4…	2620:2d:4000:1::2b	HTTP	174	GET / HTTP/1.1
445.438856423	2620:2d:4000:1::2b	2806:262:1400:666:4…	HTTP	275	HTTP/1.1 204 No Content
445.438856679	2620:2d:4000:1::2b	2806:262:1400:666:4…	TCP	86	80 → 53174 [FIN, ACK] Seq=190 Ack=89 Win=65536 Len=0 TSval=11…
445.438940059	2806:262:1400:666:4…	2620:2d:4000:1::2b	TCP	86	53174 → 80 [ACK] Seq=89 Ack=190 Win=64768 Len=0 TSval=2621325…
445.439142378	2806:262:1400:666:4…	2620:2d:4000:1::2b	TCP	86	53174 → 80 [FIN, ACK] Seq=89 Ack=191 Win=64768 Len=0 TSval=26…
445.643522341	2620:2d:4000:1::2b	2806:262:1400:666:4…	TCP	86	80 → 53174 [ACK] Seq=191 Ack=90 Win=65536 Len=0 TSval=1112620…

- Captura del Escaneo Sigiloso (nmap -sS)



Time	Source	Destination	Protocol	Length	Info
120.015507531	192.168.0.1	239.255.255.250	SSDP	397	NOTIFY * HTTP/1.1
120.015507666	192.168.0.1	239.255.255.250	SSDP	401	NOTIFY * HTTP/1.1
120.016882452	192.168.0.1	239.255.255.250	SSDP	393	NOTIFY * HTTP/1.1
125.748167863	fe80::4275:c3ff:fe3…	ff02::1	ICMPv6	174	Router Advertisement from 40:75:c3:30:10:18
125.769200023	fe80::a00:27ff:fe8a…	ff02::16	ICMPv6	130	Multicast Listener Report Message v2
125.947211874	fe80::a00:27ff:fe8a…	ff02::16	ICMPv6	130	Multicast Listener Report Message v2
140.886927658	192.168.0.21	200.52.170.150	DNS	100	Standard query 0x31dd A connectivity-check.ubuntu.com OPT
141.003142832	VantivaUSA_30:10:18	Broadcast	ARP	60	Who has 192.168.0.21? Tell 192.168.0.1
141.003185558	PCSSystemtec_e6:c8:…	VantivaUSA_30:10:18	ARP	42	192.168.0.21 is at 08:00:27:e6:c8:97
141.016469189	200.52.170.150	192.168.0.21	DNS	292	Standard query response 0x31dd A connectivity-check.ubuntu.co…
141.018520585	192.168.0.21	91.189.91.98	TCP	74	35698 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSv…
141.211382800	91.189.91.98	192.168.0.21	TCP	74	80 → 35698 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SA…
141.211415870	192.168.0.21	91.189.91.98	TCP	66	35698 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3387833607…
141.211546284	192.168.0.21	91.189.91.98	HTTP	154	GET / HTTP/1.1
141.415201410	91.189.91.98	192.168.0.21	HTTP	251	HTTP/1.1 204 No Content
141.415201892	91.189.91.98	192.168.0.21	TCP	66	80 → 35698 [FIN, ACK] Seq=186 Ack=89 Win=65536 Len=0 TSval=46…
141.415295165	192.168.0.21	91.189.91.98	TCP	66	35698 → 80 [ACK] Seq=89 Ack=186 Win=64128 Len=0 TSval=3387833…
141.415915534	192.168.0.21	91.189.91.98	TCP	66	35698 → 80 [FIN, ACK] Seq=89 Ack=187 Win=64128 Len=0 TSval=33…
141.619665128	91.189.91.98	192.168.0.21	TCP	66	80 → 35698 [ACK] Seq=187 Ack=90 Win=65536 Len=0 TSval=4650863…
144.870806763	192.168.0.21	200.52.170.150	DNS	100	Standard query 0xd50e AAAA connectivity-check.ubuntu.com OPT
145.001134413	200.52.170.150	192.168.0.21	DNS	436	Standard query response 0xd50e AAAA connectivity-check.ubuntu…
145.002744422	2806:262:1400:666:4…	2620:2d:4002:1::196	TCP	94	33912 → 80 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 SACK_PERM TSv…
145.210388362	2620:2d:4002:1::196	2806:262:1400:666:4…	TCP	94	80 → 33912 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1440 SA…
145.210412649	2806:262:1400:666:4…	2620:2d:4002:1::196	TCP	86	33912 → 80 [ACK] Seq=1 Ack=1 Win=64896 Len=0 TSval=593002652 …
145.210628979	2806:262:1400:666:4…	2620:2d:4002:1::196	HTTP	174	GET / HTTP/1.1
145.409491253	2620:2d:4002:1::196	2806:262:1400:666:4…	HTTP	271	HTTP/1.1 204 No Content
145.409491595	2620:2d:4002:1::196	2806:262:1400:666:4…	TCP	86	80 → 33912 [FIN, ACK] Seq=186 Ack=89 Win=65536 Len=0 TSval=11…

4. Conclusiones: Resultados Obtenidos y Recomendaciones

4.1 Resultados Obtenidos

En esta práctica se realizaron diferentes tipos de escaneos con Nmap dentro de un entorno seguro de máquinas virtuales, lo que permitió identificar cómo se comporta una red ante técnicas de reconocimiento. Los resultados mostraron que la máquina víctima no tenía puertos abiertos, por lo que todos los escaneos devolvieron puertos cerrados; sin embargo, esto permitió comparar la velocidad, el alcance y el nivel de sigilo entre escaneos básicos, rápidos, agresivos y sigilosos. De esta manera se comprobó que Nmap es una herramienta eficaz para obtener información sobre un host, incluso cuando la superficie de ataque es mínima.

El monitoreo con Wireshark permitió observar el tráfico generado por los escaneos y entender cómo un administrador de red puede detectar actividad sospechosa mediante patrones como múltiples paquetes SYN o solicitudes repetidas a distintos puertos. Esto demuestra que, aunque algunos escaneos buscan ser discretos, siguen generando rastros visibles en un sistema de monitoreo adecuado.

4.2 Recomendaciones

- Realizar siempre pruebas en un entorno aislado para evitar afectar las redes.
- Usar Nmap de forma complementaria con otras herramientas si se requiere un análisis más profundo.
- Aplicar filtros en Wireshark para facilitar la detección de patrones sospechosos.
- Habilitar algún servicio en la máquina víctima si se desean resultados más detallados.

En general, la práctica permitió comprender la eficacia de Nmap para el reconocimiento y la utilidad de Wireshark para detectar comportamientos anómalos en la red.