

Informe de Testeo de Seguridad en Redes de Datos

Nombre: Ricardo Cáceres Vera

Asignatura: Seguridad en Redes De Datos

Docente: José Morales Antunez

Fecha: 28 de agosto 2025

Introducción

El presente informe tiene como objetivo evaluar la seguridad de una red local mediante la ejecución de pruebas controladas de testeo. Para ello, se configuró un entorno compuesto por Kali Linux (192.168.1.35) como equipo de ataque, Metasploitable (192.168.1.34) como máquina vulnerable y Windows (192.168.1.33) como estación de apoyo.

Durante el análisis se aplicaron herramientas y técnicas de seguridad orientadas a simular ataques comunes y observar el comportamiento de los servicios de red. Entre las actividades realizadas destacan: el uso de hping3 para pruebas de conectividad y puertos, la captura y análisis de tráfico con Wireshark aplicando filtros específicos, y la validación de servicios y protocolos utilizados en la comunicación.

El objetivo principal es identificar debilidades y riesgos en la red, contrastar la diferencia entre protocolos cifrados y no cifrados, y finalmente proponer medidas de mejora que permitan fortalecer la seguridad y garantizar la disponibilidad, integridad y confidencialidad de la información.

1. hping3 (Pruebas de envío)

ICMP (ping) – Se observa el envío y la respuesta Echo Reply, confirmando que el host destino está disponible.

The screenshot shows a Kali Linux terminal window and a NetworkMiner tool window side-by-side.

Kali Linux Terminal:

```
root@kali:~# ip -br addr show
ping -c 4 192.168.1.34
ping: interface eth0 link layer address 00:0c:29:1d:01:01 brd 192.168.1.255
  lo      UNKNOWN    127.0.0.1/8  ::1/128
eth0    inet 192.168.1.34 brd 192.168.1.255 mask 255.255.255.0
        mac 00:0c:29:1d:01:01
ping: interface eth0 link layer address 00:0c:29:1d:01:01 brd 192.168.1.255
PING 192.168.1.34 (192.168.1.34) 56(84) bytes of data.
64 bytes from 192.168.1.34: icmp_seq=1 ttl=64 time=1.09 ms
64 bytes from 192.168.1.34: icmp_seq=2 ttl=64 time=1.09 ms
64 bytes from 192.168.1.34: icmp_seq=3 ttl=64 time=1.082 ms
64 bytes from 192.168.1.34: icmp_seq=4 ttl=64 time=1.94 ms
— 192.168.1.34 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 301ms
rtt min/avg/max/mdev = 0.802/1.493/2.144/0.560 ms

root@kali:~#
```

NetworkMiner Tool:

The NetworkMiner tool is capturing traffic on interface ***eth0**. The captured traffic table shows the following details:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.0000000000	192.168.1.35	192.168.1.34	ICMP	98	Echo (ping) request Id=0x0002 seq=1/256 ttl=64 (req)
2	0.0001076769	192.168.1.34	192.168.1.35	ICMP	98	Echo (ping) reply Id=0x0002 seq=1/256 ttl=64 (rep)
3	1.0126590646	192.168.1.35	192.168.1.34	ICMP	98	Echo (ping) request Id=0x0002 seq=2/256 ttl=64 (req)
4	1.0127590646	192.168.1.34	192.168.1.35	ICMP	98	Echo (ping) reply Id=0x0002 seq=2/256 ttl=64 (rep)
5	2.0026988413	192.168.1.35	192.168.1.34	ICMP	98	Echo (ping) request Id=0x0002 seq=3/256 ttl=64 (req)
6	2.0036749721	192.168.1.34	192.168.1.35	ICMP	98	Echo (ping) reply Id=0x0002 seq=3/256 ttl=64 (rep)
7	3.0136452554	192.168.1.35	192.168.1.34	ICMP	98	Echo (ping) request Id=0x0002 seq=4/256 ttl=64 (req)
8	3.0146122776	192.168.1.34	192.168.1.35	ICMP	98	Echo (ping) reply Id=0x0002 seq=4/256 ttl=64 (rep)

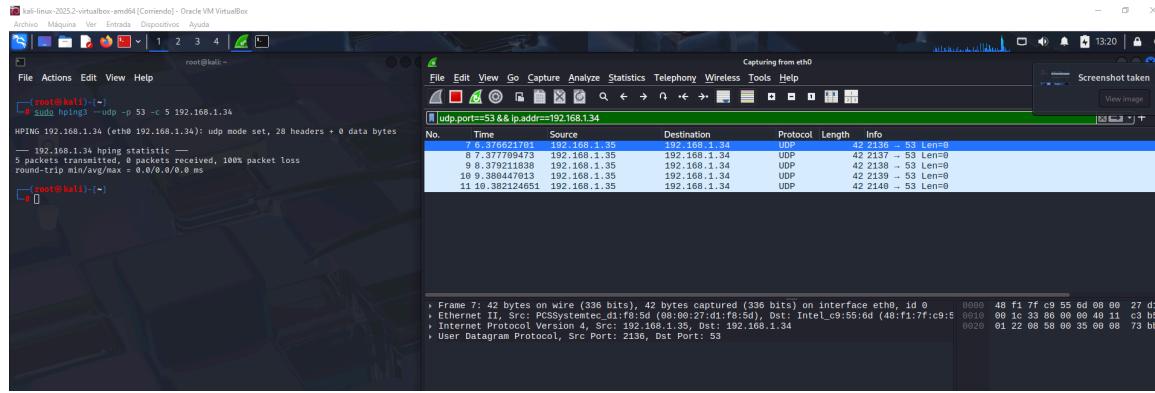
Details for the first frame:

```
> Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0, id 0 0000 48 f1 7f c9 55 6d 00 00 27 d1
> Ethernet II, Src: PCSysteme_d1:f8:5d (08:00:27:d1:f8:5d), Dst: Intel_c9:55:6d (48:01:fc:c9:55:6d)
> Internet Protocol Version 4, Src: 192.168.1.35, Dst: 192.168.1.34
> Internet Control Message Protocol
```

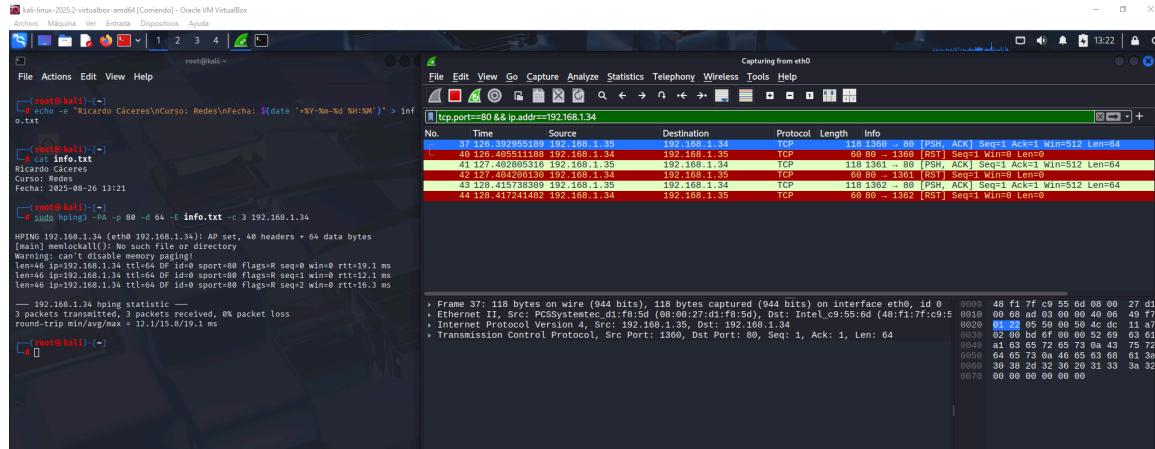
NetworkMiner also displays a timeline and a detailed view of the selected frame.

TCP SYN → 80/TCP – La captura muestra el intercambio SYN y la respuesta SYN/ACK, lo que confirma que el puerto 80 está abierto y en escucha.

UDP → 53/UDP – El tráfico enviado no recibe respuesta. Esto indica que el puerto está cerrado o filtrado.

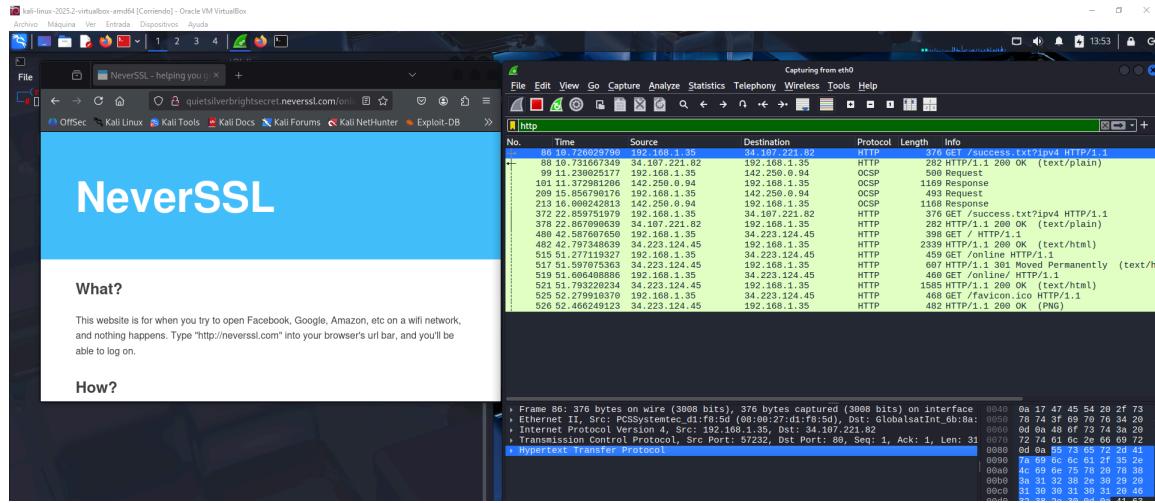


TCP con payload – El host respondió con RST, lo que indica que no había una sesión establecida para procesar los datos enviados.

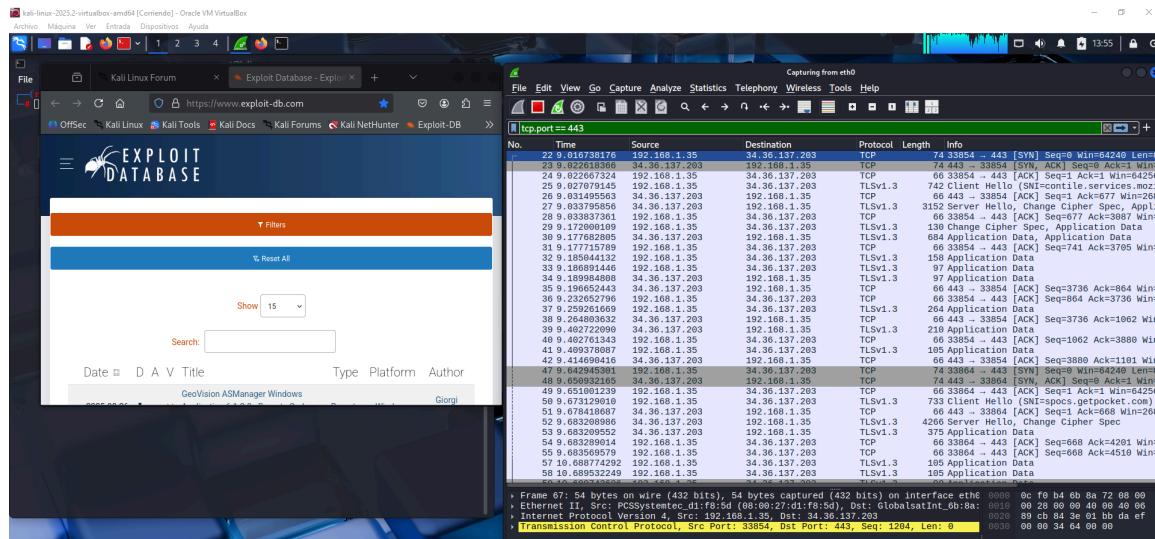


2. Wireshark (Navegación y protocolos)

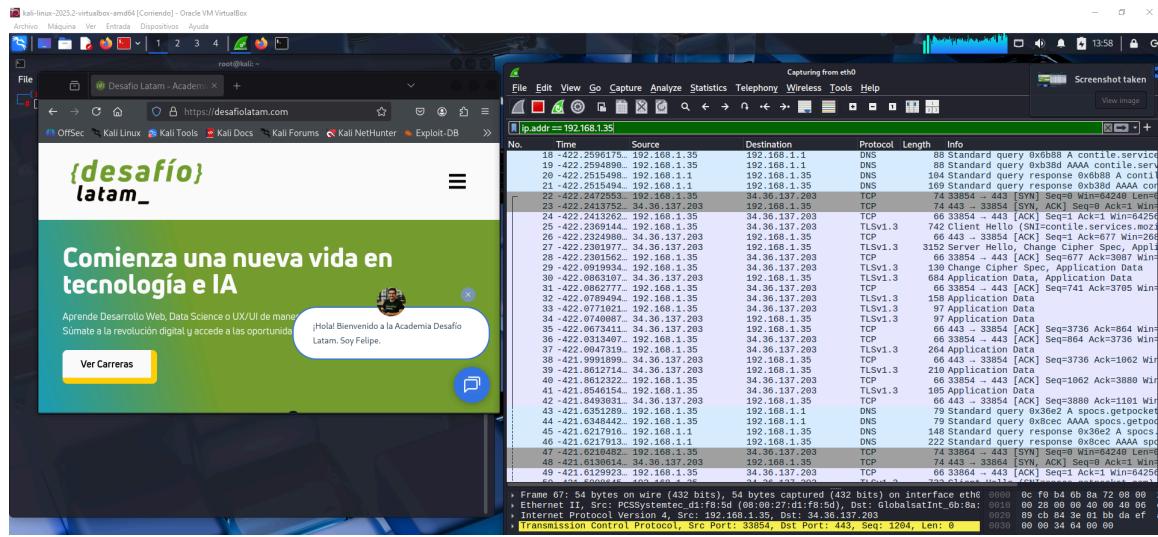
HTTP (NeverSSL) – En la captura se aprecia la petición GET / HTTP/1.1 en texto plano, evidenciando que el protocolo HTTP no ofrece cifrado.



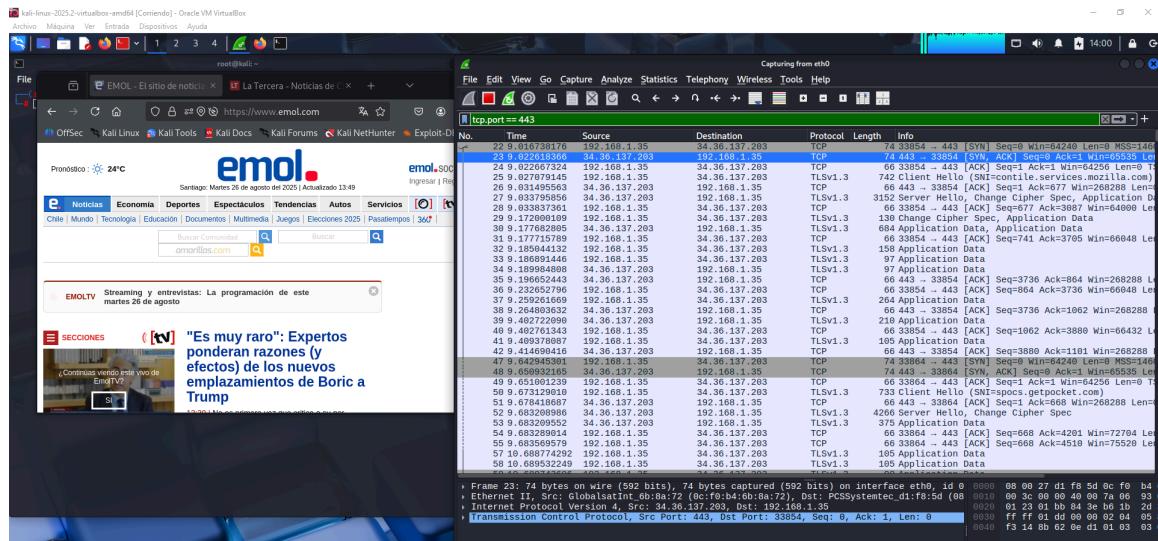
HTTPS (Exploit-DB) – Se observa el paquete Client Hello, que corresponde al inicio del handshake TLS. A partir de aquí la comunicación es cifrada.



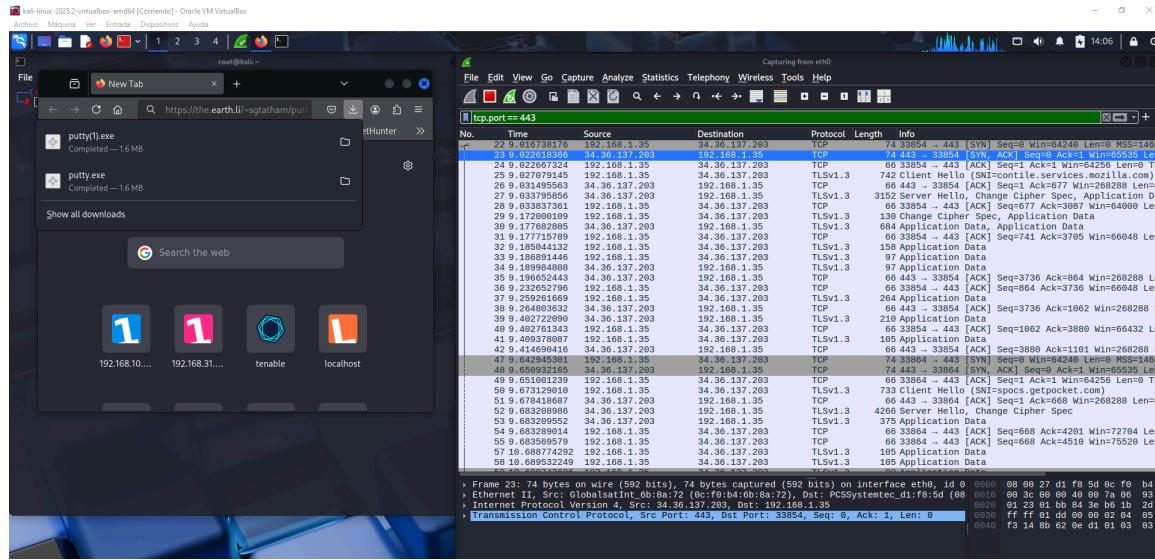
HTTPS (Desafío Latam) – La captura muestra el flujo de handshake TLS seguido de tráfico cifrado, validando la confidencialidad del sitio.



HTTPS (Emol) – Se identifican múltiples flujos de datos TLS Application Data, típico de un portal con alto volumen de contenido.

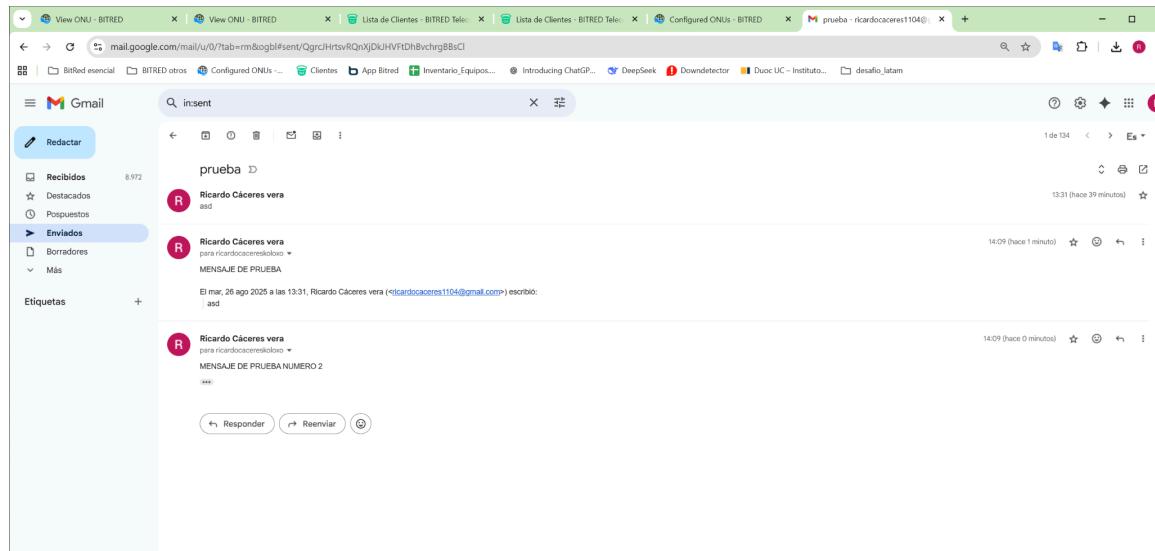


Descarga pequeña (PuTTY) – El tráfico capturado corresponde a transferencia de datos cifrados como Application Data en TLS.



2.c Envío de correo – Evidencia completa

La evidencia se compone de dos partes: primero, la interfaz de Gmail muestra el mensaje enviado en la carpeta 'Enviados'. Segundo, en Wireshark con el filtro `tcp.port==443` se observan paquetes TLSv1.2/1.3 clasificados como Application Data, lo que confirma que el mensaje fue transmitido de forma cifrada.

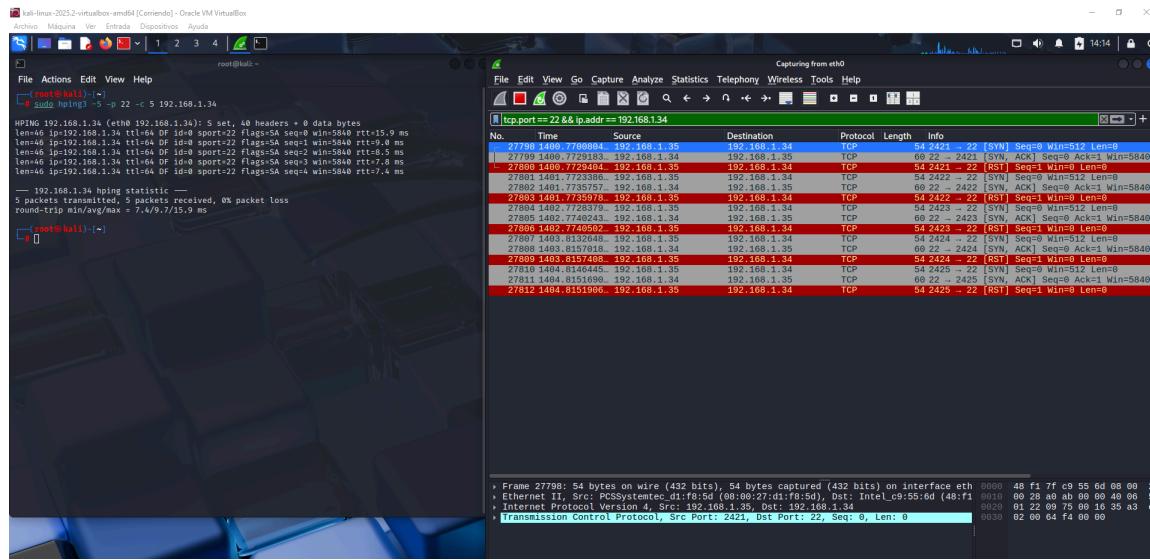


2.d DNS – Consultas y respuestas

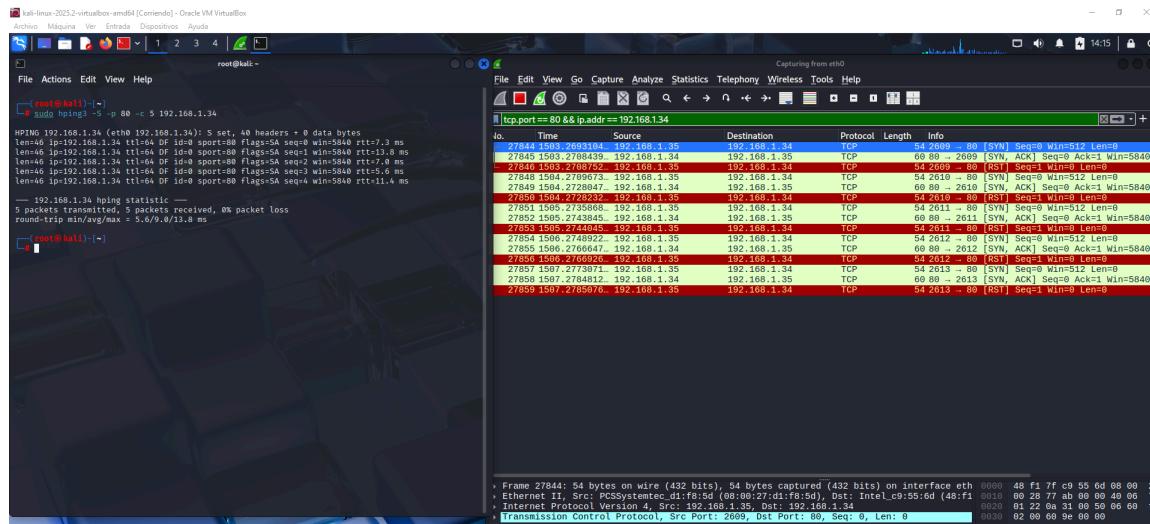
La captura muestra una consulta DNS tipo A hacia contile.services.mozilla.com y su respectiva respuesta, usando el filtro dns.

3. Conectividad hacia 192.168.1.34 (Metasploitable)

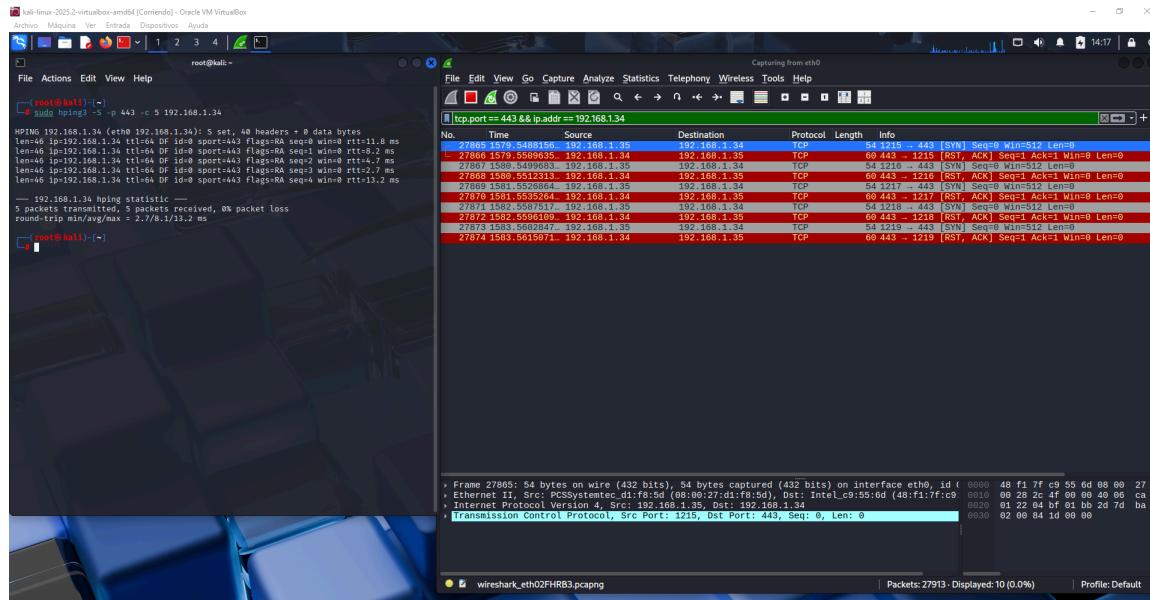
22/TCP (SSH) – Se confirma puerto abierto al recibir SYN/ACK.



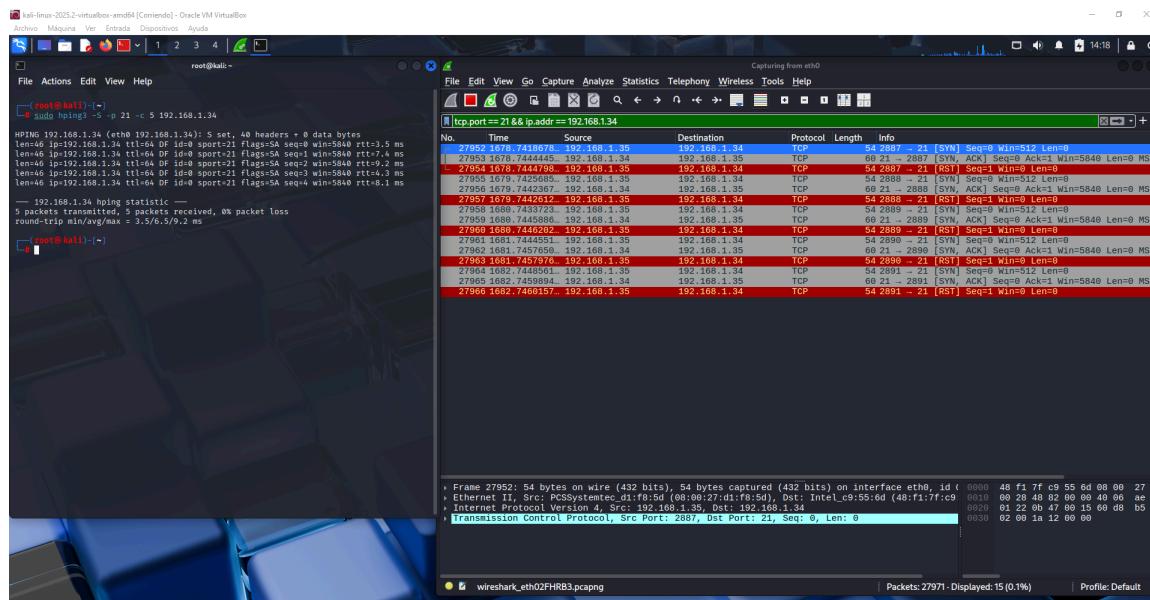
80/TCP (HTTP) – Se confirma puerto abierto al recibir SYN/ACK.



443/TCP (HTTPS) – El host respondió con RST, confirmando que el puerto está cerrado.



21/TCP (FTP) – Se confirma puerto abierto al recibir SYN/ACK.



4. Recomendaciones para mejorar la seguridad de la red analizada

1. **Eliminar servicios innecesarios:** Deshabilitar FTP y cualquier otro puerto que no se utilice en producción, reduciendo la superficie de ataque.
2. **Migrar a protocolos seguros:** Sustituir HTTP por HTTPS en todos los servicios, asegurando el uso de TLS 1.3 o superior.
3. **Segmentar y controlar accesos:** Aplicar VLANs, ACLs y reglas de firewall para limitar la comunicación entre áreas críticas de la red.
4. **Monitoreo y alertas:** Implementar un sistema de registros centralizados (Syslog) junto a IDS/IPS para detectar y responder a incidentes en tiempo real.
5. **Buenas prácticas de autenticación:** Aplicar contraseñas robustas, doble factor de autenticación y, de ser posible, integrar AAA (TACACS+/RADIUS) para gestionar accesos.
6. **Actualizaciones y parches:** Mantener los sistemas y aplicaciones con sus últimas actualizaciones de seguridad.

Conclusion

El testeo permitió identificar diferencias críticas entre protocolos inseguros y seguros: mientras que HTTP transmite información en claro, HTTPS garantiza confidencialidad gracias al cifrado TLS. También se comprobó la presencia de servicios expuestos en la máquina objetivo (como FTP y HTTP) que representan riesgos si no son controlados.

Las pruebas de puertos confirmaron el estado real de cada servicio (abierto, cerrado o filtrado), y el análisis con Wireshark evidenció la utilidad de aplicar filtros adecuados para interpretar tráfico y detectar posibles vulnerabilidades.

En conclusión, la red cumple con su función operativa, pero presenta debilidades que deben corregirse. La implementación de protocolos cifrados, la reducción de servicios expuestos y el fortalecimiento de controles de acceso son medidas fundamentales para garantizar la disponibilidad, integridad y confidencialidad de la información.