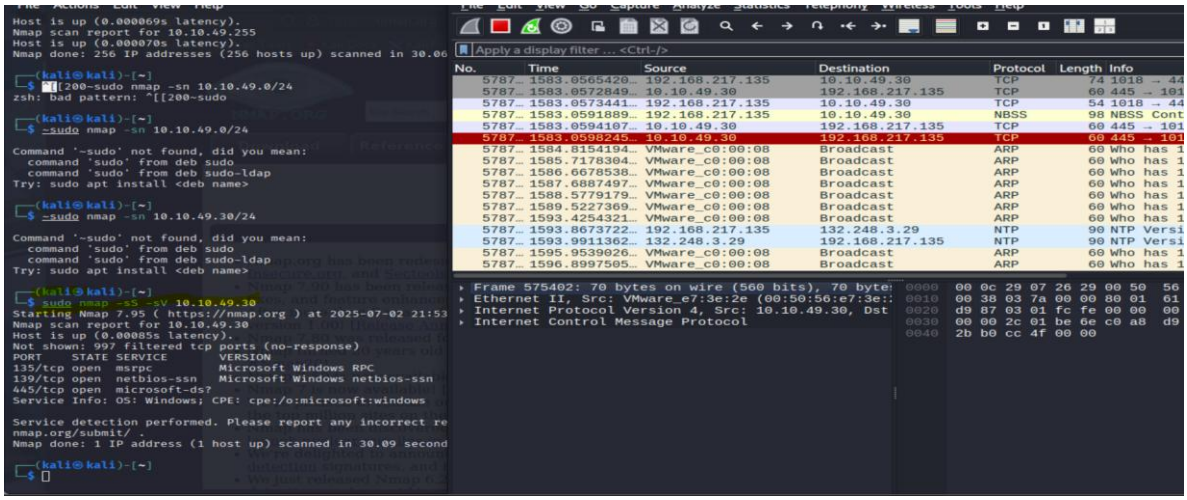
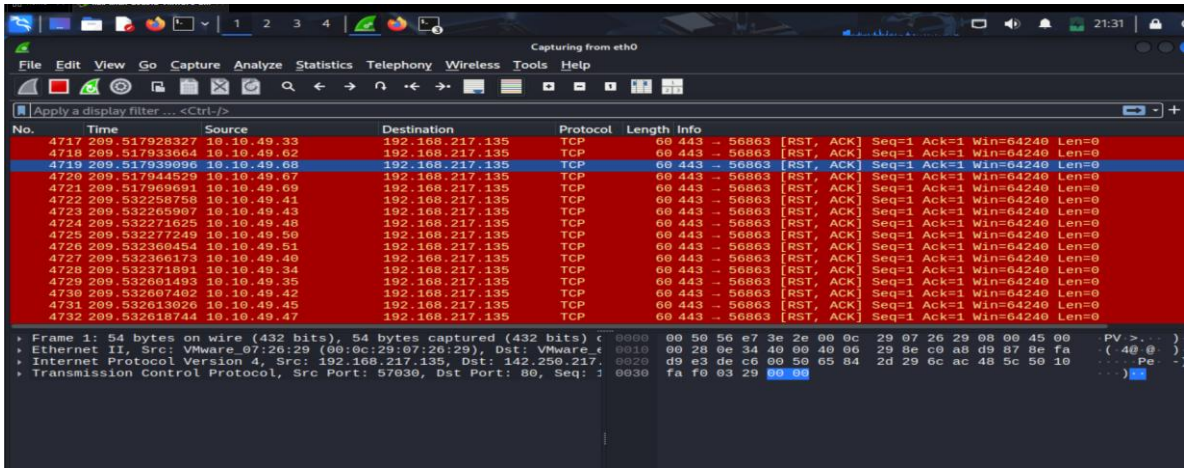
Ejercicio : Reconocimento Activo

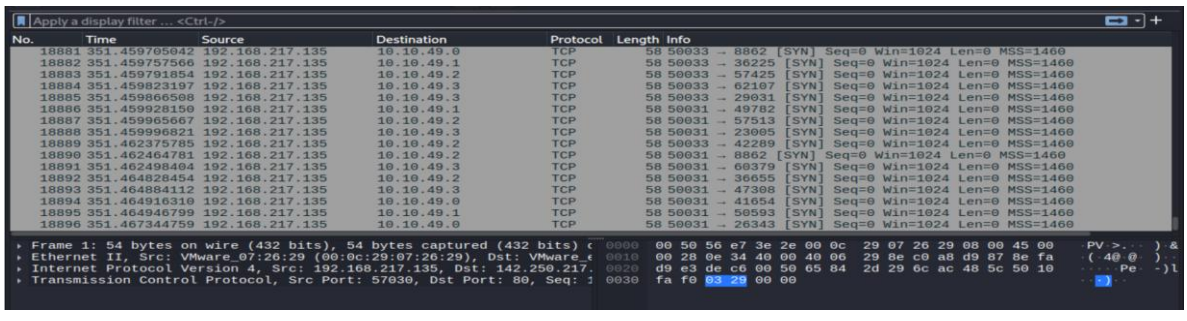
En Wireshark deberían ver:

- Tráfico SYN enviado a múltiples IPs del segmento.

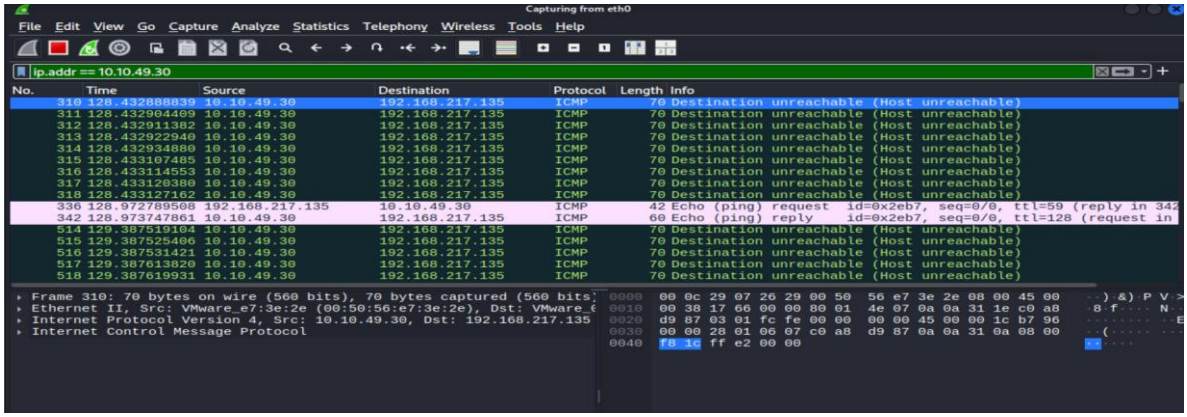


- Respuestas SYN-ACK desde los hosts activos.

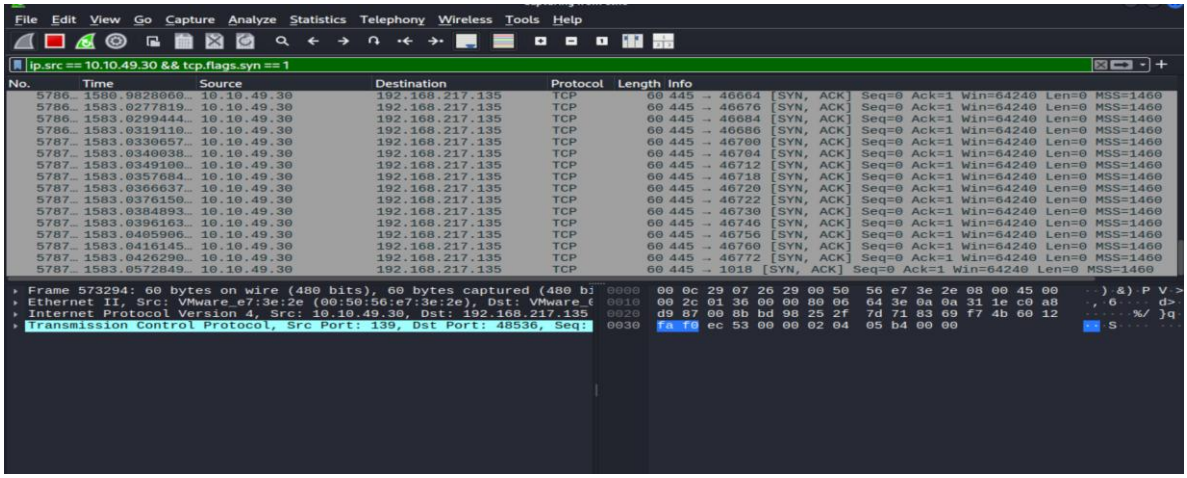




- Tráfico ICMP si usan ping scan.



- Escaneos dirigidos a múltiples puertos por host.



Comandos usados

sudo nmap -sn 10.10.49.0/24

sudo nmap -sS -p1-1000 10.10.49.30

sudo nmap -sS -p- 10.10.49.30

En Wireshark

ip.src == 10.10.49.30 && tcp.flags.syn == 1

Reflexion

El escaneo permitió identificar dispositivos activos y servicios expuestos. Algunos hosts representan riesgo al tener puertos abiertos innecesarios,

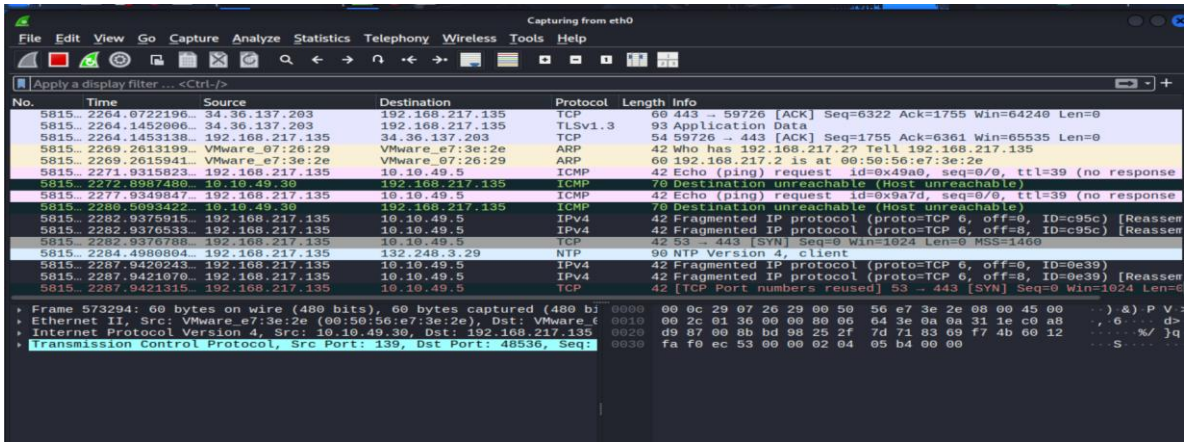
---

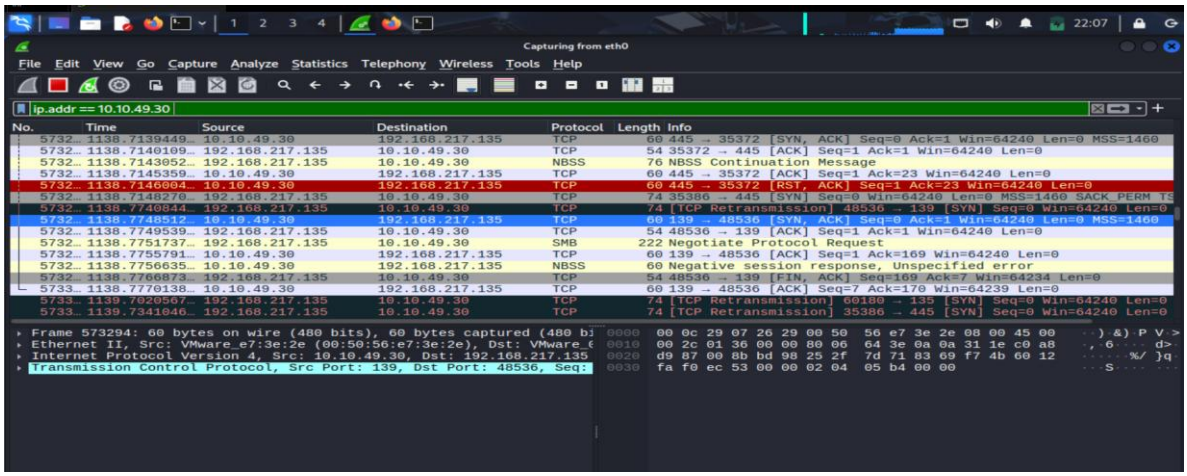
Ejercicio 2: Escaneo sigiloso a un host en tu red

Escoge un host dentro de tu red y realiza un escaneo que utilice técnicas de evasión para evitar su detección por firewalls o sistemas de monitoreo. Evalúa si lograste obtener información sin generar tráfico evidente.

En Wireshark deberían ver:

- Tráfico con fragmentación de paquetes TCP/IP.
- Uso de un puerto fuente no estándar (ej. 53, 123).
- Intervalos largos entre los paquetes (bajo volumen).
- Tráfico que no completa handshakes TCP.





Comandos usados

sudo nmap -sS -f --source-port 53 --scan-delay 5s 10.10.49.5

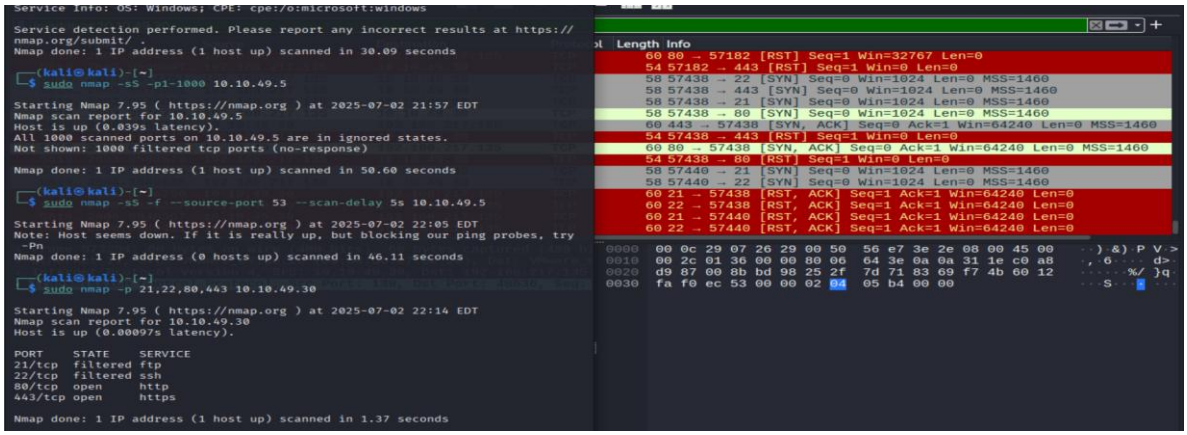
ip.addr == 10.10.49.30

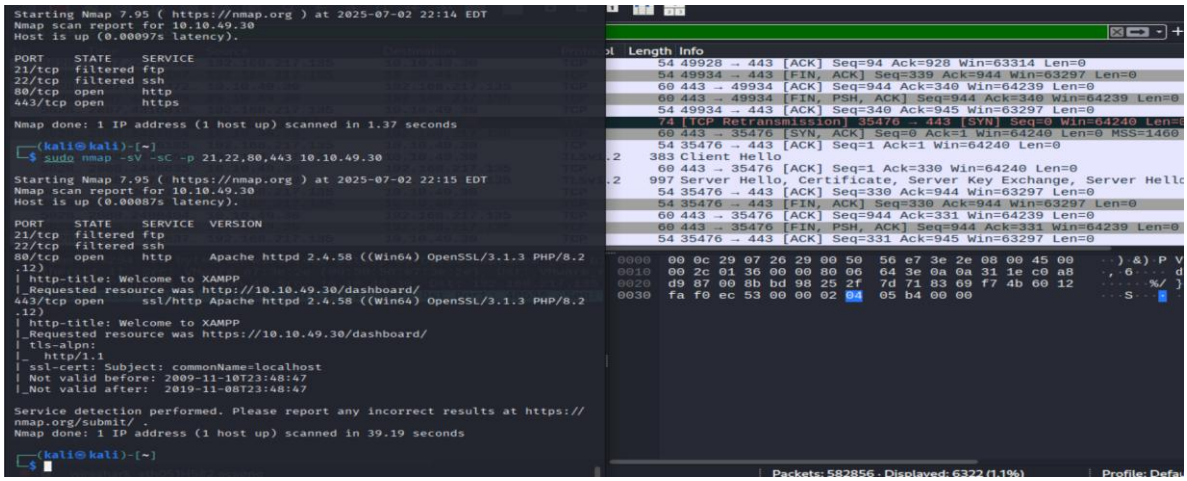
Ejercicio 3: Enumeración avanzada de servicios

Identifica un host dentro de tu red que tenga servicios web, FTP, o SSH, y utiliza técnicas avanzadas para obtener información detallada de esos servicios (como banners, versiones, métodos HTTP, etc.).

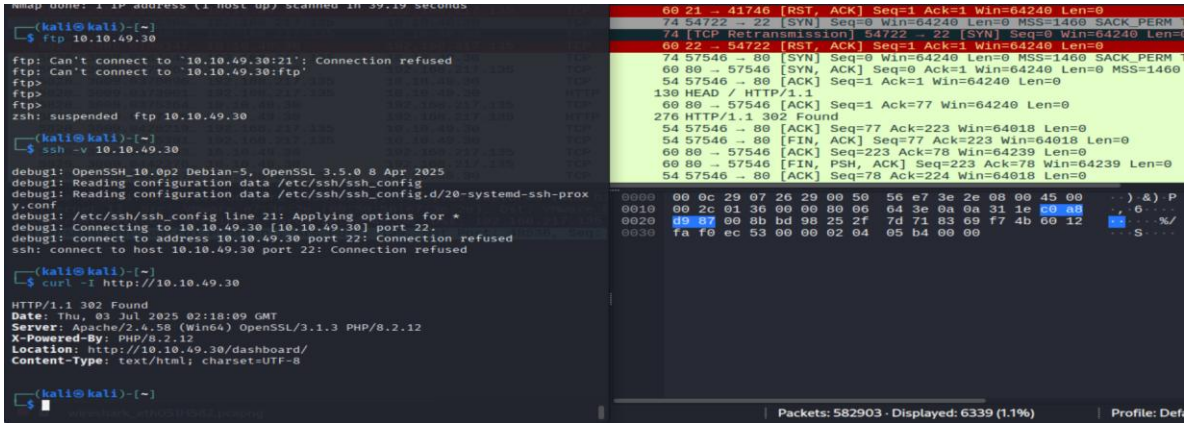
En Wireshark deberían ver:

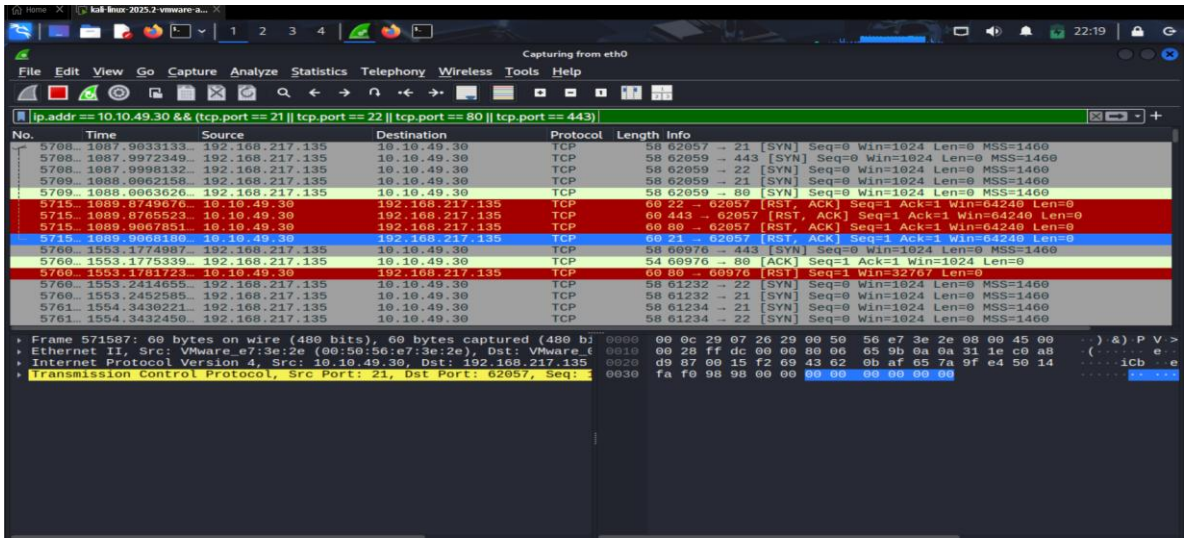
- Solicitudes hacia puertos 21, 22, 80, 443, u otros comunes.





- Tráfico con comandos FTP, HTTP o SSH.
- Respuestas con datos identificables: versiones de servicios, encabezados HTTP, mensajes de bienvenida de FTP/SSH.





Comandos Usados

sudo nmap -p 21,22,80,443 10.10.49.30

sudo nmap -sV -sC -p 21,22,80,443 10.10.49.30

ftp 10.10.49.30

ssh -v 10.10.49.30

curl -I http://10.10.49.30

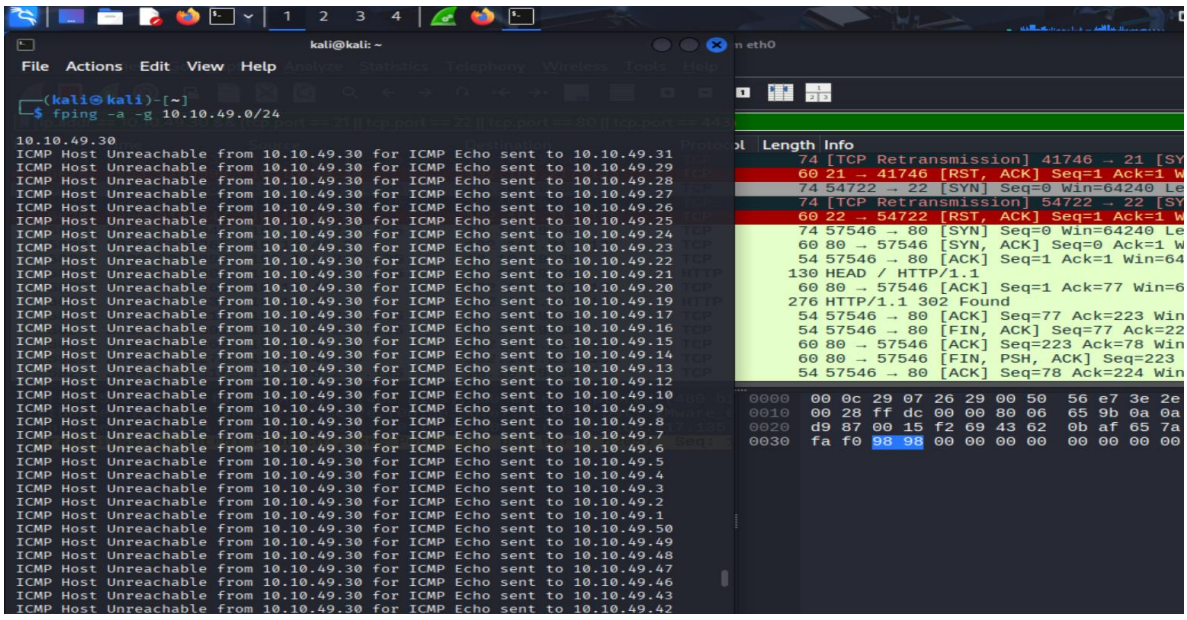
ip.addr == 10.10.49.30 && (tcp.port == 21 || tcp.port == 22 || tcp.port == 80 || tcp.port == 443)

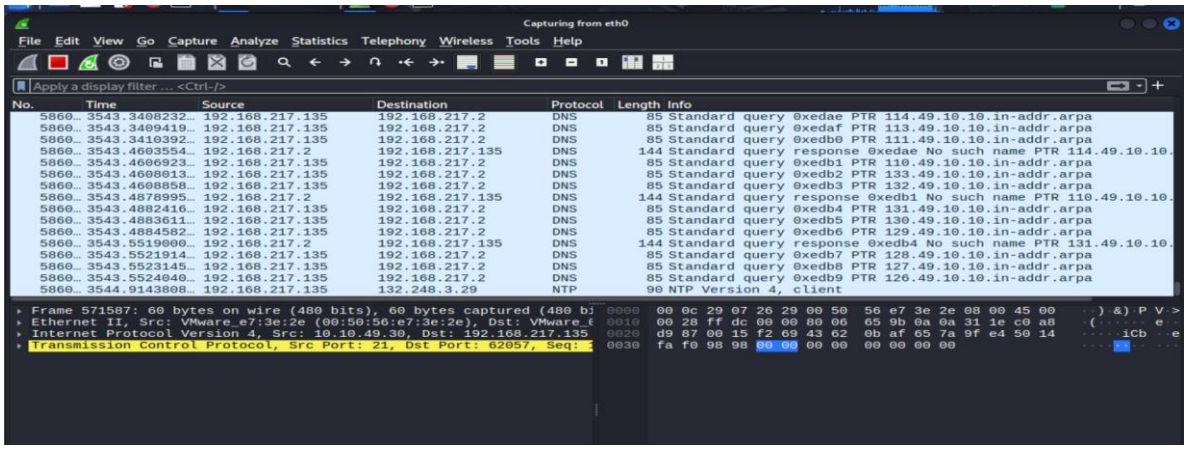
Ejercicio 4: Detección de hosts sin ICMP habilitado

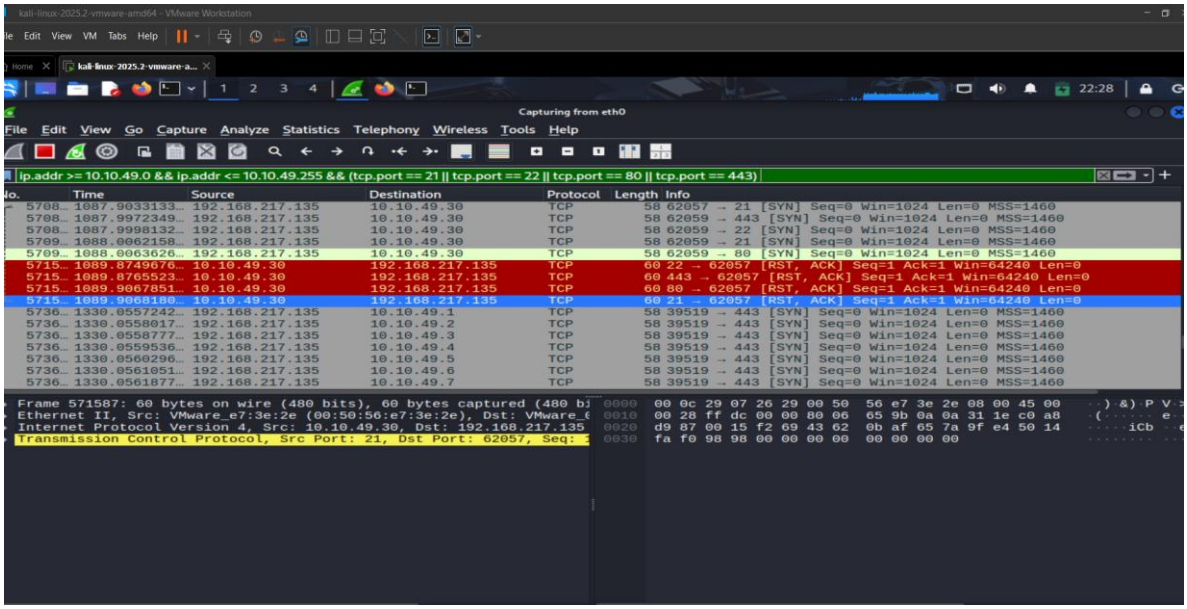
Encuentra dentro de tu red aquellos hosts que no responden a ping (ICMP), pero que tienen puertos abiertos accesibles. Analiza si puedes detectarlos sin depender de ICMP.

En Wireshark deberían ver:

- Escaneos TCP sin tráfico ICMP.
- Solicitudes TCP SYN enviadas directamente a puertos específicos.
- Respuestas SYN-ACK de hosts que no respondieron al ping.




Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
5860… 3543.3408232… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedae PTR 114.49.10.10.in-addr.arpa
5860… 3543.3409419… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedaf PTR 113.49.10.10.in-addr.arpa
5860… 3543.3410392… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedb0 PTR 111.49.10.10.in-addr.arpa
5860… 3543.4603554… 192.168.217.2 192.168.217.135 DNS 144 Standard query response 0xedae No such name PTR 114.49.10.10.
5860… 3543.4606923… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedb1 PTR 110.49.10.10.in-addr.arpa
5860… 3543.4608013… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedb2 PTR 133.49.10.10.in-addr.arpa
5860… 3543.4608858… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedb3 PTR 132.49.10.10.in-addr.arpa
5860… 3543.4878995… 192.168.217.2 192.168.217.135 DNS 144 Standard query response 0xedb1 No such name PTR 110.49.10.10.
5860… 3543.4882416… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedb4 PTR 131.49.10.10.in-addr.arpa
5860… 3543.4883611… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedb5 PTR 130.49.10.10.in-addr.arpa
5860… 3543.4884582… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedb6 PTR 129.49.10.10.in-addr.arpa
5860… 3543.5519000… 192.168.217.2 192.168.217.135 DNS 144 Standard query response 0xedb4 No such name PTR 131.49.10.10.
5860… 3543.5521914… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedb7 PTR 128.49.10.10.in-addr.arpa
5860… 3543.5523145… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedb8 PTR 127.49.10.10.in-addr.arpa
5860… 3543.5524040… 192.168.217.135 192.168.217.2 DNS 85 Standard query 0xedb9 PTR 126.49.10.10.in-addr.arpa
5860… 3544.9143808… 192.168.217.135 132.248.3.29 NTP 90 NTP Version 4, client
Frame 571587: 60 bytes on wire (480 bits), 60 bytes captured (480 bi
Ethernet II, Src: VMware_e7:3e:2e (00:50:56:e7:3e:2e), Dst: VMware_
Internet Protocol Version 4, Src: 10.10.49.30, Dst: 192.168.217.135
Transmission Control Protocol, Src Port: 21, Dst Port: 62057, Seq:
0000 00 0c 29 07 26 29 00 50 56 e7 3e 2e 08 00 45 00
0010 00 28 ff dc 00 00 80 06 65 9b 0a 0a 31 1e c0 a8
0020 d9 87 00 15 f2 69 43 62 0b af 65 7a 9f e4 50 14
0030 fa f0 98 98 00 00 00 00 00 00 00 00


kali-linux-2025.2-vmware-amd64 - VMware Workstation
File Edit View VM Tabs Help
Home
kali-linux-2025.2-vmware-a...
22:28
Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr >= 10.10.49.0 && ip.addr <= 10.10.49.255 && (tcp.port == 21 || tcp.port == 22 || tcp.port == 80 || tcp.port == 443)
No. Time Source Destination Protocol Length Info
5708… 1087.9033133… 192.168.217.135 10.10.49.30 TCP 58 62057 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5708… 1087.9972349… 192.168.217.135 10.10.49.30 TCP 58 62059 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5708… 1087.9998132… 192.168.217.135 10.10.49.30 TCP 58 62059 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5709… 1088.0062158… 192.168.217.135 10.10.49.30 TCP 58 62059 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5709… 1088.0063626… 192.168.217.135 10.10.49.30 TCP 58 62059 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5715… 1089.8749676… 10.10.49.30 192.168.217.135 TCP 60 22 → 62057 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
5715… 1089.8765523… 10.10.49.30 192.168.217.135 TCP 60 443 → 62057 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
5715… 1089.9067851… 10.10.49.30 192.168.217.135 TCP 60 80 → 62057 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
5715… 1089.9068180… 10.10.49.30 192.168.217.135 TCP 60 21 → 62057 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
5736… 1330.0557242… 192.168.217.135 10.10.49.1 TCP 58 39519 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5736… 1330.0558017… 192.168.217.135 10.10.49.2 TCP 58 39519 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5736… 1330.0558777… 192.168.217.135 10.10.49.3 TCP 58 39519 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5736… 1330.0559536… 192.168.217.135 10.10.49.4 TCP 58 39519 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5736… 1330.0560296… 192.168.217.135 10.10.49.5 TCP 58 39519 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5736… 1330.0561051… 192.168.217.135 10.10.49.6 TCP 58 39519 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
5736… 1330.0561877… 192.168.217.135 10.10.49.7 TCP 58 39519 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
Frame 571587: 60 bytes on wire (480 bits), 60 bytes captured (480 bi
Ethernet II, Src: VMware_e7:3e:2e (00:50:56:e7:3e:2e), Dst: VMware_
Internet Protocol Version 4, Src: 10.10.49.30, Dst: 192.168.217.135
Transmission Control Protocol, Src Port: 21, Dst Port: 62057, Seq:
0000 00 0c 29 07 26 29 00 50 56 e7 3e 2e 08 00 45 00
0010 00 28 ff dc 00 00 80 06 65 9b 0a 0a 31 1e c0 a8
0020 d9 87 00 15 f2 69 43 62 0b af 65 7a 9f e4 50 14
0030 fa f0 98 98 00 00 00 00 00 00 00 00