

---

## **Práctica 9.2: ARP Poisoning (ARP Spoofing)**

Ataque de Envenenamiento ARP y Técnicas de Mitigación

Uriel Felipe Vázquez Orozco, Euler Molina Martínez

Enero 27, 2026



## Resumen Ejecutivo

Esta práctica documenta la implementación y demostración del ataque ARP Poisoning (también conocido como ARP Spoofing), una técnica de ataque de capa 2 que permite a un atacante interceptar, modificar o detener el tráfico de red mediante la manipulación de las tablas ARP de los dispositivos víctima. El objetivo es comprender el funcionamiento del protocolo ARP, sus vulnerabilidades inherentes, y las contramedidas efectivas para proteger la infraestructura de red.

**Resultados:** Se logró demostrar el ataque de envenenamiento ARP utilizando herramientas especializadas en una topología de laboratorio. Se verificó la capacidad del atacante para posicionarse como Man-in-the-Middle (MITM) entre una víctima y el gateway, capturando tráfico en tiempo real. Posteriormente, se implementaron las contramedidas Dynamic ARP Inspection (DAI) y DHCP Snooping que mitigan completamente esta vulnerabilidad.

## Identificación del Problema

El protocolo ARP (Address Resolution Protocol) presenta vulnerabilidades de diseño fundamentales:

- **Sin autenticación:** ARP no verifica la identidad del remitente
- **Respuestas no solicitadas:** Los hosts aceptan respuestas ARP sin haberlas solicitado (Gratuitous ARP)
- **Sin cifrado:** Las tramas ARP viajan en texto claro
- **Caché mutable:** La tabla ARP puede ser actualizada dinámicamente por cualquier host



**Amenaza de seguridad:** Un atacante puede enviar respuestas ARP falsas para asociar su dirección MAC con la IP de otro dispositivo (como el gateway), redirigiendo todo el tráfico hacia sí mismo antes de reenviarlo al destino legítimo.

### Impacto del ataque:

1. **Intercepción de tráfico:** Captura de credenciales, datos sensibles
2. **Modificación de datos:** Alteración de información en tránsito
3. **Denegación de servicio:** Interrupción de conectividad
4. **Secuestro de sesiones:** Apropiación de sesiones autenticadas

## Metodología Aplicada

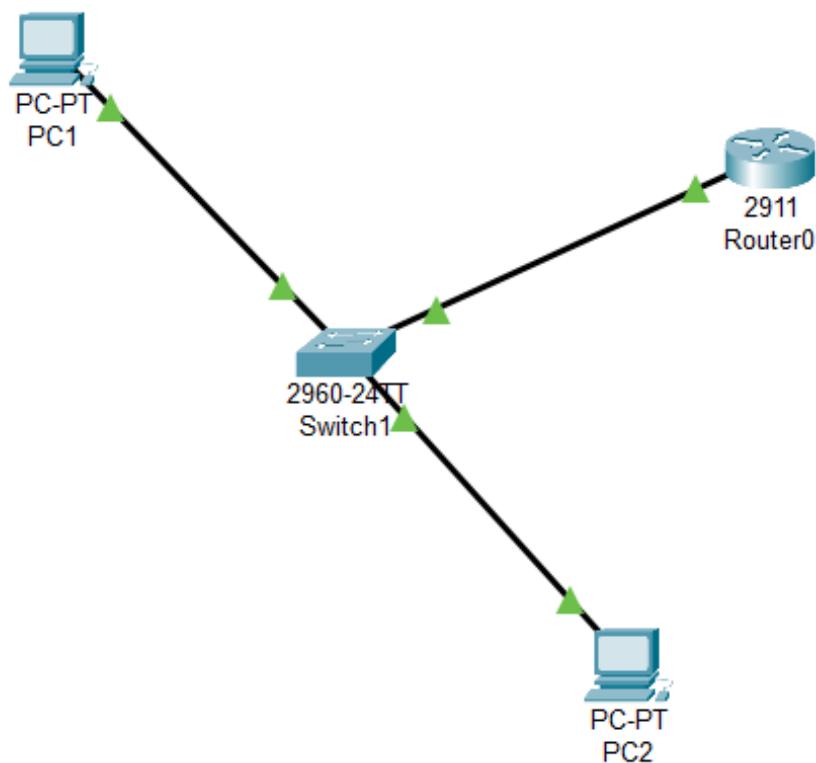
### Equipos utilizados:

- 1 Switch Cisco Catalyst 2960 con IOS 15.x
- 1 Router Cisco 2911 como gateway
- 1 PC atacante con sistema Linux (Kali Linux)

- 1 PC víctima con Windows/Linux
- Herramienta principal: **Ettercap** (suite completa para ataques MITM)
- Herramienta de análisis: Wireshark

**Proceso:**

1. **Diseño de topología:** Configuración de red de laboratorio
2. **Análisis del protocolo ARP:** Estudio del funcionamiento normal
3. **Ejecución del ataque con Ettercap:** Demostración de ARP Poisoning
4. **Captura de tráfico:** Verificación de intercepción MITM
5. **Implementación de contramedidas:** DAI y DHCP Snooping
6. **Validación de mitigación:** Verificación de protección efectiva

**Topología de Red Implementada****Figure 1:** Topología de red para ataque ARP Poisoning**Descripción de la topología:**

- **R1 (Gateway):** Router que actúa como puerta de enlace - 192.168.10.1
- **SW1:** Switch de acceso donde se conectan todos los hosts
- **Atacante:** PC con Kali Linux en 192.168.10.100
- **Víctima:** PC objetivo del ataque en 192.168.10.50
- **Test:** PC adicional para pruebas en 192.168.10.60

### Configuración de direccionamiento:

Dispositivo	Interfaz	Dirección IP	Dirección MAC	Rol
R1 (Gateway)	Gi0/0	192.168.10.1/24	00:1A:2B:3C:4D:01	Gateway de red
SW1	VLAN 10	192.168.10.2/24	00:1A:2B:3C:4D:02	Switch de acceso
Atacante	eth0	192.168.10.100/24	00:0C:29:AA:BB:CC	Origen del ataque
Víctima	eth0	192.168.10.50/24	00:0C:29:DD:EE:FF	Objetivo del ataque

### Configuración de VLAN:

VLAN ID	Nombre	Propósito
10	LAN_Interna	Red de usuarios

## Configuración Inicial

### Configuración del Router Gateway

#### Cisco IOS Terminal

```

Router> enable
Router# configure terminal
Router(config)# hostname R1
R1(config)# interface GigabitEthernet0/0
R1(config-if)# description Gateway LAN Interna

```

```
R1(config-if)# ip address 192.168.10.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# end
R1# write memory
```

## Configuración del Switch (Sin Protecciones Iniciales)

### Cisco IOS Terminal

```
Switch> enable
Switch# configure terminal
Switch(config)# hostname SW1
SW1(config)# vlan 10
SW1(config-vlan)# name LAN_Interna
SW1(config-vlan)# exit
SW1(config)# interface range FastEthernet0/1-3
SW1(config-if-range)# switchport mode access
SW1(config-if-range)# switchport access vlan 10
SW1(config-if-range)# exit
SW1(config)# interface GigabitEthernet0/1
SW1(config-if)# description Uplink a Router
SW1(config-if)# switchport mode access
SW1(config-if)# switchport access vlan 10
SW1(config-if)# exit
SW1(config)# interface vlan 10
SW1(config-if)# ip address 192.168.10.2 255.255.255.0
SW1(config-if)# exit
SW1(config)# end
SW1# write memory
```



**Configuración vulnerable:** Esta configuración inicial no tiene protecciones contra ARP Poisoning habilitadas, permitiendo que cualquier host envíe respuestas ARP falsas.

## Desarrollo Detallado

### Paso 1: Comprepción del Protocolo ARP

#### Funcionamiento Normal de ARP

- ARP Request:** Host A necesita la MAC de Host B, envía broadcast “Quien tiene 192.168.10.1?”
- ARP Reply:** Host B responde con su MAC “192.168.10.1 esta en 00:1A:2B:3C:4D:01”
- Actualización de caché:** Host A almacena la asociación IP-MAC en su tabla ARP

```
# Ver tabla ARP en Linux
$ arp -n
Address          HWtype  HWaddress           Flags Mask   Iface
192.168.10.1    ether    00:1a:2b:3c:4d:01  C      eth0
```

#### Cisco IOS Terminal

```
! Ver tabla ARP en Cisco
R1# show ip arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 192.168.10.1 - 001a.2b3c.4d01 ARPA GigabitEthernet0/0
Internet 192.168.10.50 10 000c.29dd.eeff ARPA GigabitEthernet0/0
Internet 192.168.10.100 12 000c.29aa.bbcc ARPA GigabitEthernet0/0
```

### Gratuitous ARP

Un Gratuitous ARP es una respuesta ARP no solicitada que los hosts envían para:

- Anunciar su presencia en la red
- Actualizar caches ARP después de cambios de IP
- Detectar conflictos de IP



**Vulnerabilidad:** Los hosts aceptan estos paquetes sin validación, permitiendo que un atacante actualice las tablas ARP de otros dispositivos.

### Paso 2: Ejecución del Ataque ARP Poisoning con Ettercap

Ettercap es una suite completa para ataques Man-in-the-Middle que automatiza todo el proceso de envenenamiento ARP, incluyendo el reenvío de paquetes.

## Inicio de Ettercap

```
# Iniciar ettercap en modo grafico  
$ sudo ettercap -G
```

## Paso 2.1: Escaneo de Hosts

Una vez iniciado Ettercap:

1. Seleccionar **Sniff** → **Unified sniffing**
2. Elegir la interfaz de red (eth0)
3. Ir a **Hosts** → **Scan for hosts**
4. Ver los hosts descubiertos en **Hosts** → **Hosts list**

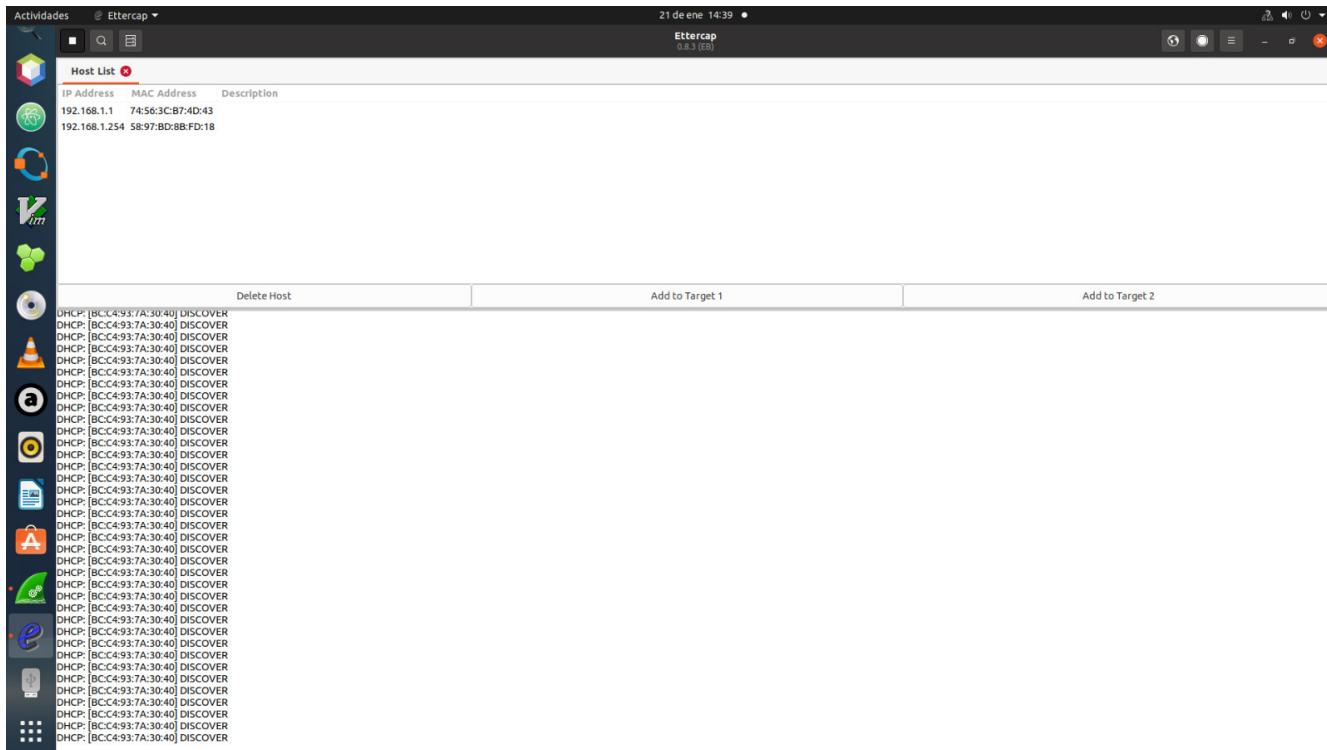


Figure 2: Lista de hosts descubiertos en Ettercap

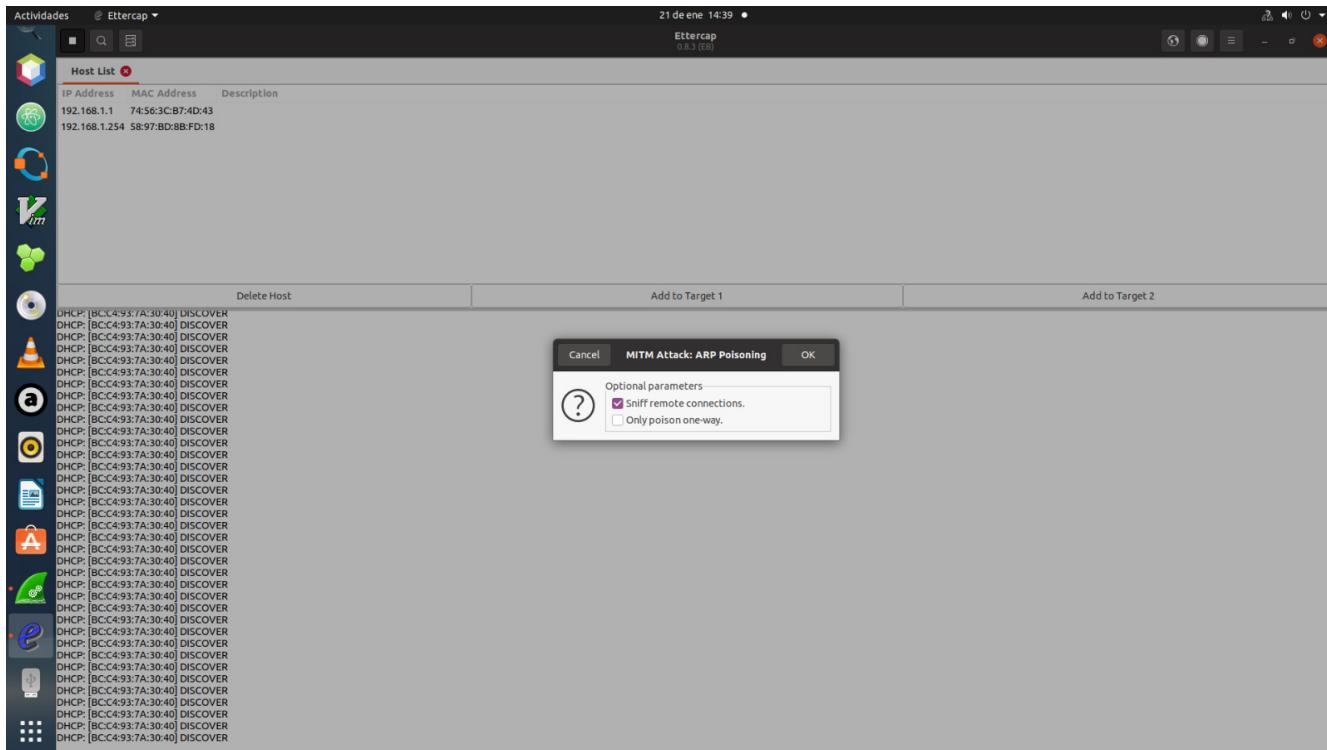
## Paso 2.2: Selección de Objetivos

Desde la lista de hosts:

1. Seleccionar el **Gateway** (192.168.10.1) → Click en **Add to Target 1**
2. Seleccionar la **Víctima** (192.168.10.50) → Click en **Add to Target 2**

### Paso 2.3: Configuración del Ataque ARP Poisoning

1. Ir a **Mitm** → **ARP poisoning**
2. Marcar la opción **Sniff remote connections**
3. Click en **OK** para iniciar el envenenamiento



**Figure 3:** Parámetros del ataque MITM ARP Poisoning

### Paso 2.4: Inicio del Ataque

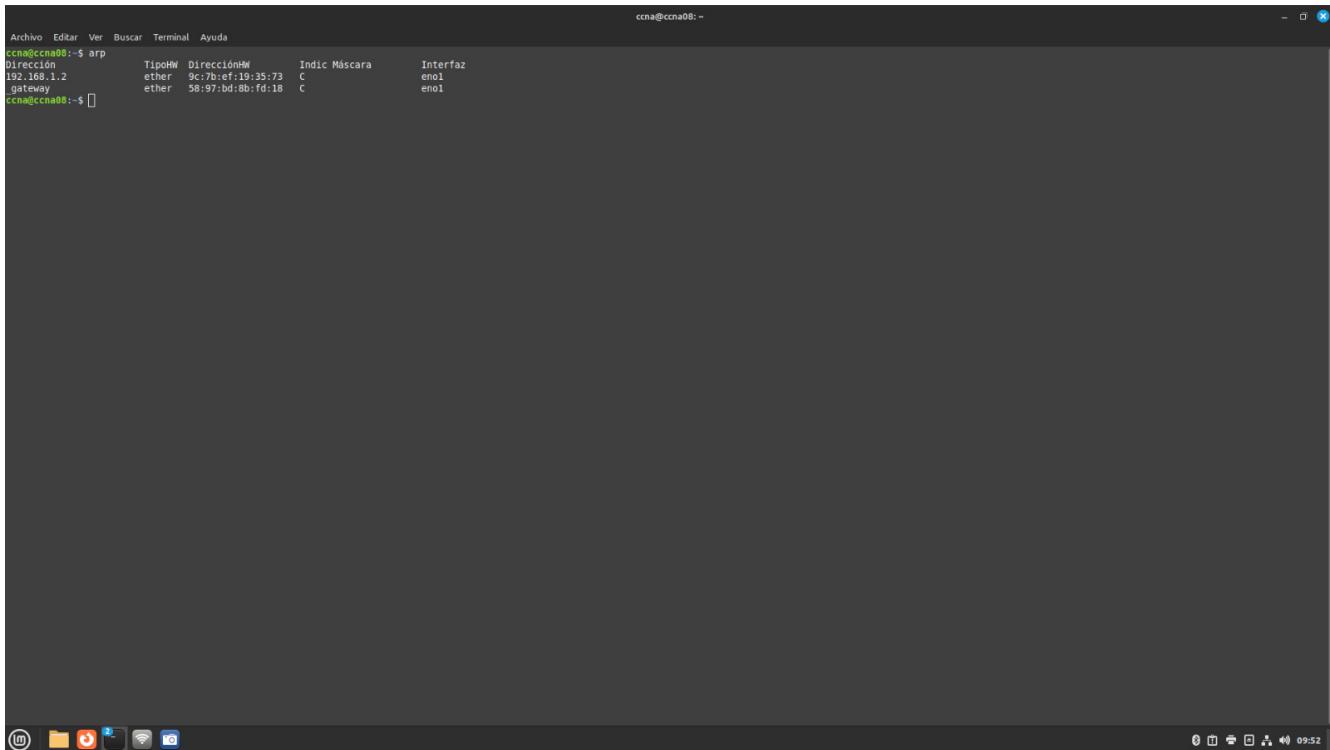
1. Ir a **Start** → **Start sniffing**
2. Ettercap comenzará a enviar paquetes ARP falsos automáticamente
3. Todo el tráfico entre la víctima y el gateway pasará por el atacante



**Ventaja de Ettercap:** A diferencia de arpspoof que requiere dos terminales, Ettercap maneja automáticamente el envenenamiento bidireccional y el reenvío de paquetes (IP forwarding).

### Paso 3: Verificación del Ataque

#### Tabla ARP de la Víctima (Antes del Ataque)



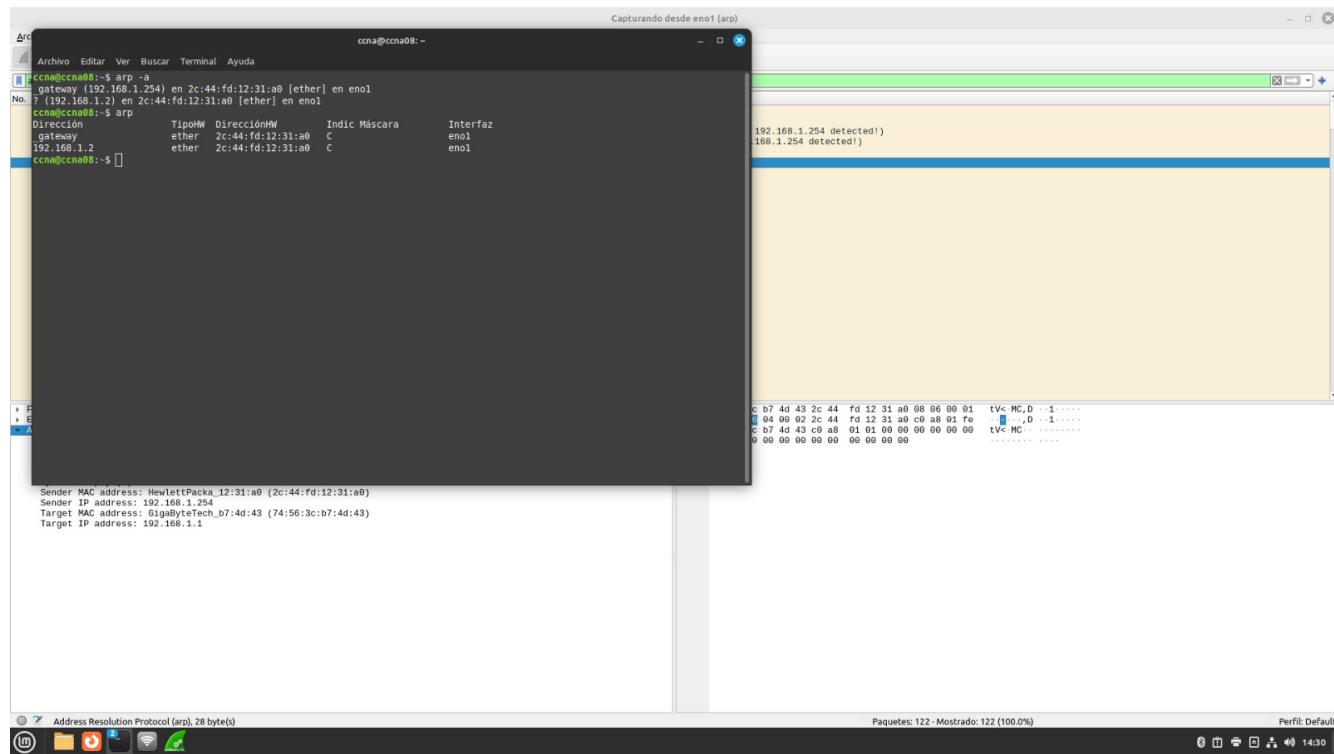
A screenshot of a terminal window titled "ccna@ccna08: ~". The window contains the output of the "arp" command. The output shows one entry for the gateway:

Dirección IP	Tipo HW	Dirección HW	Indic	Máscara	Interfaz
192.168.1.2	ether	9c:7b:ef:19:35:73	C		en0
gateway	ether	58:97:bd:8b:fd:18	C		en0

The terminal prompt "ccna@ccna08:~" is visible at the bottom.

**Figure 4:** Tabla ARP antes de ser envenenada

La tabla ARP muestra la MAC real del gateway.

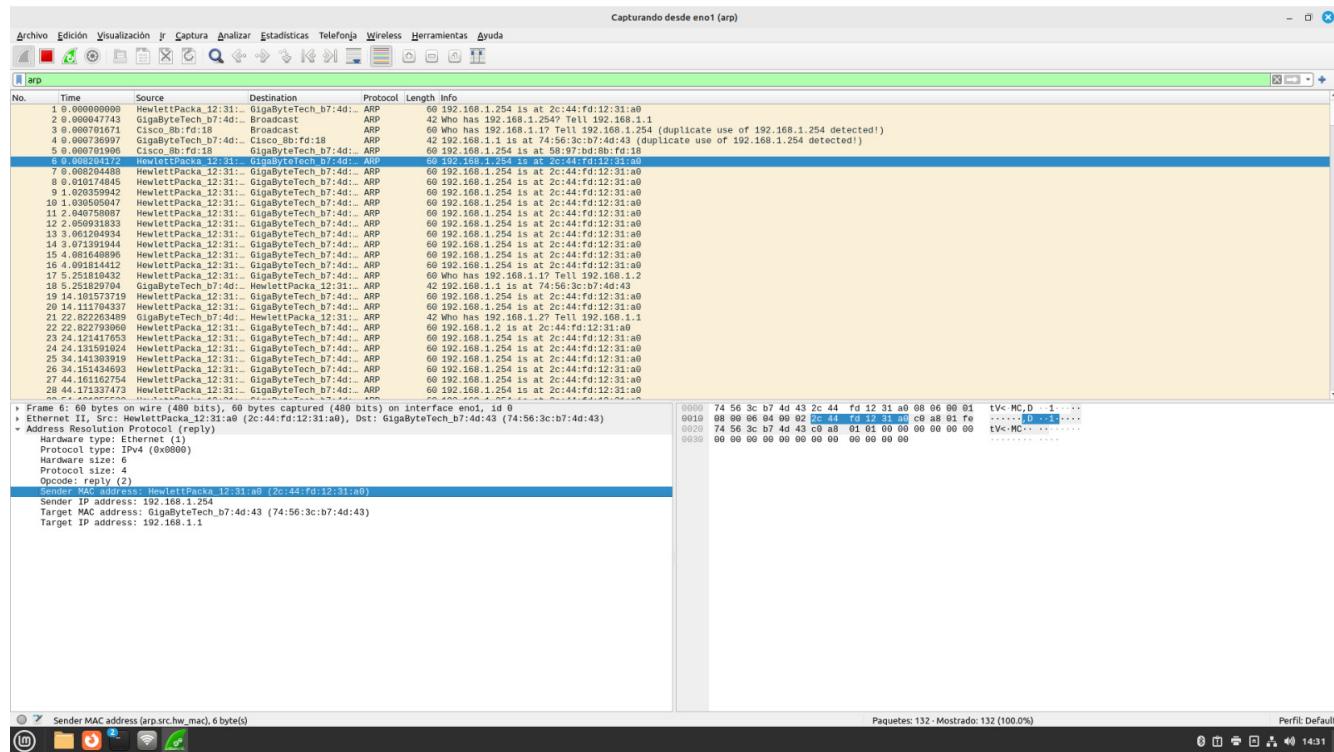
**Tabla ARP de la Víctima (Durante el Ataque)**

**Figure 5:** Tabla ARP envenenada - misma MAC para gateway y atacante

**✗ Tabla ARP envenenada:** La MAC del gateway ahora apunta a la MAC del atacante en lugar de la MAC real del router. Ambas entradas (gateway y atacante) muestran la misma dirección MAC.

**Captura de Tráfico con Wireshark**

Simultáneamente, se puede usar Wireshark para analizar el tráfico interceptado:



**Figure 6:** Captura de Wireshark durante el ataque

Filtros útiles en Wireshark:

- arp - Ver paquetes ARP del envenenamiento
- http - Ver tráfico HTTP con credenciales
- tcp - Ver todo el tráfico TCP interceptado
- ip.addr == 192.168.10.50 - Filtrar por IP de la víctima

✓ **Intercepción exitosa:** El atacante puede ver todo el tráfico entre la víctima y el gateway, incluyendo credenciales en protocolos no cifrados.

### Captura de Credenciales HTTP

```
# Usando dsniff para capturar credenciales automáticamente
$ sudo dsniff -i eth0
dsniff: listening on eth0
-----
01/27/26 15:32:45  tcp 192.168.10.50.54321 -> 93.184.216.34.80 (http)
GET /login HTTP/1.1
Authorization: Basic dXNlcjpwYXNzd29yZA==
```

## Paso 4: Análisis del Impacto

### Estadísticas del Ataque

#### Cisco IOS Terminal

```
SW1# show mac address-table
Mac Address Table
-----
Vlan Mac Address Type Ports
-----
10 000c.29aa.bbcc DYNAMIC Fa0/1 (Atacante)
10 000c.29dd.eeff DYNAMIC Fa0/2 (Victima)
10 001a.2b3c.4d01 DYNAMIC Gi0/1 (Gateway)
```

### Verificación en el Gateway

#### Cisco IOS Terminal

```
R1# show ip arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 192.168.10.1 - 001a.2b3c.4d01 ARPA GigabitEthernet0/0
Internet 192.168.10.50 2 000c.29aa.bbcc ARPA GigabitEthernet0/0
! NOTA: La MAC de la victima ahora muestra la MAC del atacante
```

## Implementación de Contramedidas

### Contramedida 1: DHCP Snooping

DHCP Snooping es prerequisito para Dynamic ARP Inspection:

#### Cisco IOS Terminal

```
SW1(config)# ip dhcp snooping
SW1(config)# ip dhcp snooping vlan 10
SW1(config)# no ip dhcp snooping information option
! Configurar puertos trusted (hacia el servidor DHCP/router)
SW1(config)# interface GigabitEthernet0/1
SW1(config-if)# description Uplink a Router (DHCP Server)
```

```
SW1(config-if)# ip dhcp snooping trust  
SW1(config-if)# exit  
! Los puertos de acceso permanecen untrusted por defecto
```

**i** **DHCP Snooping** crea una tabla de binding que asocia direcciones MAC con IPs asignadas por DHCP. Esta tabla es utilizada por DAI para validar paquetes ARP.

## Contramedida 2: Dynamic ARP Inspection (DAI)

### Cisco IOS Terminal

```
SW1(config)# ip arp inspection vlan 10  
! Configurar puertos trusted  
SW1(config)# interface GigabitEthernet0/1  
SW1(config-if)# ip arp inspection trust  
SW1(config-if)# exit  
! Configurar rate limiting en puertos de acceso  
SW1(config)# interface range FastEthernet0/1-3  
SW1(config-if-range)# ip arp inspection limit rate 15  
SW1(config-if-range)# exit
```

## Contramedida 3: Entradas ARP Estáticas (Para IPs Fijas)

Para dispositivos con IP estática que no usan DHCP:

### Cisco IOS Terminal

```
SW1(config)# ip source binding 000c.29dd.eeff vlan 10 192.168.10.50 interface FastEthernet0/2
```

O configurar ARP estático en los hosts:

```
# En la víctima (Linux)  
$ sudo arp -s 192.168.10.1 00:1a:2b:3c:4d:01  
  
# En Windows  
C:\> arp -s 192.168.10.1 00-1a-2b-3c-4d-01
```

## Contramedida 4: Validaciones Adicionales de DAI

### Cisco IOS Terminal

```
SW1(config)# ip arp inspection validate src-mac dst-mac ip
! Verificar que:
! - MAC origen en encabezado Ethernet coincide con MAC en ARP
! - MAC destino en encabezado Ethernet coincide con MAC en ARP
! - IP en ARP es valida (no 0.0.0.0, 255.255.255.255, etc.)
```

## Contramedida 5: Port Security

### Cisco IOS Terminal

```
SW1(config)# interface range FastEthernet0/1-3
SW1(config-if-range)# switchport port-security
SW1(config-if-range)# switchport port-security maximum 2
SW1(config-if-range)# switchport port-security violation restrict
SW1(config-if-range)# switchport port-security mac-address sticky
SW1(config-if-range)# exit
```

## Validación y Pruebas

### Verificación de DHCP Snooping

### Cisco IOS Terminal

```
SW1# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
10
DHCP snooping is operational on following VLANs:
10
Insertion of option 82 is disabled
```

Interface	Trusted	Rate limit (pps)
GigabitEthernet0/1	yes	unlimited

FastEthernet0/1	no	unlimited
FastEthernet0/2	no	unlimited
FastEthernet0/3	no	unlimited

```
SW1# show ip dhcp snooping binding
MacAddress IpAddress Lease(sec) Type VLAN Interface
-----
00:0C:29:DD:EE:FF 192.168.10.50 86400 dhcp-snooping 10 FastEthernet0/2
```

## Verificación de Dynamic ARP Inspection

### Cisco IOS Terminal

```
SW1# show ip arp inspection
Source Mac Validation : Enabled
Destination Mac Validation : Enabled
IP Address Validation : Enabled
```

Vlan	Configuration	Operation	ACL Match	Static ACL
10	Enabled	Active		

Vlan	Forwarded	Dropped	DHCP Drops	ACL Drops
10	125	47	47	0

```
SW1# show ip arp inspection interfaces
Interface Trust State Rate (pps) Burst Interval
-----
Fa0/1 Untrusted 15 1
Fa0/2 Untrusted 15 1
Fa0/3 Untrusted 15 1
Gi0/1 Trusted 0 N/A
```

## Intento de Ataque Post-Mitigación

```
$ sudo arpspoof -i eth0 -t 192.168.10.50 192.168.10.1  
0:c:29:aa:bb:cc 0:c:29:dd:ee:ff 0806 42: arp reply 192.168.10.1 is-at  
→ 0:c:29:aa:bb:cc
```

En el switch, verificar los logs:

### Cisco IOS Terminal

```
SW1# show logging  
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/1, vlan 10.  
([000c.29aa.bbcc/192.168.10.1/000c.29dd.eeff/192.168.10.50])  
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/1, vlan 10.  
([000c.29aa.bbcc/192.168.10.1/000c.29dd.eeff/192.168.10.50])
```

## Verificación en la Víctima

```
$ arp -n  
Address          HWtype  HWaddress          Flags Mask   Iface  
192.168.10.1    ether    00:1a:2b:3c:4d:01  C      eth0
```

✓ **Mitigación exitosa:** La tabla ARP de la víctima mantiene la entrada correcta del gateway. Los paquetes ARP falsos del atacante son descartados por DAI.

## Estadísticas de DAI

### Cisco IOS Terminal

```
SW1# show ip arp inspection statistics  
Vlan Forwarded Dropped DHCP Drops ACL Drops  
-----  
10 125 47 47 0
```

Vlan	DHCP Permits	ACL Permits	Probe Permits	Source MAC Failures
10	125	0	0	0
Vlan	Dest MAC Failures	IP Validation Failures	Invalid Protocol Data	
10	0	0	0	

## Problemas Encontrados y Soluciones

### Problema: DHCP Snooping Sin Bindings

**Descripción:** Despues de habilitar DHCP Snooping, la tabla de bindings estaba vacía y DAI descartaba todo el tráfico ARP.

**Diagnóstico:** Los hosts ya tenían IPs asignadas antes de habilitar DHCP Snooping, por lo que no hubo transacciones DHCP que registrar.

**Solución aplicada:** Forzar renovación DHCP en los clientes o crear bindings estáticos:

```
# En Linux
$ sudo dhclient -r && sudo dhclient eth0

# En Windows
C:\> ipconfig /release && ipconfig /renew
```

O crear entrada estática:

#### Cisco IOS Terminal

```
SW1(config)# ip source binding 000c.29dd.eeff vlan 10 192.168.10.50 interface FastEthernet0/2
```

### Problema: Puerto en Estado err-disabled

**Descripción:** Un puerto de acceso entró en estado err-disabled después de exceder el rate limit de DAI.

**Diagnóstico:** Se detectó un burst de paquetes ARP (posiblemente de un escaneo de red o el ataque).

**Solución aplicada:** Recuperar el puerto y ajustar el rate limit:

**Cisco IOS Terminal**

```
SW1# show interfaces status err-disabled
Port Name Status Reason
Fa0/1 err-disabled arp-inspection
SW1(config)# interface FastEthernet0/1
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
! Aumentar rate limit si es necesario
SW1(config-if)# ip arp inspection limit rate 30 burst interval 2
```

**Problema: Tráfico Legítimo Bloqueado por DAI**

**Descripción:** Hosts con IP estática configurada manualmente eran bloqueados por DAI.

**Diagnóstico:** DAI no encontraba entrada en la tabla de DHCP Snooping para estos hosts.

**Solución aplicada:** Crear ACL para permitir bindings estáticos:

**Cisco IOS Terminal**

```
SW1(config)# arp access-list ARP_ESTATICOS
SW1(config-arp-acl)# permit ip host 192.168.10.50 mac host 000c.29dd.eeff
SW1(config-arp-acl)# exit
SW1(config)# ip arp inspection filter ARP_ESTATICOS vlan 10
```

**Experiencia Adquirida****Conocimientos Técnicos Clave****Funcionamiento de ARP**

- Protocolo de capa 2 sin autenticación
- Broadcast para requests, unicast para replies
- Cache dinámico con timeouts configurables

**Vector de Ataque ARP Poisoning**

El ataque permite:

1. Man-in-the-Middle (MITM)

2. Denegación de servicio (DoS)
3. Secuestro de sesiones
4. Captura de credenciales

## Mecanismos de Protección

Mecanismo	Función	Efectividad
DHCP Snooping	Crear tabla de bindings IP-MAC	Alta
Dynamic ARP Inspection	Validar paquetes ARP contra bindings	Alta
Port Security	Limitar MACs por puerto	Media
ARP Estático	Entradas inmutables en hosts	Media
VLANs	Segmentación de dominios broadcast	Media
802.1X	Autenticación de puertos	Alta

## Comandos Cisco IOS Críticos

### Cisco IOS Terminal

```
! Configuracion de DHCP Snooping
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 10
Switch(config-if)# ip dhcp snooping trust
! Configuracion de DAI
Switch(config)# ip arp inspection vlan 10
Switch(config-if)# ip arp inspection trust
Switch(config-if)# ip arp inspection limit rate 15
! Validaciones adicionales
Switch(config)# ip arp inspection validate src-mac dst-mac ip
! Verificacion
Switch# show ip dhcp snooping binding
Switch# show ip arp inspection
Switch# show ip arp inspection interfaces
Switch# show ip arp inspection statistics
```

## Herramientas de Ataque/Defensa

Herramienta	Función	Plataforma
arp spoof	Envenenamiento ARP	Linux
ettercap	Suite MITM completa	Linux/Windows
Scapy	Crafting de paquetes	Python
arpwatch	Detección de cambios ARP	Linux
Wireshark	Analisis de tráfico	Multiplataforma

## Lecciones Aprendidas

### Defensa en Capas

No confiar en una sola contramedida. Implementar:

- DHCP Snooping + DAI (protección primaria)
- Port Security (capa adicional)
- Monitoreo con arpwatch (detección)
- Segmentación con VLANs (reducción de superficie)

### Prerrequisitos de DAI

DAI depende de DHCP Snooping para funcionar correctamente. Sin la tabla de bindings, DAI descartará todo el tráfico ARP de puertos untrusted.

### Impacto en Rendimiento

- Rate limiting previene DoS por floods ARP
- Demasiado restrictivo puede afectar aplicaciones legítimas
- Monitorear estadísticas regularmente

### Protocolos Vulnerables a MITM

Una vez establecido el MITM, los siguientes protocolos son especialmente vulnerables:

- HTTP (credenciales en texto claro)
- FTP (credenciales en texto claro)
- Telnet (todo en texto claro)
- SMTP sin TLS
- POP3/IMAP sin TLS

**i Recomendación:** Usar siempre protocolos cifrados (HTTPS, SFTP, SSH, TLS) como capa adicional de protección.

## Exploración de Aplicaciones y Sugerencias

### Recursos y Referencias Utilizados

#### Documentación Técnica Oficial

##### Cisco Systems

- **Cisco Catalyst Switch Security Configuration Guide** - DAI Configuration
- **Understanding and Configuring DHCP Snooping** - Cisco White Paper
- **Layer 2 Security Features** - Cisco Design Zone

##### RFCs

- **RFC 826:** “An Ethernet Address Resolution Protocol”
- **RFC 5227:** “IPv4 Address Conflict Detection”

### Herramientas Utilizadas

#### Kali Linux Tools

- **arp spoof (dsniff):** Herramienta de envenenamiento ARP
- **ettercap:** Suite completa de ataques MITM
- **Wireshark:** Análisis de capturas de red

#### Python

- **Scapy:** Librería para crafting de paquetes

### Configuraciones de Referencia

Los archivos de configuración se encuentran en el directorio configs/:

- **R1-gateway-v1.cfg:** Configuración del router gateway
- **SW1-vulnerable-v1.cfg:** Configuración inicial sin protecciones

- **SW1-secure-v1.cfg:** Configuración con DAI y DHCP Snooping
- 

**Documento:** Práctica 9.2 - ARP Poisoning

**Fecha:** Enero 27, 2026

**Autores:** Uriel Felipe Vázquez Orozco, Euler Molina Martínez

**Materia:** Redes de Computadoras 2

**Profesor:** M.C. Manuel Eduardo Sánchez Solchaga