



NETWORK SECURITY  
ELEC-H504

---

# ARP Poisoning Attack Detection and Prevention in a Software-Defined Network

---

*Students :*

MICKAEL KOVEL  
SIMON PEETROONS  
ELIA BYRNE

*Teacher :*

DRICOT JEAN-MICHEL

June 18, 2025



Abstract

The Address Resolution Protocol (ARP) is a fundamental component of IPv4 networking, responsible for resolving IP addresses into MAC addresses on local networks. Despite its critical role, ARP lacks built-in security mechanisms such as authentication and integrity checks. This structural weakness enables a well-known attack vector: ARP poisoning. By sending forged ARP messages, an attacker can manipulate ARP tables, redirect traffic, and launch man-in-the-middle (MITM) attacks or denial-of-service (DoS) operations.

This report provides a comprehensive analysis of ARP poisoning, detailing its underlying mechanisms, common tools used to execute it, and real-world attack scenarios. In addition, we evaluate detection methods and discuss multiple prevention and mitigation strategies, ranging from basic static ARP entries to more advanced solutions such as Dynamic ARP Inspection (DAI) and VLAN segmentation.

We also explore why ARP poisoning remains feasible in 2025, highlighting the challenges posed by legacy protocol designs. Finally, the report considers future alternatives, such as IPv6’s Secure Neighbor Discovery (SEND), and emphasizes the need for proactive defense, continuous monitoring, and enhanced security awareness in local networks.

Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Objectives of the Report . . . . .	3
1.2	General Context of Network Security . . . . .	3
1.3	Importance of Understanding ARP Poisoning . . . . .	3
<b>2</b>	<b>Overview of ARP (Address Resolution Protocol)</b>	<b>4</b>
2.1	How ARP Works . . . . .	4
2.2	ARP Packet Structure . . . . .	4
2.3	Role of the ARP Cache . . . . .	5
2.4	Security Limitations of ARP . . . . .	5
<b>3</b>	<b>ARP Poisoning: Definition and Mechanism</b>	<b>5</b>
3.1	What is ARP Poisoning? . . . . .	5
3.2	Attack Process: Sending Forged ARP Packets . . . . .	5
3.3	Attacker Objectives (e.g., MITM, DoS) . . . . .	6
<b>4</b>	<b>Attack Scenarios</b>	<b>6</b>
4.1	In a Local Area Network (LAN) . . . . .	6
4.2	ARP Poisoning in Public Wi-Fi Networks . . . . .	6
4.3	Example: Man-in-the-Middle and Password Capture . . . . .	6
4.4	Impact on Encrypted and Unencrypted Protocols . . . . .	7
<b>5</b>	<b>Common Attack Tools</b>	<b>7</b>
5.1	Ettercap . . . . .	7
5.2	Cain & Abel . . . . .	7
5.3	Arpspoof / Dsniff . . . . .	7
5.4	Demonstration Example (Controlled Simulation with Wireshark) . . . . .	8



<b>6</b>	<b>Security Risks and Consequences</b>	<b>8</b>
6.1	Sensitive Data Theft . . . . .	8
6.2	Traffic Redirection . . . . .	8
6.3	Service Disruption . . . . .	9
<b>7</b>	<b>Detection of ARP Poisoning</b>	<b>9</b>
7.1	Symptoms on Victim Machines . . . . .	9
7.2	Detection Tools: arpswatch, Wireshark . . . . .	9
7.3	Manual Detection Methods (Inconsistent ARP Cache, etc.) . . . . .	9
<b>8</b>	<b>Prevention and Mitigation Measures</b>	<b>10</b>
8.1	Static ARP Entries: Benefits and Limitations . . . . .	10
8.2	ARP Packet Filtering (Firewalls, ACLs) . . . . .	10
8.3	Network Segmentation (VLANs) . . . . .	10
8.4	Port Security on Switches . . . . .	10
8.5	DHCP Snooping and Dynamic ARP Inspection (DAI) . . . . .	11
8.6	Continuous Network Monitoring . . . . .	11
<b>9</b>	<b>Case Study / Practical Simulation</b>	<b>11</b>
9.1	Lab or Virtual Environment Setup . . . . .	11
9.2	Executing the Attack . . . . .	12
9.3	Observation and Analysis . . . . .	12
9.4	Applying Countermeasures . . . . .	13
<b>10</b>	<b>Limitations and Perspectives</b>	<b>13</b>
10.1	Why ARP Poisoning Is Still Possible in 2025 . . . . .	13
10.2	Proposed Improvements to ARP . . . . .	13
10.3	Secure Alternatives (e.g., IPv6 ND and SEND) . . . . .	14
<b>11</b>	<b>Conclusion</b>	<b>14</b>
11.1	Summary of Risks and Best Practices . . . . .	14
11.2	Need for Awareness and Detection Tools . . . . .	14
11.3	Outlook on Local Network Security . . . . .	15
	<b>References</b>	<b>16</b>
	<b>Annexes</b>	<b>16</b>

# 1 Introduction

## 1.1 Objectives of the Report

This report aims to provide a comprehensive technical analysis of ARP poisoning, a well-known attack targeting the Address Resolution Protocol at the data link layer of computer networks. The objectives are:

- To explain the underlying mechanisms of ARP and how it is exploited
- To illustrate real-world attack scenarios and their consequences
- To present and evaluate detection and prevention techniques
- To assess the limitations of current solutions and explore future alternatives

This analysis is relevant to students, network administrators, and security professionals who seek to better understand and defend against Layer 2 threats.

## 1.2 General Context of Network Security

Network security has traditionally focused on perimeter defense, firewalls, and encryption at higher layers of the OSI model. However, attacks targeting the lower layers — particularly Layer 2 (Data Link Layer) — remain a persistent threat.

The Address Resolution Protocol (ARP), designed in the early 1980s, plays a critical role in IPv4 networking by mapping IP addresses to physical MAC addresses. It was developed at a time when security was not a priority, and therefore lacks essential protections such as authentication and integrity verification.

This vulnerability is particularly relevant in local area networks (LANs), where attackers with internal access can exploit ARP to conduct man-in-the-middle (MITM) attacks, intercept sensitive data, or cause service disruptions.

## 1.3 Importance of Understanding ARP Poisoning

ARP poisoning exemplifies the risks of legacy protocols operating in modern networks. Although widely known, the attack remains effective in 2025 due to:

- The simplicity and reliability of the attack
- The lack of built-in defenses in standard ARP implementations
- The widespread use of unsecured or misconfigured LAN environments

Understanding ARP poisoning is essential for anyone responsible for network defense. Not only does it demonstrate the need for better visibility and monitoring at Layer 2, but it also highlights the broader challenge of securing protocols that were never designed with malicious actors in mind.

This report addresses these concerns by exploring the ARP protocol, its vulnerabilities, exploitation techniques, and the available mitigation strategies.

## 2 Overview of ARP (Address Resolution Protocol)

### 2.1 How ARP Works

The Address Resolution Protocol (ARP) is a network protocol used to map IP addresses (Layer 3) to MAC addresses (Layer 2) in a local area network (LAN). When a device wants to communicate with another device on the same network but only knows its IP address, it broadcasts an ARP request asking "Who has this IP address?" The device with the corresponding IP replies with its MAC address, which the requester then uses to send the packet.

### 2.2 ARP Packet Structure

An ARP packet is encapsulated in an Ethernet frame and contains several fields:

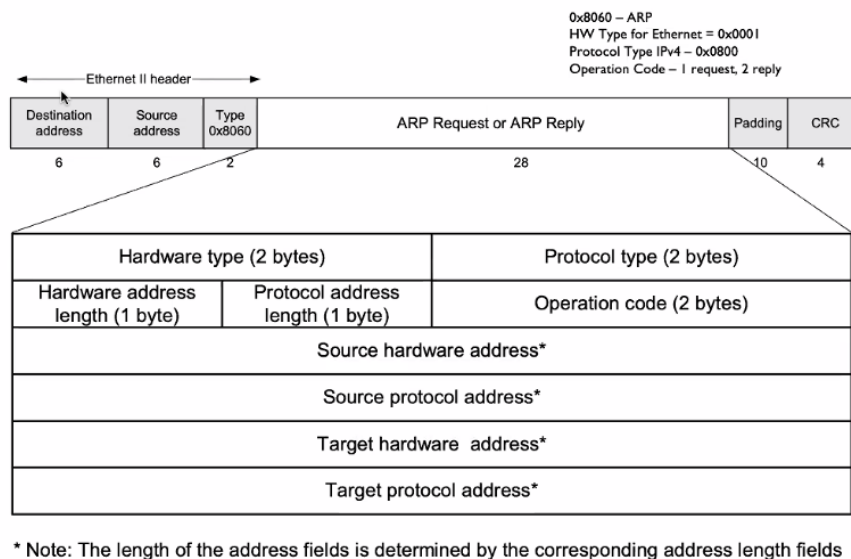


Figure 1: ARP Packet including Ethernet header, padding and CRC [1]

- **Hardware type (HTYPE):** Specifies the type of hardware (usually Ethernet, value 1)
- **Protocol type (PTYPE):** Specifies the protocol (usually IPv4, value 0x0800)
- **Hardware size (HLEN) and Protocol size (PLEN):** Lengths of MAC and IP addresses
- **Opcode:** Indicates request (1) or reply (2)
- **Sender MAC and IP address**
- **Target MAC and IP address**

This structure allows ARP to resolve addresses efficiently but is also inherently vulnerable due to the lack of authentication

## 2.3 Role of the ARP Cache

Each device maintains an ARP cache: a table mapping IP addresses to MAC addresses. This cache reduces network traffic by avoiding repeated ARP requests for the same address. Entries are either dynamically learned through ARP exchanges or statically configured by the system administrator. However, dynamic entries can be overwritten by malicious actors, which is the basis of ARP poisoning attacks.

## 2.4 Security Limitations of ARP

ARP was designed in the early days of networking, with little consideration for malicious behavior. It lacks:

- **Authentication:** Any host can send an ARP reply, even unsolicited
- **Integrity checks:** No verification that the reply is genuine
- **Protection mechanisms:** Broadcast-based by nature, exposing all requests and replies

These limitations make it possible for attackers to perform ARP spoofing and poisoning, enabling attacks such as Man-in-the-Middle and denial-of-service.

# 3 ARP Poisoning: Definition and Mechanism

## 3.1 What is ARP Poisoning?

ARP Poisoning, also known as ARP Spoofing, is a technique by which an attacker sends falsified ARP messages over a local area network (LAN) in order to associate their MAC address with the IP address of another host. This misleads other devices in the network into sending their traffic to the attacker instead of the legitimate recipient. The attack exploits the stateless nature of ARP, which accepts replies without authentication or prior request.

## 3.2 Attack Process: Sending Forged ARP Packets

The attacker typically performs the following steps:

1. Scans the network to identify IP and MAC address pairs.
2. Crafts and sends ARP reply packets falsely mapping their own MAC address to the IP of a legitimate device (e.g., the gateway).
3. Once the ARP cache of the target(s) is poisoned, traffic meant for the legitimate device is redirected to the attacker.

These packets can be sent periodically to maintain the poisoned state, as most systems will eventually refresh their ARP entries. The attack can be bidirectional (spoofing both victim and gateway) to fully intercept communications.

### 3.3 Attacker Objectives (e.g., MITM, DoS)

ARP Poisoning can enable multiple types of attacks:

- **Man-in-the-Middle (MITM):** Intercept, modify, or record traffic between two parties without their knowledge.
- **Denial of Service (DoS):** By redirecting packets to a non-existent MAC address or black hole, the attacker can disrupt communication.
- **Session hijacking, credential theft, traffic injection:** Once in the communication flow, attackers can exploit unsecured protocols to steal or manipulate data.

This makes ARP Poisoning a versatile and dangerous attack vector in unprotected networks [2].

## 4 Attack Scenarios

### 4.1 In a Local Area Network (LAN)

ARP poisoning is most effective in environments where devices are on the same Layer 2 segment, such as a switched Ethernet LAN. Since ARP relies on broadcast messages within the subnet, an attacker connected to the same switch as the target can intercept ARP requests and replies. In a LAN, once the attacker poisons the ARP cache of hosts or the default gateway, they can redirect traffic through their own machine.

In switched environments, despite the segmentation of traffic, ARP poisoning works because switches forward ARP broadcasts to all ports, and spoofed replies are accepted without verification.

### 4.2 ARP Poisoning in Public Wi-Fi Networks

Public wireless networks are particularly vulnerable to ARP poisoning due to their open nature and lack of physical boundaries. Attackers can easily join the same subnet and perform spoofing without needing administrative privileges on the access point.

These networks often lack network isolation (e.g., client isolation or private VLANs), making it trivial for an attacker to launch a man-in-the-middle attack on nearby users. As devices automatically connect to familiar open SSIDs, the risk of ARP poisoning in cafés, airports, and hotels is significantly high.

### 4.3 Example: Man-in-the-Middle and Password Capture

Once ARP poisoning is successful, attackers can insert themselves between a victim and their intended destination, capturing all traffic in real time. Tools like **Ettercap** or **dsniff** facilitate this process by automatically poisoning targets and sniffing credentials from unencrypted protocols like HTTP, FTP, SMTP, or POP3.

In a common scenario, a victim connects to a banking site over HTTP (instead of HTTPS), and the attacker captures login credentials. Even with HTTPS, attackers may attempt SSL stripping attacks by downgrading the connection and presenting a forged certificate if victims are not vigilant.



## 4.4 Impact on Encrypted and Unencrypted Protocols

**Unencrypted protocols** are highly vulnerable to ARP poisoning because sensitive data (e.g., passwords, session cookies) is transmitted in cleartext. These include protocols like HTTP, Telnet, POP3, and FTP.

**Encrypted protocols** such as HTTPS, SSH, and TLS-based services are more resistant to interception. However, attackers can still block or delay packets to cause connection resets (DoS). Even if the content is encrypted, some information like the domain name (via SNI and TLS handshake), IP addresses, and packet sizes or timing can still be observed, which may reveal which service is being accessed or when.

Encryption alone is not always sufficient without additional protections like certificate pinning or DNSSEC.

# 5 Common Attack Tools

## 5.1 Ettercap

Ettercap is one of the most widely used tools for conducting ARP poisoning and man-in-the-middle attacks on local networks. It supports both active and passive sniffing and can intercept traffic in switched networks by performing ARP spoofing. Ettercap offers:

- Real-time traffic interception and filtering
- Plugin support for extending attacks (e.g., DNS spoofing, SSL stripping)
- Compatibility with GUI and CLI modes

Ettercap automates the poisoning process and visualizes intercepted data in a user-friendly interface, making it especially effective in educational and demonstration scenarios [3].

## 5.2 Cain & Abel

Cain & Abel is a multifunctional tool available on Windows platforms, primarily known for password recovery, but it also includes features for ARP poisoning and packet sniffing. It is capable of:

- Performing MITM attacks through ARP cache poisoning
- Capturing credentials transmitted over insecure protocols
- Cracking encrypted passwords using dictionary and brute-force attacks

Although development has stopped, Cain & Abel remains popular in penetration testing and educational labs due to its wide range of functionalities [4].

## 5.3 Arpspoof / Dsniff

The **dsniff** suite, developed by Dug Song, includes a lightweight ARP spoofing tool called **arpspoof**. Unlike Ettercap, **arpspoof** focuses solely on injecting forged ARP replies without additional features. It is typically used in combination with other sniffing tools like **tcpdump** or **Wireshark**. **dsniff** also includes utilities for intercepting passwords from various protocols:



- `dsniff`: Password sniffer
- `macof`: Switch flooding tool
- `filesnarf`, `mailsnarf`: Data extraction tools

These command-line tools are efficient, scriptable, and suitable for controlled network experiments [5], [6].

## 5.4 Demonstration Example (Controlled Simulation with Wireshark)

In a controlled lab environment, ARP poisoning can be simulated safely using virtual machines or isolated physical hosts. A typical setup includes:

- One attacker machine (e.g., Kali Linux)
- Two victim machines (client and gateway)
- Tools: `arpspoof` or `ettercap`, combined with Wireshark for packet capture

The attacker uses `arpspoof` to redirect traffic between the client and the gateway. With Wireshark, the ARP replies and captured data can be observed and analyzed in real time. This allows students to explore the mechanics of the attack and understand the security risks it introduces

# 6 Security Risks and Consequences

## 6.1 Sensitive Data Theft

One of the primary risks of ARP poisoning is the theft of sensitive information. By performing a man-in-the-middle attack, the attacker gains the ability to intercept and inspect packets transmitted between victims. If protocols such as HTTP, FTP, or Telnet are used, credentials and other personal information are often sent in plaintext and can be easily extracted. Even in encrypted environments, metadata such as visited domains and connection times can be valuable for surveillance or profiling.

## 6.2 Traffic Redirection

ARP poisoning allows attackers to reroute traffic to malicious servers. This technique can be used to:

- Redirect users to phishing sites that mimic legitimate services
- Intercept DNS queries and provide falsified responses (DNS spoofing)
- Disrupt VPN tunnels by dropping or rerouting encrypted traffic

This redirection is particularly dangerous in corporate environments where internal services are trusted based on IP address

## 6.3 Service Disruption

By poisoning ARP caches and redirecting packets to invalid or non-existent MAC addresses, attackers can effectively cause denial-of-service (DoS) conditions. Victims may lose connectivity to critical systems such as gateways or DNS servers. Furthermore, frequent spoofed updates can overload ARP tables on older devices or cause instability in poorly configured switches

# 7 Detection of ARP Poisoning

## 7.1 Symptoms on Victim Machines

Victims of ARP poisoning may observe unusual network behavior, such as:

- Sudden loss of connectivity or intermittent access to certain services
- Duplicate IP address warnings issued by the operating system
- Slower-than-normal network performance due to traffic rerouting
- Unexpected certificate errors on HTTPS sites due to MITM interception

These symptoms can indicate that a device's ARP cache has been compromised

## 7.2 Detection Tools: arpswatch, Wireshark

Several tools are available to automate the detection of ARP poisoning:

**Arpswatch** monitors ARP activity and logs IP-to-MAC mappings. It sends alerts when a MAC address changes for a known IP, which is a strong indicator of spoofing [7].

**Wireshark** allows manual inspection of ARP packets. A typical sign of poisoning is the presence of unsolicited ARP replies or inconsistent mappings between the same IP and multiple MAC addresses.

**tcpdump** is a lightweight command-line tool that can be used to capture and inspect ARP traffic in real time. It is especially useful in constrained or headless environments. Anomalies such as repeated unsolicited ARP replies, a high frequency of ARP packets, or conflicting MAC addresses for the same IP can indicate an ongoing ARP poisoning attempt.

## 7.3 Manual Detection Methods (Inconsistent ARP Cache, etc.)

Even without dedicated tools, ARP poisoning can be detected through:

- **Cache inspection:** On Linux, `ip neigh` or `arp -a` commands show ARP table contents. Entries that change unexpectedly or show duplicate MACs should be investigated.
- **Packet analysis:** Monitoring ARP replies without corresponding requests, or multiple devices claiming the same IP.
- **Ping sweep and correlation:** Manually verifying if devices respond from the MAC address expected for their IP.

These techniques are useful in environments where automated tools are not available or when deeper analysis is needed

## 8 Prevention and Mitigation Measures

### 8.1 Static ARP Entries: Benefits and Limitations

One of the most basic protections against ARP poisoning is to statically configure ARP entries on critical devices. This ensures that the MAC address associated with a particular IP cannot be altered by spoofed ARP replies.

**Benefits:**

- Prevents dynamic updates from attackers
- Simple to implement on a small scale

**Limitations:**

- Not scalable for large networks
- Requires manual maintenance and updates
- Vulnerable if an attacker gains access to the host and modifies entries

Static ARP tables are therefore better suited for isolated or high-value systems rather than general use

### 8.2 ARP Packet Filtering (Firewalls, ACLs)

Network-based firewalls and intrusion prevention systems can be configured to block unsolicited ARP replies or validate them against known mappings. Access control lists (ACLs) on switches and routers can enforce restrictions based on MAC/IP pairings.

Modern endpoint firewalls may also support rules to monitor and log changes in ARP tables, detecting anomalies locally. However, this relies on up-to-date configurations and active monitoring

### 8.3 Network Segmentation (VLANs)

Implementing VLANs (Virtual LANs) helps isolate traffic at the Layer 2 level. Since ARP poisoning attacks require access to the same broadcast domain as the victim, segmenting the network limits the attack surface.

VLANs are particularly effective in environments such as campus networks or corporate offices where users can be grouped by department or privilege level. This approach reduces the number of devices exposed to a potential attacker

### 8.4 Port Security on Switches

Port security features on managed switches allow administrators to define specific MAC addresses allowed per port. If a device attempts to spoof another MAC address or exceeds the configured limit, the port can be automatically shut down or restricted.

This mechanism effectively prevents attackers from impersonating other hosts on the network, particularly in environments like university labs or shared offices

## 8.5 DHCP Snooping and Dynamic ARP Inspection (DAI)

**DHCP Snooping** creates a binding table of legitimate IP-MAC pairs based on DHCP traffic. **Dynamic ARP Inspection (DAI)** then uses this table to validate ARP packets on the network.

DAI drops ARP packets with invalid bindings and logs the incident. It is one of the most effective Layer 2 security features available on enterprise switches and is particularly useful in dynamic environments such as Wi-Fi networks

## 8.6 Continuous Network Monitoring

Regular monitoring of ARP traffic can help detect anomalies in real time. Network management systems (NMS) or intrusion detection systems (IDS) can:

- Alert on duplicate IP addresses or frequent MAC changes
- Correlate ARP activity with baseline network behavior
- Record ARP logs for forensic analysis

Proactive monitoring, combined with baseline configuration and anomaly detection, is essential to ensure long-term protection against ARP poisoning

# 9 Case Study / Practical Simulation

## 9.1 Lab or Virtual Environment Setup

The topology used to emulate the attack is composed by a router, running debian-netsec provided in the gns3 virtual machine used for the course. This decision comes from the fact that the router runs arp spoofing detection script leveraging tcpdump and ip tables as well as ebtables in order to block both layer3 and layer2 traffic.

A switch is connected to the router, and 3 machines running debian-netsec (2 workstations and 1 attacker) are connected to the switch.

The NAT is used to provide internet connection to the topology, that was needed to update the distro and to install needed tools i.e. ettercap.

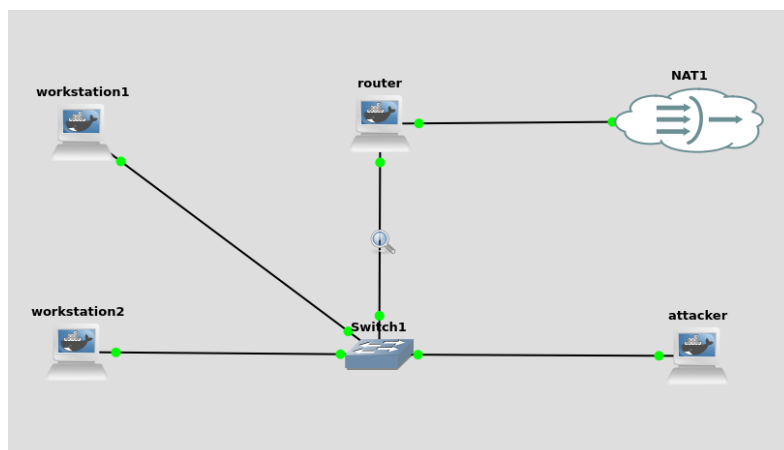


Figure 2: GNS3 topology

## 9.2 Executing the Attack

The attack is run by the attacker machine using `ettercap`, performing a MITM on a specific victim - here a workstation (192.168.1.10).

```
ettercap -T -M arp:remote /192.168.1.10// /192.168.1.1//
```

The attacker is thus able to intercept the traffic generated from the victim workstation machine and directed to the gateway (192.168.1.1) (the debian router).

## 9.3 Observation and Analysis

To confirm that the ARP poisoning attack was effective, network traffic was monitored on the attacker's host using `tcpdump`, while the victim's host attempted to reach the Internet.

The following command was executed on attacker's host to capture all IP packets:

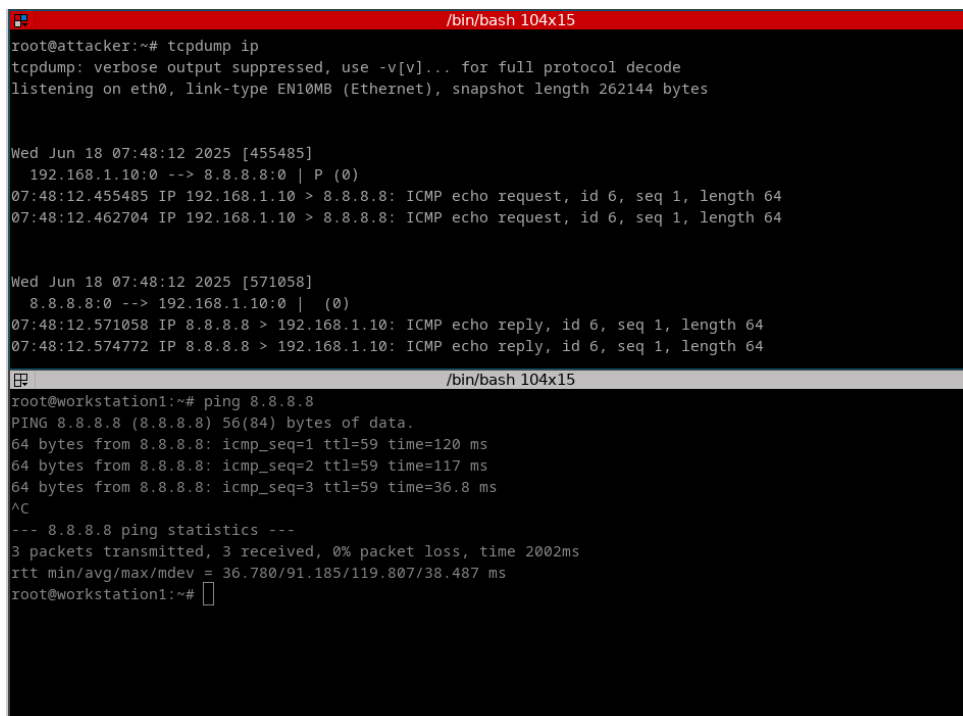
```
tcpdump ip
```

Simultaneously, on victim's host, we triggered outbound traffic by running a simple ping to Internet :

```
ping google.com or  
ping 8.8.8.8
```

When the ARP spoofing attack was active, `tcpdump` on attacker's host displayed IP packets originating from victim's host and destined for `google.com`. This indicates that the attacker successfully inserted itself in the communication path between the victim and the gateway, achieving a man-in-the-middle (MITM) position.

The appearance of redirected traffic on attacker's host shows that the ARP tables of both the victim and the router have been poisoned correctly.



```
/bin/bash 104x15
root@attacker:~# tcpdump ip
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes

Wed Jun 18 07:48:12 2025 [455485]
 192.168.1.10:0 --> 8.8.8.8:0 | P (0)
07:48:12.455485 IP 192.168.1.10 > 8.8.8.8: ICMP echo request, id 6, seq 1, length 64
07:48:12.462704 IP 192.168.1.10 > 8.8.8.8: ICMP echo request, id 6, seq 1, length 64

Wed Jun 18 07:48:12 2025 [571058]
 8.8.8.8:0 --> 192.168.1.10:0 | (0)
07:48:12.571058 IP 8.8.8.8 > 192.168.1.10: ICMP echo reply, id 6, seq 1, length 64
07:48:12.574772 IP 8.8.8.8 > 192.168.1.10: ICMP echo reply, id 6, seq 1, length 64

/bin/bash 104x15
root@workstation1:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=59 time=120 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=59 time=117 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=59 time=36.8 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 36.780/91.185/119.807/38.487 ms
root@workstation1:~#
```

Figure 3: Successful MITM (attacker)

## 9.4 Applying Countermeasures

Once the spoofing detection script is run, tcpdump captures the traffic and looks for ARP Replies claiming to be the router IP (192.168.1.1), but with any MAC other than the trusted router MAC. Subsequently the attacker's mac is blocked by iptables and ebtables rules. See Listing 1

# 10 Limitations and Perspectives

## 10.1 Why ARP Poisoning Is Still Possible in 2025

Despite being a well-documented vulnerability for over two decades, ARP poisoning remains viable in 2025 due to several enduring issues:

- **Lack of Authentication:** The ARP protocol was designed without any mechanism to verify the authenticity of requests or replies.
- **Backward Compatibility:** Modern networks still rely on legacy IPv4 infrastructure where ARP is fundamental.
- **Limited Adoption of Advanced Protections:** Features like Dynamic ARP Inspection (DAI) and port security are not enabled by default and require administrative expertise.
- **Ease of Execution:** Tools for ARP spoofing are freely available and simple to use, making this attack accessible even to non-expert attackers.

In many organizations, these factors combine to leave internal networks vulnerable despite external protections such as firewalls.

## 10.2 Proposed Improvements to ARP

Over the years, researchers have proposed several enhancements to secure ARP:

- **Secure ARP (S-ARP):** Introduces digital signatures to ARP messages, ensuring authenticity and integrity [8].
- **Ticket-based ARP:** Uses cryptographic tickets from a trusted authority to validate MAC-IP bindings [9].
- **Active probing techniques:** Devices verify mappings via challenge-response mechanisms before updating ARP caches.

While these solutions improve security, most require significant protocol or infrastructure changes, limiting their deployment in existing networks.

### 10.3 Secure Alternatives (e.g., IPv6 ND and SEND)

IPv6 replaces ARP with the Neighbor Discovery Protocol (NDP), which handles similar address resolution functions but over ICMPv6. NDP suffers from similar vulnerabilities to ARP, such as spoofing attacks.

To address this, IPv6 includes a cryptographic extension called **Secure Neighbor Discovery (SEND)**:

- SEND uses **Cryptographically Generated Addresses (CGA)** and **RSA signatures** to verify the source of neighbor advertisements.
- It protects against address spoofing, replay attacks, and router impersonation.

However, SEND is not widely deployed due to its complexity, lack of support on legacy devices, and operational overhead. Nonetheless, it represents the most secure alternative to ARP currently available and reflects the direction of future secure network protocols[10].

## 11 Conclusion

### 11.1 Summary of Risks and Best Practices

ARP poisoning remains a significant threat in modern IPv4 networks due to the inherent lack of authentication in the ARP protocol. This attack enables adversaries to intercept, modify, or block traffic, leading to serious security breaches such as credential theft, data leakage, and service disruption.

Several best practices can mitigate these risks:

- Deploy VLANs and segment networks to limit broadcast domains.
- Use switch-level protections such as port security and DAI.
- Configure static ARP entries for critical systems.
- Monitor ARP traffic regularly using tools like arpswatch or Wireshark.

### 11.2 Need for Awareness and Detection Tools

While technological countermeasures exist, their effectiveness depends on proper deployment and operational awareness. Organizations must:

- Train network administrators and users to recognize symptoms of ARP-based attacks.
- Integrate detection tools into standard network monitoring workflows.
- Perform periodic audits of ARP caches and MAC/IP mappings.

Proactive detection remains essential, as many networks still run without ARP-specific defenses.



### 11.3 Outlook on Local Network Security

Securing Layer 2 communication is an ongoing challenge. As networks evolve and migrate to IPv6, ARP will eventually be phased out in favor of Neighbor Discovery Protocol. However, unless features like Secure Neighbor Discovery (SEND) are widely adopted, similar vulnerabilities may persist.

In the short term, combining robust network configurations with vigilant monitoring offers the most practical defense. Long-term improvements will require broader adoption of cryptographic protections and built-in protocol hardening across the networking stack.

## Annexes

### ARP Spoofing Detection Script

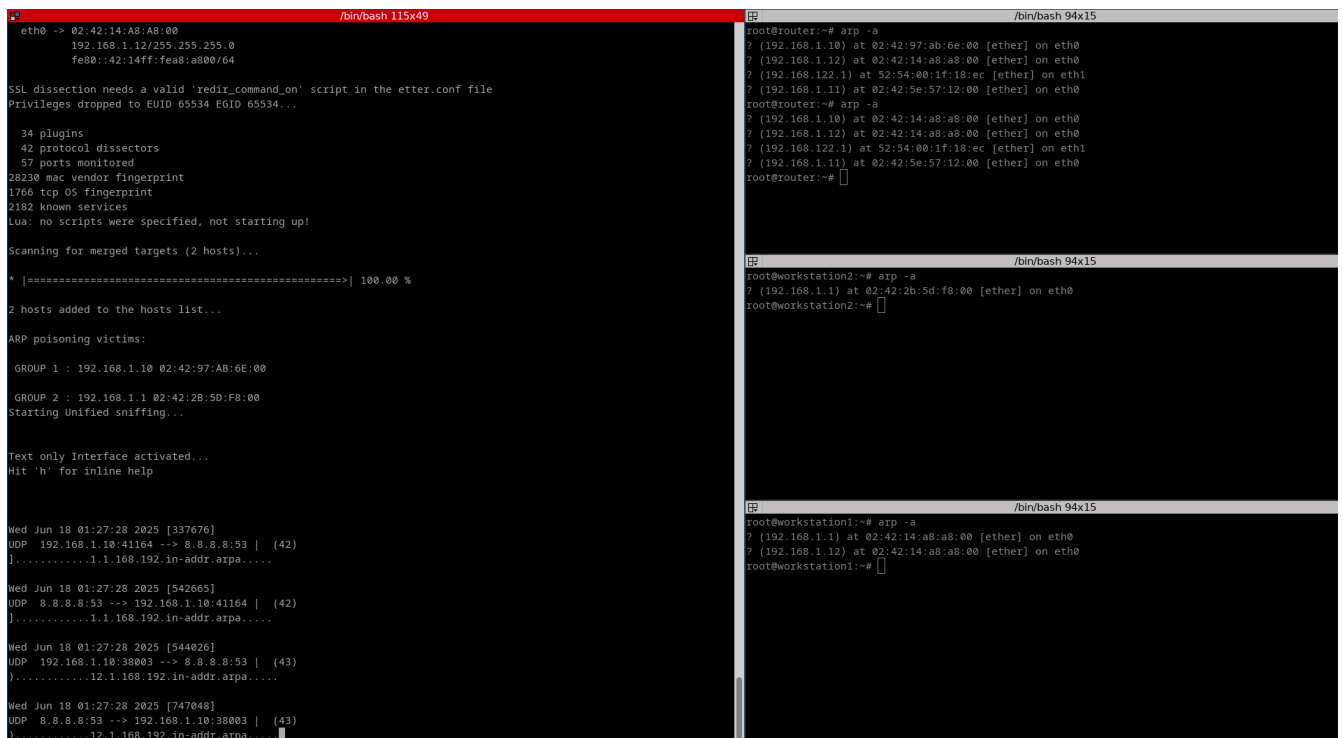
```
1  #!/bin/bash
2
3  IFACE="eth0"
4  LOG="/var/log/arp_spoof_detect.log"
5  BLOCKED="/tmp/blocked_macs.txt"
6
7  # Set router IP + router MAC manually here:
8  ROUTER_IP="192.168.1.1"
9  ROUTER_MAC="aa:bb:cc:dd:ee:ff"
10
11 # Init blocked list
12 touch $BLOCKED
13
14 echo "[*] Starting tcpdump ARP monitor on $IFACE..."
15 echo "[*] Logging to $LOG"
16
17 tcpdump -l -i $IFACE arp | while read line; do
18     echo "$line" >>$LOG
19
20     if echo "$line" | grep -q "Reply"; then
21
22         if echo "$line" | grep "$ROUTER_IP" | grep -v "$ROUTER_MAC";
23             then
24                 echo "[!] Possible ARP spoof detected! $line"
25
26                 ATTACKER_MAC=$(echo "$line" | grep -oE '([0-9a-f]{2}:){5}[0-9a-f]{2}')
27
28                 # Already blocked? Skip.
29                 if grep -q "$ATTACKER_MAC" $BLOCKED; then
30                     echo "[*] Already blocked $ATTACKER_MAC      skipping"
31                     continue
32                 fi
33
34                 echo "[*] Blocking new attacker MAC $ATTACKER_MAC"
35
36                 iptables -I INPUT -m mac --mac-source "$ATTACKER_MAC" -j DROP
37                 iptables -I OUTPUT -m mac --mac-source "$ATTACKER_MAC" -j
38                     DROP
39                 iptables -I FORWARD -m mac --mac-source "$ATTACKER_MAC" -j
40                     DROP
41
42                 if command -v ebtables >/dev/null 2>&1; then
43                     echo "[*] ebtables found      adding L2 block for
44                         $ATTACKER_MAC"
45                     ebtables -A INPUT -s "$ATTACKER_MAC" -j DROP
```

```

42     ebtables -A OUTPUT -s "$ATTACKER_MAC" -j DROP
43     ebtables -A FORWARD -s "$ATTACKER_MAC" -j DROP
44     else
45         echo "[*] ebtables not installed      skipping L2 block"
46     fi
47
48     echo "$ATTACKER_MAC" >>$BLOCKED
49
50     echo "Block rule inserted for $ATTACKER_MAC"
51 fi
52 fi
53 done

```

Listing 1: Python ARP spoofing attack script



```

eth0 -> 02:42:14:A8:A8:00
192.168.1.12/255.255.255.0
fe80::42:14ff:fea8:a800/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Privileges dropped to EUID 65534 EGID 65534...

34 plugins
42 protocol dissectors
57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

* |=====| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

GROUP 1 : 192.168.1.10 02:42:97:A8:6E:00
GROUP 2 : 192.168.1.1 02:42:2B:5D:F8:00
Starting Unified sniffing...

Text only Interface activated...
Hit 'h' for inline help

Wed Jun 18 01:27:28 2025 [337076]
UDP 192.168.1.10:41164 -> 8.8.8.8:53 | (42)
).....1.1.168.192.in-addr.arpa.....

Wed Jun 18 01:27:28 2025 [542665]
UDP 8.8.8.8:53 -> 192.168.1.10:41164 | (42)
).....1.1.168.192.in-addr.arpa.....

Wed Jun 18 01:27:28 2025 [544026]
UDP 192.168.1.10:38003 -> 8.8.8.8:53 | (43)
).....12.1.168.192.in-addr.arpa.....

Wed Jun 18 01:27:28 2025 [747048]
UDP 8.8.8.8:53 -> 192.168.1.10:38003 | (43)
).....12.1.168.192.in-addr.arpa.....

```

Figure 4: ARP Spoof



Figure 5: Curl Intercept

Figure 6: WiresharkFigure 7: Mitigation Wireshark

## References

- [1] Dr. Anand Bhojan. Cs3103 lecture 2: Arp, dhcp, vlan, 2020. Accessed: May 2025 <https://deunitato.github.io/NUSCSMODS/2020-08-18-cs3103-lecture-2-arp-dhcp-vlan/>.
- [2] Ramin Sadre. Computer system security. Lecture slides, TP, Université catholique de Louvain, 2025.
- [3] Ettercap Project. Ettercap: A suite for man-in-the-middle attacks [source code], 2025. Accessed: June 2025 <https://github.com/Ettercap/ettercap>.
- [4] xchwarze. Cain & abel: Password recovery tool for microsoft operating systems [source code], 2025. Accessed: June 2025 <https://github.com/xchwarze/Cain>.
- [5] mauricelambert. Arpspoof: A simple arp spoofing tool in python [source code], 2025. Accessed: May 2025 <https://github.com/mauricelambert/ArpSpoof>.
- [6] tecknicaltom. dsniff: A collection of tools for network auditing and penetration testing [source code], 2025. Accessed: June 2025 <https://github.com/tecknicaltom/dsniff>.
- [7] Kali Linux. arpwatc: A tool to monitor ethernet macip pairings [tool documentation], 2025. Accessed: May 2025 <https://www.kali.org/tools/arpwatch/>.
- [8] Biju Issac. Secure arp and secure dhcp protocols to mitigate security attacks. *International Journal of Network Security*, 8(2):107–118, 2009.
- [9] Wesam Lootah, William Enck, and Patrick McDaniel. Tarp: Ticket-based address resolution protocol. In *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC)*, pages 108–116, 2005.
- [10] Scott Hogg. Holding IPv6 neighbor discovery to a higher standard of security, 2015. Accessed: June 2025 <https://blogs.infoblox.com/ipv6-coe/holding-ipv6-neighbor-discovery-to-a-higher-standard-of-security/>.