

Incident Response Report: ARP Poisoning Attack

Mohammed Firdaws Alnuur

Future Interns

Nairobi, Kenya

Feb 2025

Declaration and approval

I declare that this report is my original work and has not been previously submitted for approval.
To the best of my knowledge, all sources have been cited appropriately

Student Name: Mohammed Firdaws Alnuur

Email: firdawsalnuur4@gmail.com

Abstract

This report examines an Address Resolution Protocol (ARP) poisoning attack conducted as part of a simulated cybersecurity incident response exercise. The objective was to analyze the attack methodology, assess its impact, and develop mitigation strategies. The attack exploited ARP vulnerabilities using Ettercap to manipulate ARP tables, allowing unauthorized interception of network traffic and credential theft. The report details the attack chain, indicators of compromise, and countermeasures, including network monitoring, ARP cache management, and security protocol enforcement. By implementing best practices such as Dynamic ARP Inspection and encrypted communication, organizations can enhance network security and prevent similar attacks.

Chapter 1: Introduction

1.1 Executive Summary

This report analyzes an Address Resolution Protocol (ARP) poisoning attack conducted as part of a simulated cybersecurity incident response exercise. The attack exploited the lack of authentication in ARP, allowing an attacker to intercept network traffic and capture sensitive data. The report provides a detailed overview of the attack, its impact, and the mitigation steps taken to prevent recurrence.

1.2 Incident Overview

Incident Type:

ARP Poisoning Attack

Initial Access Vector:

Exploitation of ARP protocol vulnerabilities using Ettercap to manipulate ARP tables and intercept traffic.

1.3 Objectives

- 1 Simulate an ARP poisoning attack
- 2 Capture network traffic and sensitive credentials
- 3 Implement mitigation strategies

1.4 Attack Chain Analysis

1. Reconnaissance:

The attack occurred due to the vulnerability of the Address Resolution Protocol (ARP), which lacks authentication. This allowed an attacker to:

- Use Ettercap to scan the network and identify hosts.
- Perform ARP poisoning, tricking devices into associating the attacker's MAC address with a legitimate IP address (., 192.168.100.1).
- Capture sensitive data, such as login credentials, using Wireshark.

2. Initial Exploitation:

- Performed ARP poisoning by sending spoofed ARP replies, associating the attacker's MAC address with a legitimate IP address (192.168.100.1).

3. Persistence:

- Continued to manipulate ARP tables to sustain network interception.

4. Data Exfiltration:

- Captured network traffic using Wireshark.
- Extracted login credentials from intercepted packets.

5. Analysis and Response:

- Detected anomalies in ARP tables.
- Identified changes in MAC addresses.

1.5 Indicators of Compromise (IOCs)

Type	Details
Malicious MAC Address	08-00-27-38-c8-03
Targeted IP Address	192.168.100.1
Attack Tool Used	Ettercap
Captured Data	Login credentials via Wireshark

Chapter 2. Impact Analysis

2.1: Affected Systems:

- Devices connected to the local network were vulnerable to traffic interception.

Potential Damage:

- Unauthorized access to sensitive credentials.
- Possible session hijacking.
- Man-in-the-Middle (MitM) attack execution.

2.2: Mitigation Steps Taken

1. Detection:

Identify the malicious MAC address:

- Use arp -a (Windows) or arp -n (Linux) to check the ARP table for anomalies.
- Compare against known device MAC addresses.

1. Containment and Eradication:

- Cleared ARP cache:
 - Windows: netsh interface ip delete arpcache
 - Linux/macOS: sudo ip -s -s neigh flush all
- Restarted affected devices to refresh ARP tables.

2. Recovery:

- Implemented static ARP entries for critical network components.
- Deployed ARP monitoring tools (Arpwatch, XArp) to detect future spoofing attempts.

2.3: Prevention Recommendations

1. Network Security Enhancements:

- Implement Dynamic ARP Inspection (DAI) on managed switches.
- Enable port security to restrict allowed MAC addresses.
- Use encrypted communication protocols (HTTPS, SSH, VPN) to mitigate data interception risks.

2. Monitoring and Response:

- Deploy intrusion detection systems (IDS) to monitor ARP spoofing activities.
- Conduct periodic network traffic analysis with Wireshark or similar tools.

2.4: Recommendations to Prevent Future Attacks

- Implement Dynamic ARP Inspection (DAI) on managed switches to verify ARP packets.
- Use static ARP entries for essential systems to prevent ARP spoofing.
- Enable port security on network switches to limit allowed MAC addresses.
- Implement network segmentation to isolate critical systems.
- Use encrypted protocols (HTTPS, SSH, VPN) to protect data from interception.
- Monitor network traffic regularly using tools like Wireshark or IDS/IPS solutions.

Conclusion

The ARP poisoning attack successfully demonstrated how adversaries can exploit unsecured network protocols to intercept sensitive data. By implementing network security measures such as Dynamic ARP Inspection, static ARP entries, and encrypted communication, organizations can significantly reduce the risk of ARP-based attacks. Regular monitoring and awareness training are essential in mitigating future threats.

Chapter 3 : Appendices

Wireless LAN adapter WiFi:

```
Connection-specific DNS Suffix . :  
Link-local IPv6 Address . . . . . : fe80::fd9f:72e7:a3ed:43dd%12  
IPv4 Address. . . . . : 192.168.100.8  
Subnet Mask . . . . . : 255.255.255.0  
Default Gateway . . . . . : 192.168.100.1
```

Ethernet adapter Bluetooth Network Connection:

```
Media State . . . . . : Media disconnected  
Connection-specific DNS Suffix . :
```

Ethernet adapter vEthernet (WSL (Hyper-V firewall)):

```
Connection-specific DNS Suffix . :  
Link-local IPv6 Address . . . . . : fe80::869:bb10:be3b:5594%51  
IPv4 Address. . . . . : 172.22.32.1  
Subnet Mask . . . . . : 255.255.240.0  
Default Gateway . . . . . :
```

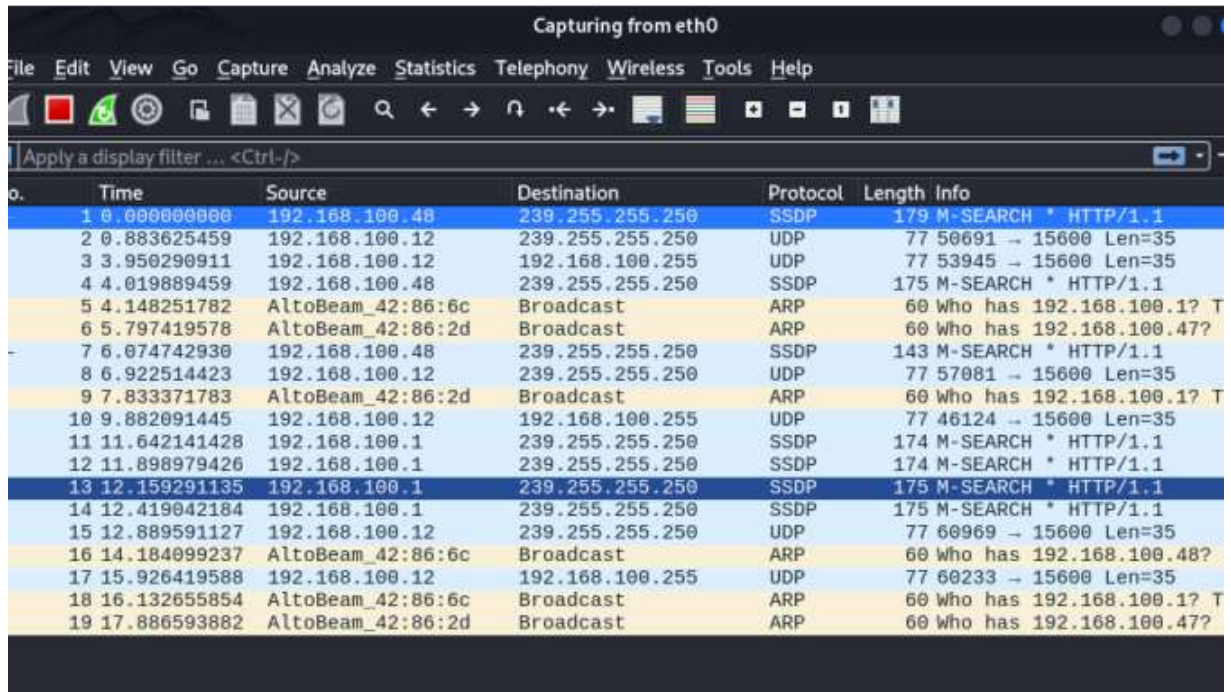
C:\Users\firda>

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.100.40 netmask 255.255.255.0 broadcast 192.168.100.255  
inet6 fe80::a00:27ff:fe38:c803 prefixlen 64 scopeid 0x20<link>  
ether 08:00:27:38:c8:03 txqueuelen 1000 (Ethernet)  
RX packets 102 bytes 9174 (8.9 KiB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 41 bytes 5292 (5.1 KiB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Interface: 192.168.100.8 --- 0xc

Internet Address	Physical Address	Type
192.168.100.1	c8-84-cf-e9-7f-52	dynamic
192.168.100.3	62-14-b8-56-e1-99	dynamic
192.168.100.12	e8-aa-cb-24-11-7e	dynamic
192.168.100.40	08-00-27-38-c8-03	dynamic
192.168.100.255	ff-ff-ff-ff-ff-ff	static
224.0.0.2	01-00-5e-00-00-02	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static
239.255.255.250	01-00-5e-7f-ff-fa	static
255.255.255.255	ff-ff-ff-ff-ff-ff	static

Ensuring that Wireshark is running in the background



The image shows a Wireshark window titled "Capturing from eth0". The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help), a toolbar with icons for file operations, capture, and analysis, and a display filter bar. The main packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.100.48	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1
2	0.883625459	192.168.100.12	239.255.255.250	UDP	77	50691 → 15600 Len=35
3	3.950290911	192.168.100.12	192.168.100.255	UDP	77	53945 → 15600 Len=35
4	4.019889459	192.168.100.48	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
5	4.148251782	AltoBeam_42:86:6c	Broadcast	ARP	60	Who has 192.168.100.1? T
6	5.797419578	AltoBeam_42:86:2d	Broadcast	ARP	60	Who has 192.168.100.47? T
7	6.074742930	192.168.100.48	239.255.255.250	SSDP	143	M-SEARCH * HTTP/1.1
8	6.922514423	192.168.100.12	239.255.255.250	UDP	77	57081 → 15600 Len=35
9	7.833371783	AltoBeam_42:86:2d	Broadcast	ARP	60	Who has 192.168.100.1? T
10	9.882091445	192.168.100.12	192.168.100.255	UDP	77	46124 → 15600 Len=35
11	11.642141428	192.168.100.1	239.255.255.250	SSDP	174	M-SEARCH * HTTP/1.1
12	11.898979426	192.168.100.1	239.255.255.250	SSDP	174	M-SEARCH * HTTP/1.1
13	12.159291135	192.168.100.1	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
14	12.419042184	192.168.100.1	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
15	12.889591127	192.168.100.12	239.255.255.250	UDP	77	60969 → 15600 Len=35
16	14.184099237	AltoBeam_42:86:6c	Broadcast	ARP	60	Who has 192.168.100.48? T
17	15.926419588	192.168.100.12	192.168.100.255	UDP	77	60233 → 15600 Len=35
18	16.132655854	AltoBeam_42:86:6c	Broadcast	ARP	60	Who has 192.168.100.1? T
19	17.886593882	AltoBeam_42:86:2d	Broadcast	ARP	60	Who has 192.168.100.47? T

Opening Ettercap to simulate the attack then I went ahead and continued Scanning for hosts I got. Which are a total of 8 hosts.



Setting 192.168.100.1 (c8-84-cf-e9-7f-52) as Target 1



Starting ARP poisoning

The mac address is now changed from c8-84-cf-e9-7f-52 to 08-00-27-38-c8-03

```
C:\Users\firda>arp -a

Interface: 192.168.199.1 --- 0xa
  Internet Address      Physical Address      Type
  192.168.199.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.100.8 --- 0xc
  Internet Address      Physical Address      Type
  192.168.100.1         08-00-27-38-c8-03     dynamic
  192.168.100.3         62-14-b8-56-e1-99     dynamic
  192.168.100.12        e8-aa-cb-24-11-7e     dynamic
  192.168.100.40        08-00-27-38-c8-03     dynamic
  192.168.100.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Capturing login details

```
C:\Users\firda>arp -a
```

```
Interface: 192.168.199.1 --- 0xa
```

Internet Address	Physical Address	Type
192.168.199.255	ff-ff-ff-ff-ff-ff	static
224.0.0.2	01-00-5e-00-00-02	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static
224.0.0.252	01-00-5e-00-00-fc	static
239.255.255.250	01-00-5e-7f-ff-fa	static

```
Interface: 192.168.100.8 --- 0xc
```

Internet Address	Physical Address	Type
192.168.100.1	c8-84-cf-e9-7f-52	dynamic
192.168.100.3	62-14-b8-56-e1-99	dynamic
192.168.100.12	e8-aa-cb-24-11-7e	dynamic
192.168.100.40	08-00-27-38-c8-03	dynamic
192.168.100.255	ff-ff-ff-ff-ff-ff	static
224.0.0.2	01-00-5e-00-00-02	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static
239.255.255.250	01-00-5e-7f-ff-fa	static
255.255.255.255	ff-ff-ff-ff-ff-ff	static

Login credentials extracted

Not secure | testphp.vulnweb.com/login.php

Kotlin | Kotlin Docu... | Strathmore eLearn... | Teams and Channel... | LearnOpenGL - Hell... | class

acunetix acuart

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art
 go

browse categories
browse artists
your cart
signup
your profile
our guestbook
AJAX Demo

links
Security art
PHP scanner
PHP vuln help
Fractal Explorer

If you are already registered please enter your login information below:

Username :

Password :

login

You can also [signup here](#).
Signup disabled. Please use the username **test** and the password **test**.

Ettercap analysis

```
GROUP 1 : 192.168.100.1 C8:84:CF:E9:7F:52

GROUP 2 : ANY (all the hosts in the list)
HTTP : 44.228.249.3:80 -> USER: simon PASS: 1234567 INFO: http://testphp.vulnweb.com/login.php
CONTENT: uname=simon&pass=1234567
```


Wireshark capture

Capturing from eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

198.168.100.1

No.	Time	Source	Destination	Protocol	Length	Info
16642	670.230413127	192.168.100.3	41.90.2.163	TCP	66	56060 → 443 [ACK] Seq=120
16643	670.234209975	41.90.2.163	192.168.100.3	TCP	1446	[TCP Retransmission] 443
16644	670.234346424	41.90.2.163	192.168.100.3	TCP	1446	[TCP Retransmission] 443
16645	670.234451174	192.168.100.3	41.90.2.163	TCP	66	[TCP Dup ACK 16642#1] 560
16646	670.234823882	41.90.2.163	192.168.100.3	TLSv1.3	1446	Continuation Data
16647	670.236019927	41.90.2.163	192.168.100.3	TLSv1.3	1446	Continuation Data
16648	670.242190011	41.90.2.163	192.168.100.3	TCP	1446	[TCP Retransmission] 443
16649	670.242319159	41.90.2.163	192.168.100.3	TCP	1446	[TCP Retransmission] 443
16650	670.242675151	41.90.2.163	192.168.100.3	TLSv1.3	1446	Continuation Data
16651	670.243307946	41.90.2.163	192.168.100.3	TLSv1.3	1446	Continuation Data
16652	670.250168679	41.90.2.163	192.168.100.3	TLSv1.3	1446	Continuation Data
16653	670.250311242	41.90.2.163	192.168.100.3	TCP	1446	[TCP Retransmission] 443
16654	670.250470719	41.90.2.163	192.168.100.3	TCP	1446	[TCP Retransmission] 443
16655	670.250736120	41.90.2.163	192.168.100.3	TCP	1446	443 → 56060 [PSH, ACK] Seq
16656	670.257331252	41.90.2.163	192.168.100.3	TLSv1.3	1446	Continuation Data
16657	670.257971026	41.90.2.163	192.168.100.3	TLSv1.3	1446	Continuation Data
16658	670.258264286	41.90.2.163	192.168.100.3	TCP	1446	[TCP Retransmission] 443
16659	670.258408682	41.90.2.163	192.168.100.3	TCP	1446	[TCP Retransmission] 443
16660	670.258504977	41.90.2.163	192.168.100.3	TCP	1446	[TCP Retransmission] 443
16661	670.258595080	41.90.2.163	192.168.100.3	TCP	1446	[TCP Retransmission] 443
16662	670.264816839	41.90.2.163	192.168.100.3	TLSv1.3	1446	Continuation Data
16663	670.265334842	41.90.2.163	192.168.100.3	TLSv1.3	1446	Continuation Data

POST /userinfo.php HTTP/1.1\r\n
Host: testphp.vulnweb.com\r\n
Connection: keep-alive\r\n
Content-Length: 24\r\n
Cache-Control: max-age=0\r\n
Origin: http://testphp.vulnweb.com\r\n
Content-Type: application/x-www-form-urlencoded\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chr...
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*...
Referer: http://testphp.vulnweb.com/login.php\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
\r\n
[Full request URI: http://testphp.vulnweb.com/userinfo.php]
[HTTP request 1/2]
[Next request in frame: 10241]
File Data: 24 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "uname" = "simon"
Form item: "pass" = "1234567"