OPEN SOURCE TECHNOLOGIES

REPORT

BY

Name: Kashish Sood

Section: KE023

Roll no: 37

Reg. no: 11902000

School of Computer Science and Engineering

Lovely Professional University

Phagwara

Punjab

REPORT

QUESTION: Being the man in the middle, you are asked to tamper ARP table of the victim machine and sniff network traffic between the victim and server. Also, provide remediation to such kind of intrusion. Use any open-source software to generate a report on the same.

1. Introduction:

1.1. Objective:

With the help of ettercap, I will tamper the ARP table of the victim machine and with the help of wireshark, I will analyze or sniff the network traffic between the victim and server.

1.2. Description:

MITM: A man-in-the-middle attack is a cyber attack in which a threat actor puts themselves in the middle of two parties, typically a user and an application, to intercept their communications and data exchanges and use them for malicious purposes like making unauthorized purchases or hacking.

The MITM attack consists of two phases: interception and decryption.

- Interception kicks off the attack. In this phase, the cybercriminal works to intercept your online activities before you reach your intended destination.
- The decryption phase consists of the criminal quietly decoding stolen data and decrypting secure connections so you can’t tell there’s a monster in the middle.

ARP Poisoning: It is also known as ARP Spoofing. It is a type of cyber attack carried out over a Local Area Network (LAN) that involves sending malicious ARP packets to a default gateway on a LAN in order to change the pairings in its IP to MAC address table. ARP Protocol translates IP addresses into MAC addresses.

ARP: The Address Resolution Protocol (ARP) handles the mapping between an Internet Protocol (IP) address and a Media Access Control (MAC) address. Linux stores all such mappings in a local system cache called an ARP table. In order to tamper ARP table, I have used Ettercap.

Ettercap: It is an open-source tool that can be used to support man-in-the-middle attacks on networks. It can capture packets and then write them back onto the network. It also enables the diversion and alteration of data virtually in real-time.

Wireshark: In order to sniff network traffic between the victim and server, I have used wireshark. It is a network protocol analyzer, or an application that captures packets from a network connection, such as from your computer to your home office or the internet. Packet is the name given to a discrete unit of data in a typical Ethernet network. Also, wireshark is the most often used packet sniffer in the world.

DNS spoofing: It is also known as DNS cache poisoning, involves infiltrating a DNS server and altering a website’s address record. As a result, users attempting to access the site are sent by the altered DNS record to the attacker’s site.

1.3. Scope: MITM attacks are likely to grow more common as we connect additional devices to Wi-Fi networks. From the internet-connected smart doorbell to online protection systems, cybercriminals have increasing opportunities to hack our networks. And many of these newer devices have little or no security. Adopting a preventive mindset and strictly adhering to secure connections will help to keep you, your business, and your staff safe from MITM and other cyber threats.

2. Analysis Report:

2.1. Pre-requisites:

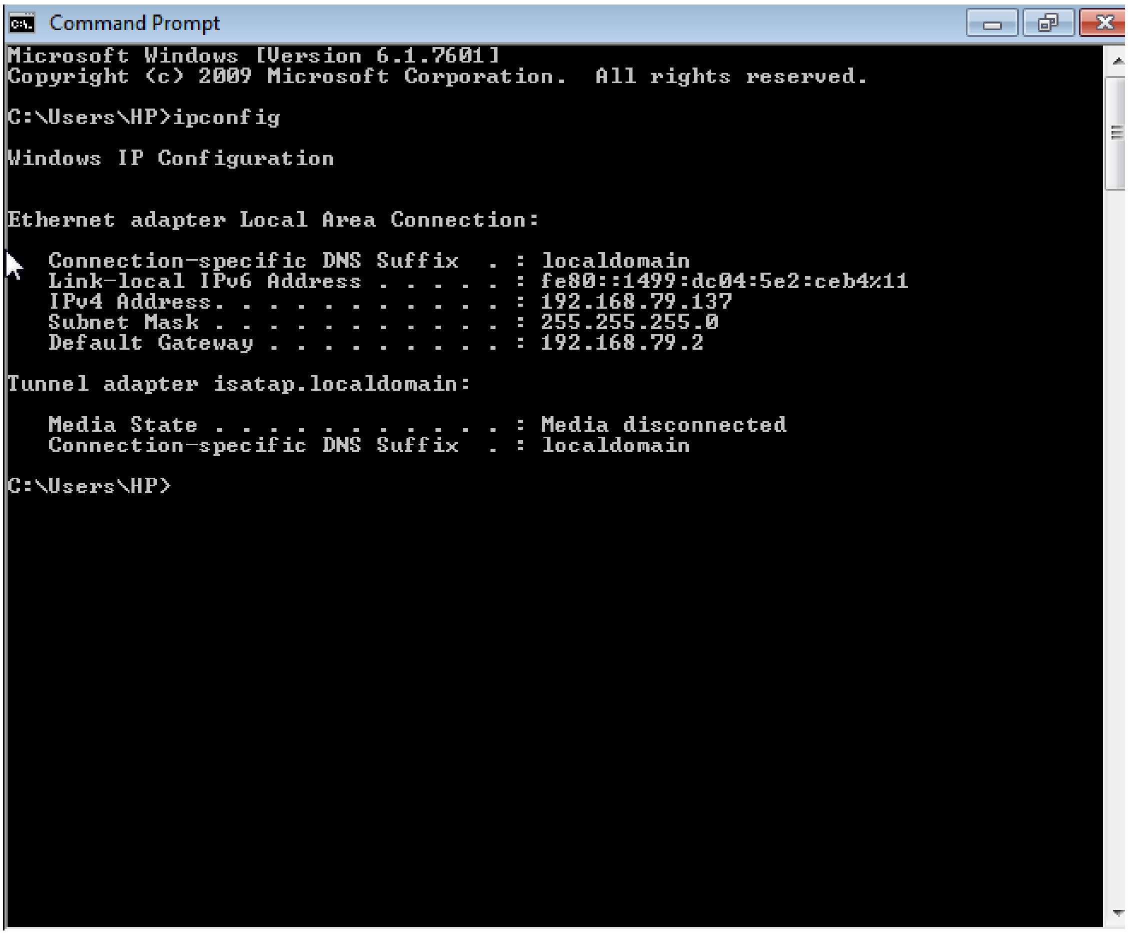
- Victim machine: Windows 7
- Attacker machine: Kali Linux

Firstly we need to perform ARP poisoning:

We would be doing it using Ettercap:

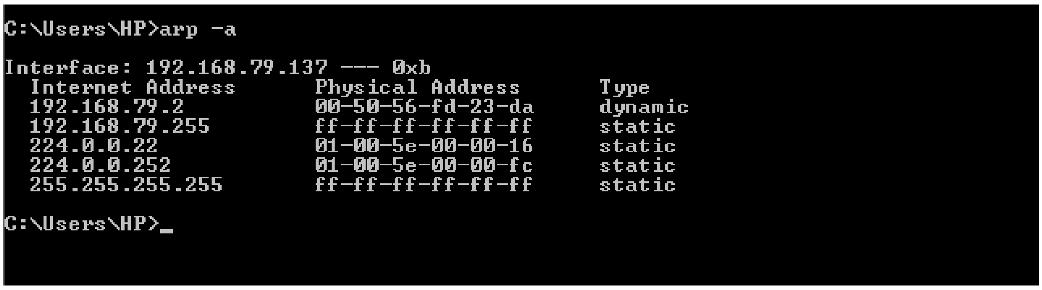
It is an open-source tool that can be used to support man-in-the-middle attacks on networks. It can capture packets and then write them back onto the network. It also enables the diversion and alteration of data virtually in real-time.

Let’s first check the IP of victim machine:



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\HP>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : localdomain
   Link-local IPv6 Address . . . . . : fe80::1499:dc04:5e2:ceb4%11
   IPv4 Address. . . . . . . . . . . : 192.168.79.137
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.79.2

Tunnel adapter isatap.localdomain:

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

C:\Users\HP>
```

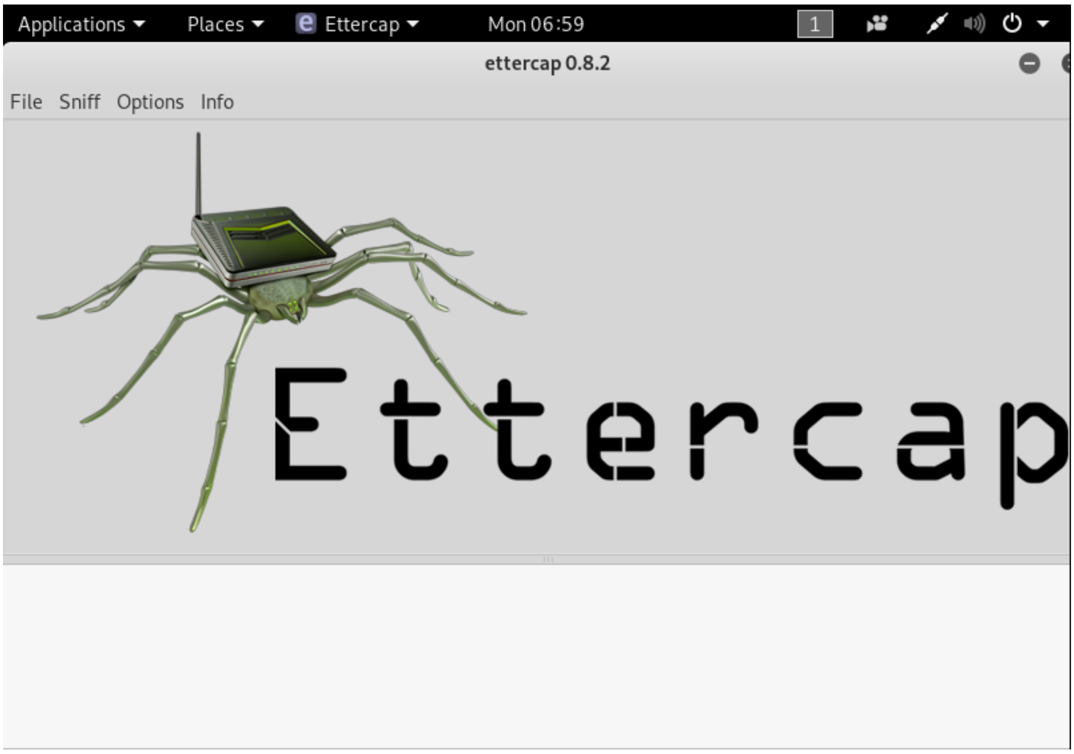
Now let’s check the initial ARP table:



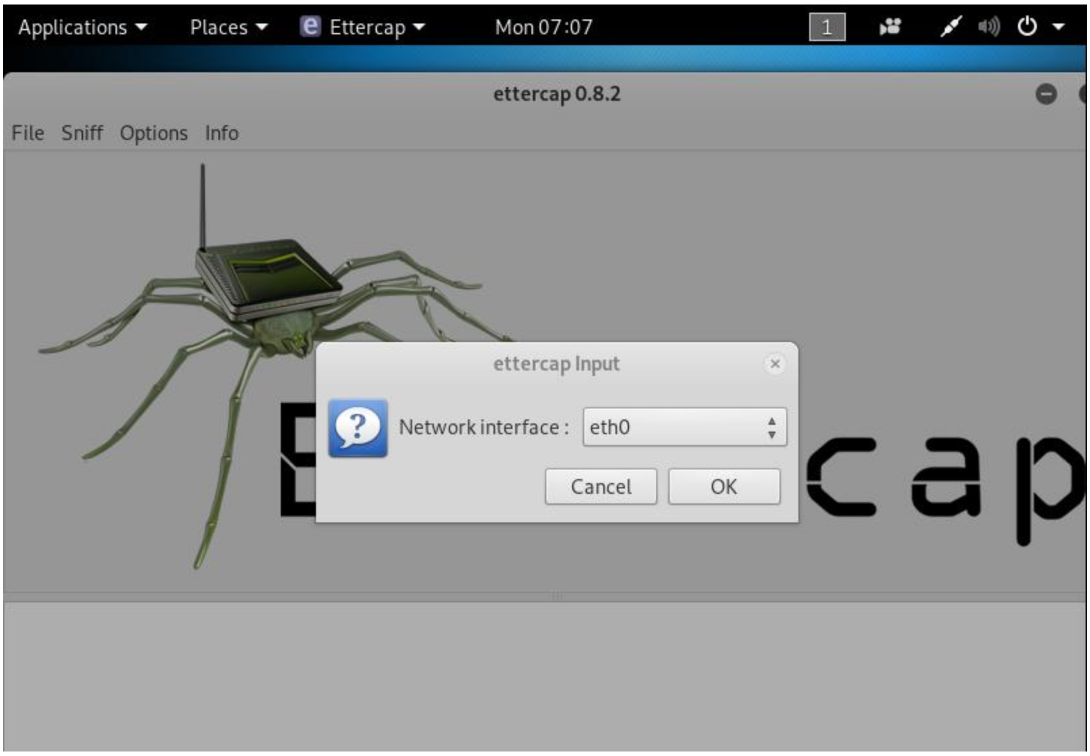
```
C:\Users\HP>arp -a

Interface: 192.168.79.137 --- 0xb
  Internet Address      Physical Address      Type
  192.168.79.2          00-50-56-fd-23-da     dynamic
  192.168.79.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\HP>_
```

Now in the Kali Linux:

Open the Ettercap tool. The main page looks like the following:

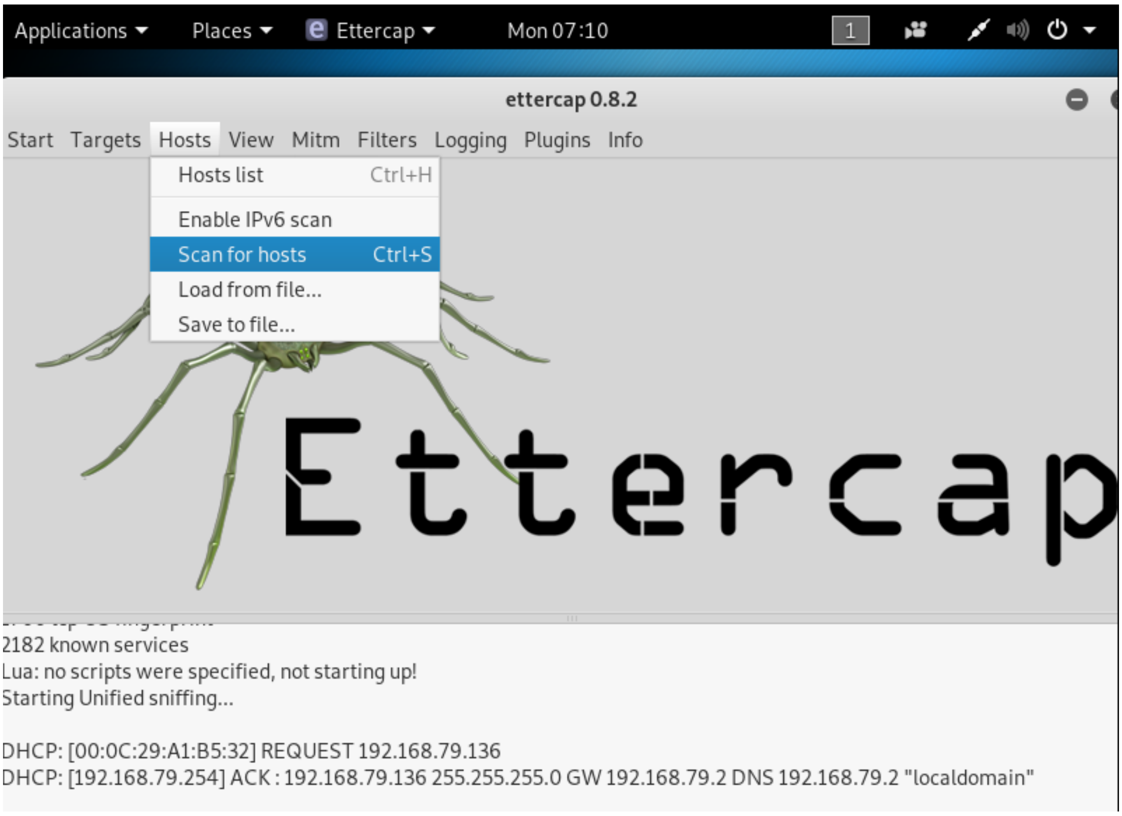


Now select the network to be sniffed:



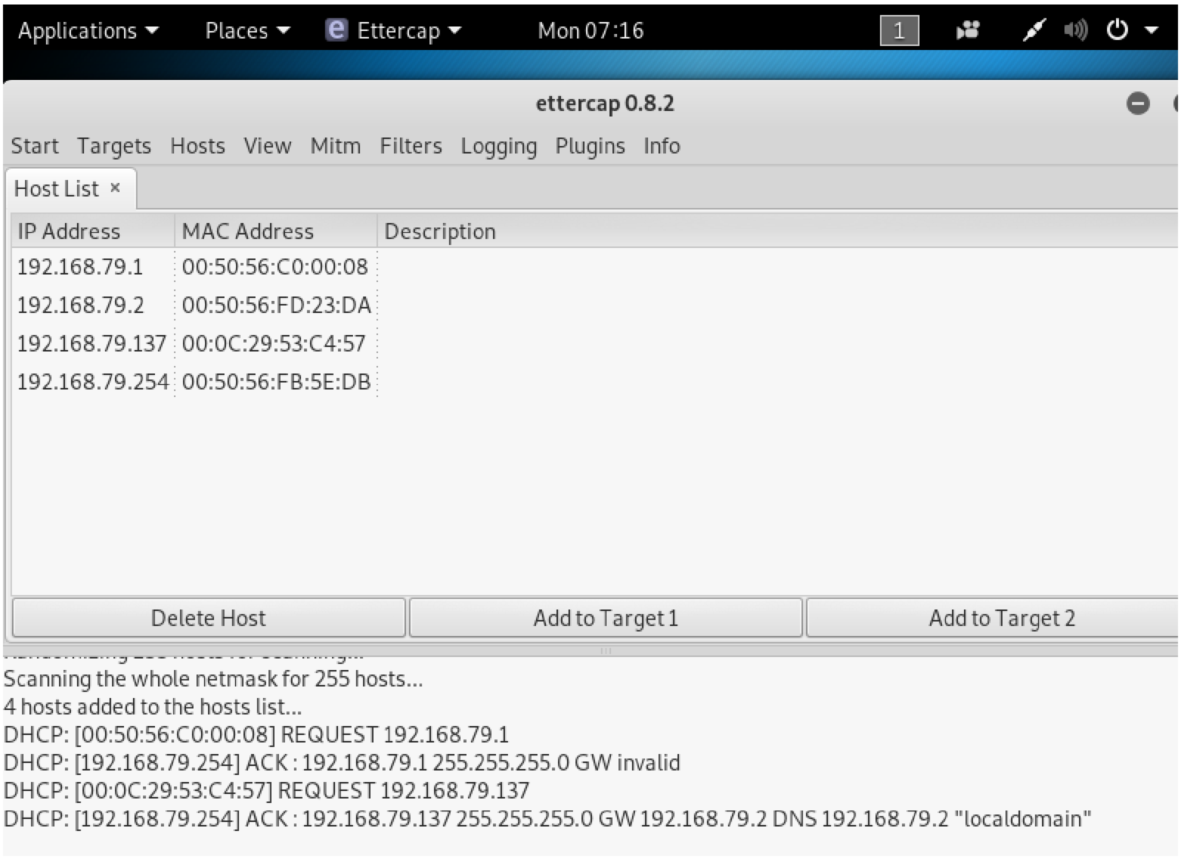
Network interface : eth0

Scan for hosts in the network:



```
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...

DHCP: [00:0C:29:A1:B5:32] REQUEST 192.168.79.136
DHCP: [192.168.79.254] ACK : 192.168.79.136 255.255.255.0 GW 192.168.79.2 DNS 192.168.79.2 "localdomain"
```

It gives the following hosts list:



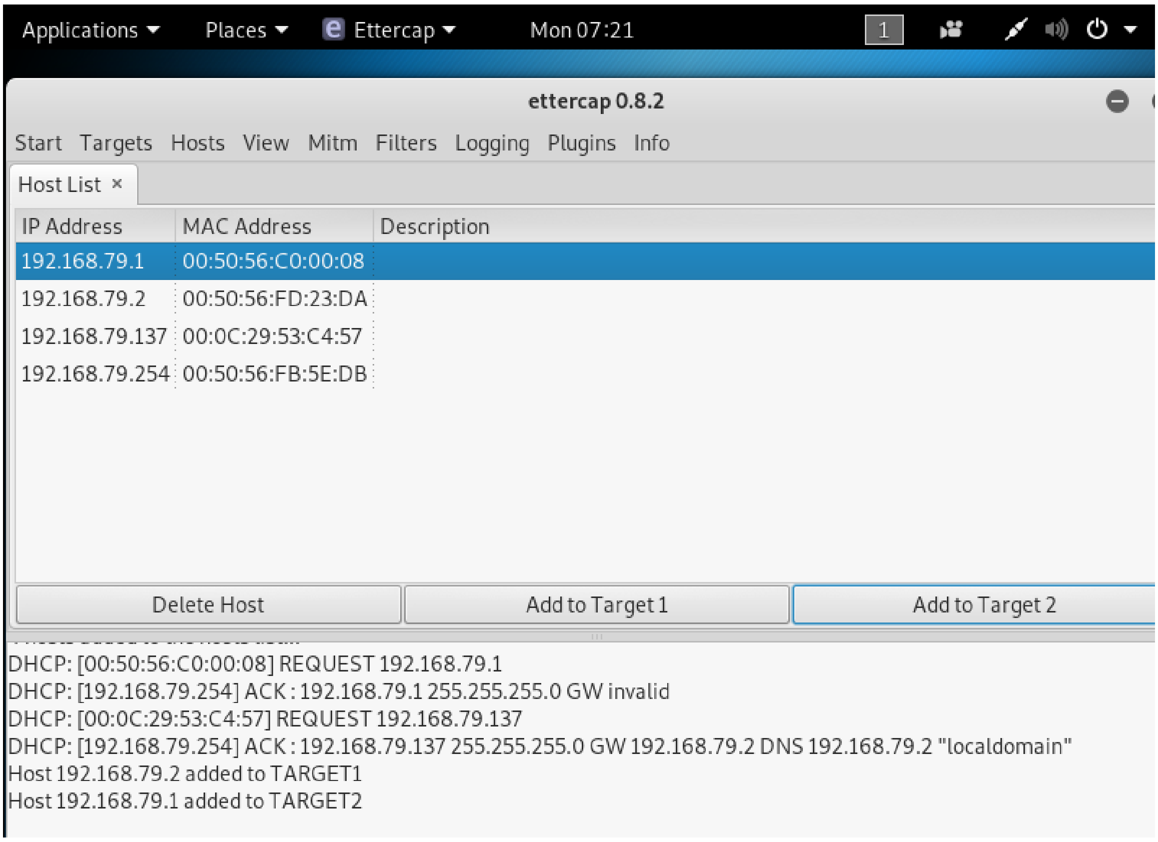
IP Address	MAC Address	Description
192.168.79.1	00:50:56:C0:00:08	
192.168.79.2	00:50:56:FD:23:DA	
192.168.79.137	00:0C:29:53:C4:57	
192.168.79.254	00:50:56:FB:5E:DB	

```
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
4 hosts added to the hosts list...
DHCP: [00:50:56:C0:00:08] REQUEST 192.168.79.1
DHCP: [192.168.79.254] ACK : 192.168.79.1 255.255.255.0 GW invalid
DHCP: [00:0C:29:53:C4:57] REQUEST 192.168.79.137
DHCP: [192.168.79.254] ACK : 192.168.79.137 255.255.255.0 GW 192.168.79.2 DNS 192.168.79.2 "localdomain"
```

We can see that our victim machine IP is available in the list.

- Now add the victim machine IP to target 1:
  (Add 192.168.79.2 to target 1)
- And add gateway IP address to target 2:
  (add 192.168.79.1 to target 2)

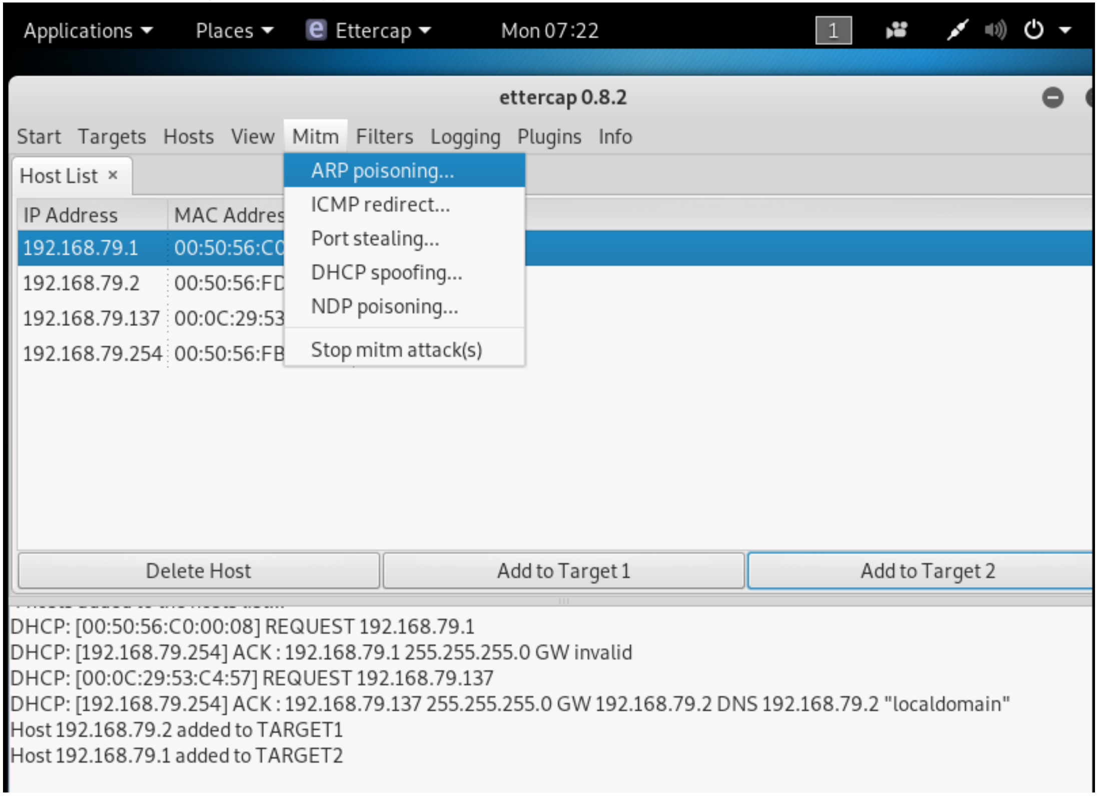
Gateway IP address is not used for any machine IP addresses it only acts as a network Gateway where the attacker can be present to perform MITM attack.



IP Address	MAC Address	Description
192.168.79.1	00:50:56:C0:00:08	
192.168.79.2	00:50:56:FD:23:DA	
192.168.79.137	00:0C:29:53:C4:57	
192.168.79.254	00:50:56:FB:5E:DB	

```
DHCP: [00:50:56:C0:00:08] REQUEST 192.168.79.1
DHCP: [192.168.79.254] ACK : 192.168.79.1 255.255.255.0 GW invalid
DHCP: [00:0C:29:53:C4:57] REQUEST 192.168.79.137
DHCP: [192.168.79.254] ACK : 192.168.79.137 255.255.255.0 GW 192.168.79.2 DNS 192.168.79.2 "localdomain"
Host 192.168.79.2 added to TARGET1
Host 192.168.79.1 added to TARGET2
```

Now Start ARP poisoning:

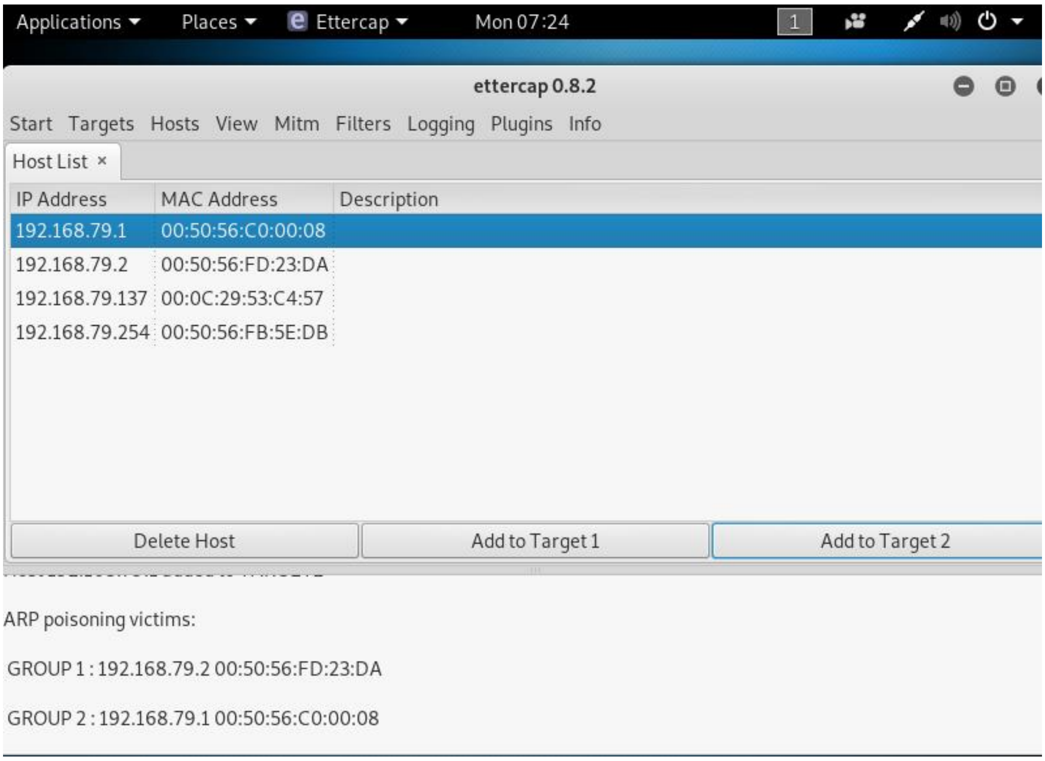


IP Address	MAC Addres
192.168.79.1	00:50:56:C0
192.168.79.2	00:50:56:FD
192.168.79.137	00:0C:29:53
192.168.79.254	00:50:56:FB

```
DHCP: [00:50:56:C0:00:08] REQUEST 192.168.79.1
DHCP: [192.168.79.254] ACK : 192.168.79.1 255.255.255.0 GW invalid
DHCP: [00:0C:29:53:C4:57] REQUEST 192.168.79.137
DHCP: [192.168.79.254] ACK : 192.168.79.137 255.255.255.0 GW 192.168.79.2 DNS 192.168.79.2 "localdomain"
Host 192.168.79.2 added to TARGET1
Host 192.168.79.1 added to TARGET2
```

You will get the result as follows:

(Here first and last IPS is for broadcasting and second is victim's IP address and third is other persons IP address whomever is connect to that network)



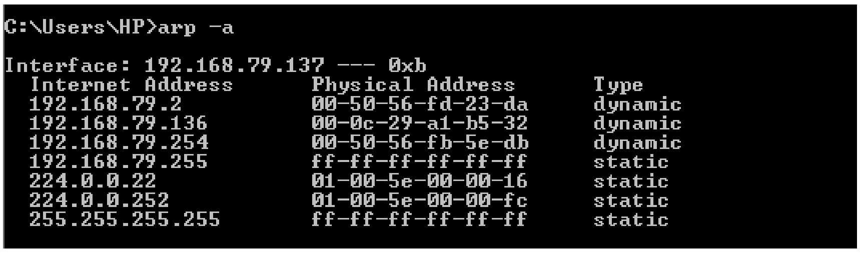
IP Address	MAC Address	Description
192.168.79.1	00:50:56:C0:00:08	
192.168.79.2	00:50:56:FD:23:DA	
192.168.79.137	00:0C:29:53:C4:57	
192.168.79.254	00:50:56:FB:5E:DB	

ARP poisoning victims:

GROUP 1 : 192.168.79.2 00:50:56:FD:23:DA

GROUP 2 : 192.168.79.1 00:50:56:C0:00:08

Now go to to Victim machine and check the ARP table:



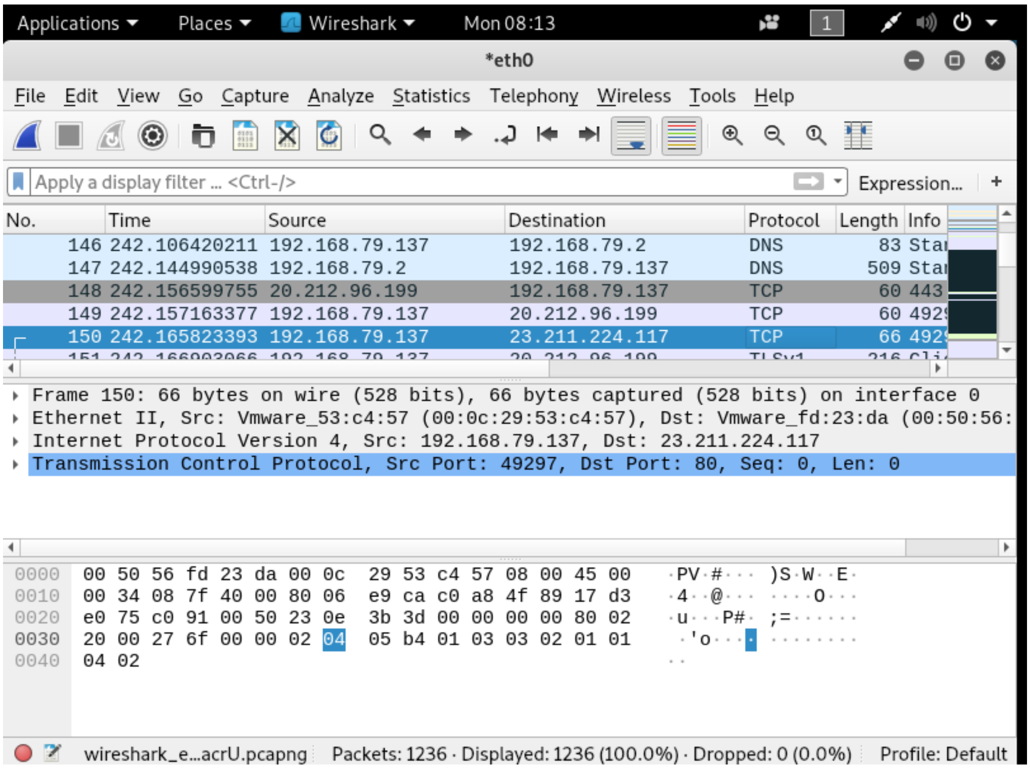
```
C:\Users\HP>arp -a

Interface: 192.168.79.137 --- 0xb
  Internet Address      Physical Address      Type
  192.168.79.2          00-50-56-fd-23-da     dynamic
  192.168.79.136        00-0c-29-a1-b5-32     dynamic
  192.168.79.254        00-50-56-fb-5e-db     dynamic
  192.168.79.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

You can see the IP: 192.168.79.254 has the same MAC Address as the IP address of the victim's machine (192.168.79.2). This is how using ettercap, we can easily tamper the ARP table.

Now for the second part of the question.

In order to sniff network traffic between the victim and server:

Identifying this attack on Wireshark:


Applications
Places
Wireshark
Mon 08:13
1
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
+

```
No.   Time           Source          Destination     Protocol Length Info
  146 242.106420211 192.168.79.137  192.168.79.2    DNS         83 Sta
  147 242.144990538 192.168.79.2    192.168.79.137  DNS        509 Sta
  148 242.156599755 20.212.96.199   192.168.79.137  TCP         60 443
  149 242.157163377 192.168.79.137  20.212.96.199   TCP         60 492
  150 242.165823393 192.168.79.137  23.211.224.117  TCP         66 492
  151 242.166903066 192.168.79.137  20.212.96.199   TLSv1      216 Cli
```

```
▸ Frame 150: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▸ Ethernet II, Src: Vmware_53:c4:57 (00:0c:29:53:c4:57), Dst: Vmware_fd:23:da (00:50:56:
▸ Internet Protocol Version 4, Src: 192.168.79.137, Dst: 23.211.224.117
▸ Transmission Control Protocol, Src Port: 49297, Dst Port: 80, Seq: 0, Len: 0
```

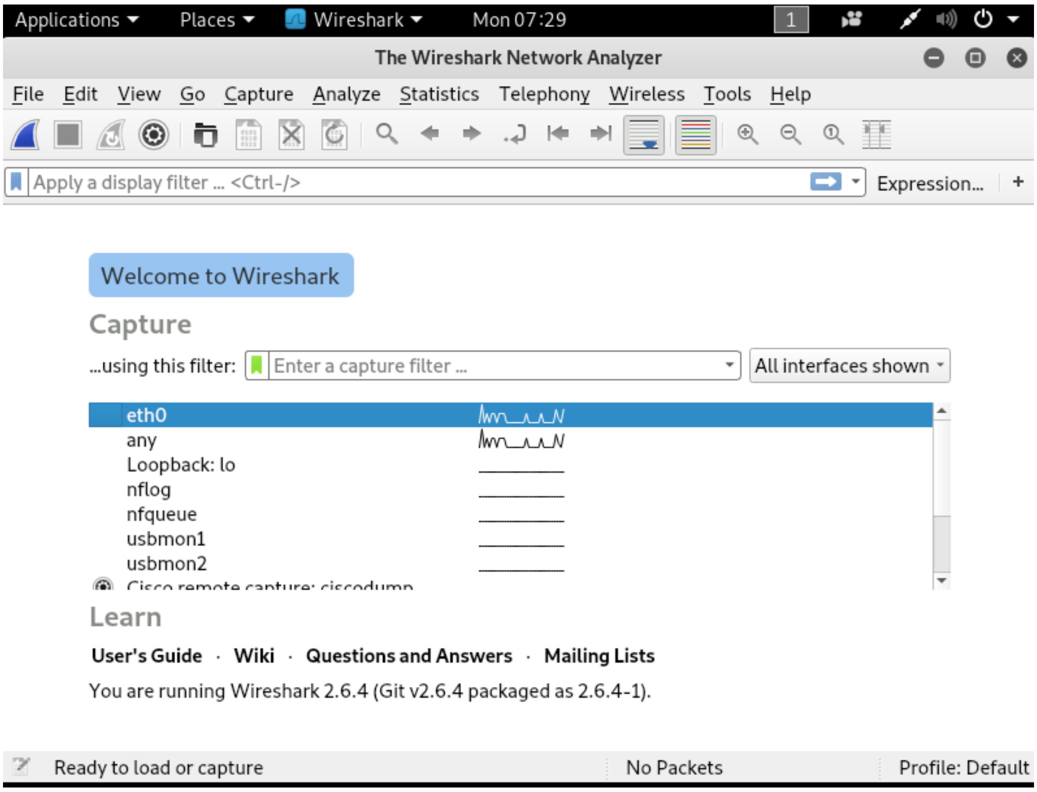
```
0000  00 50 56 fd 23 da 00 0c  29 53 c4 57 08 00 45 00   ·PV·#···  )S·W··E·
0010  00 34 08 7f 40 00 80 06  e9 ca c0 a8 4f 89 17 d3   ·4··@···  ····O···
0020  e0 75 c0 91 00 50 23 0e  3b 3d 00 00 00 00 80 02   ·u···P#·  ;=······
0030  20 00 27 6f 00 00 02 04  05 b4 01 03 03 02 01 01    ·'o····  ········
0040  04 02                                              ··
```

wireshark_e…acrU.pcapng · Packets: 1236 · Displayed: 1236 (100.0%) · Dropped: 0 (0.0%) · Profile: Default

In this image you can view that the DNS server of victim machine is connected to attacker machine and the attacker is receiving TCP request from the server.

Now sniffing the network traffic between victim and server using WIRESHARK:

The Wireshark interface is as follows:


Applications
Places
Wireshark
Mon 07:29
The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

```
eth0
any
Loopback: lo
nflog
nfqueue
usbmon1
usbmon2
Cisco remote capture: ciscodump
```

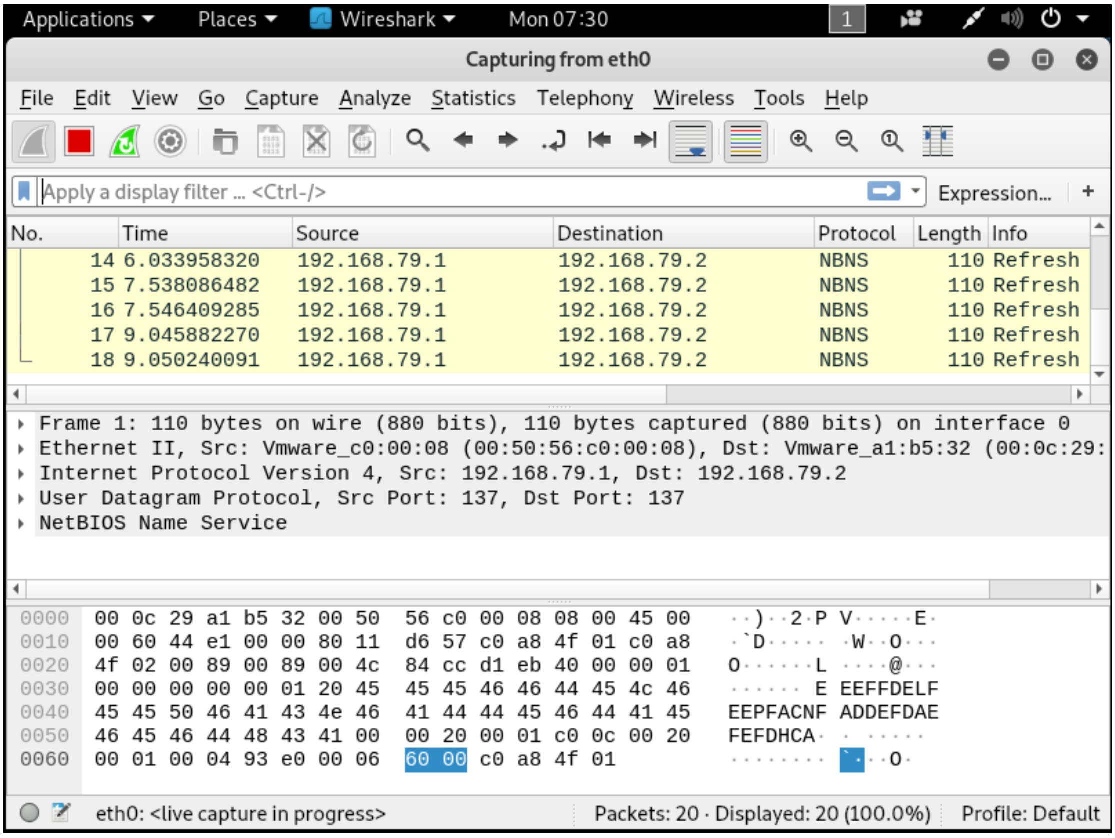
You are running Wireshark 2.6.4 (Git v2.6.4 packaged as 2.6.4-1).

Ready to load or capture

No Packets

Profile: Default

Select the network to start sniffing packets. We have selected eth0 network and the wireshark has started sniffing:


Applications
Places
Wireshark
Mon 07:30
Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+

No.	Time	Source	Destination	Protocol	Length	Info
14	6.033958320	192.168.79.1	192.168.79.2	NBNS	110	Refresh
15	7.538086482	192.168.79.1	192.168.79.2	NBNS	110	Refresh
16	7.546409285	192.168.79.1	192.168.79.2	NBNS	110	Refresh
17	9.045882270	192.168.79.1	192.168.79.2	NBNS	110	Refresh
18	9.050240091	192.168.79.1	192.168.79.2	NBNS	110	Refresh

```
Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08), Dst: Vmware_a1:b5:32 (00:0c:29:
Internet Protocol Version 4, Src: 192.168.79.1, Dst: 192.168.79.2
User Datagram Protocol, Src Port: 137, Dst Port: 137
NetBIOS Name Service
```

```
0000  00 0c 29 a1 b5 32 00 50  56 c0 00 08 08 00 45 00   ··)··2·P V·····E·
0010  00 60 44 e1 00 00 80 11  d6 57 c0 a8 4f 01 c0 a8   ·`D····· ·W··O···
0020  4f 02 00 89 00 89 00 4c  84 cc d1 eb 40 00 00 01   O······L ····@···
0030  00 00 00 00 00 01 20 45  45 45 46 46 44 45 4c 46   ······ E EEFFDELF
0040  45 45 50 46 41 43 4e 46  41 44 44 45 46 44 41 45   EEPFACNF ADDEFDAE
0050  46 45 46 44 48 43 41 00  00 20 00 01 c0 0c 00 20   FEFDHCA· · ·····
0060  00 01 00 04 93 e0 00 06  60 00 c0 a8 4f 01         ········ `··O·
```

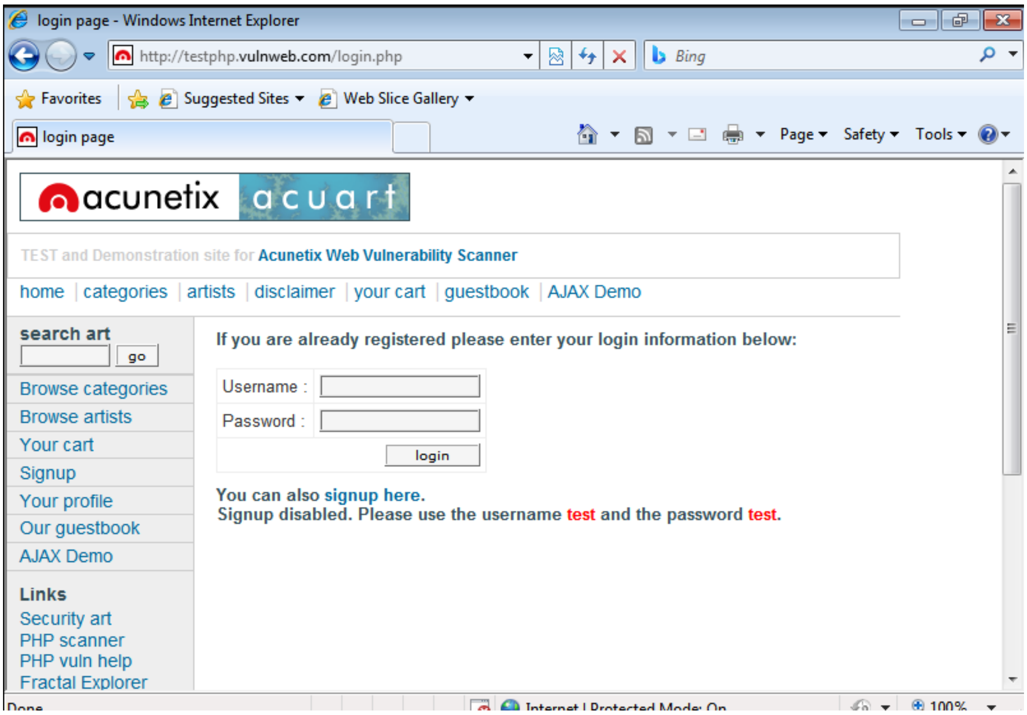
eth0: <live capture in progress>

Packets: 20 · Displayed: 20 (100.0%)

Profile: Default

Now on the Victim machine, we opened a website Vulweb (a cybersecurity website to test attacks)

And opened its Login page:



TEST and Demonstration site for Acunetix Web Vulnerability Scanner

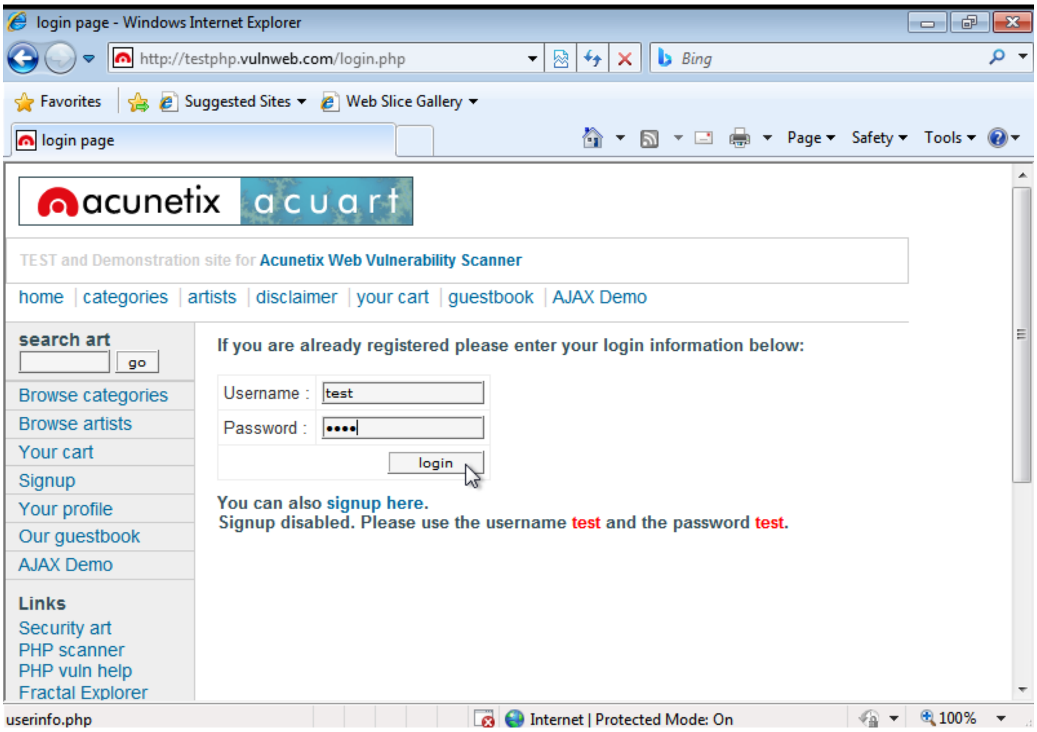
If you are already registered please enter your login information below:

Username :

Password :

You can also signup here.
Signup disabled. Please use the username test and the password test.

Now enter the login credentials on the website, so we can capture it on the wireshark interface:



TEST and Demonstration site for Acunetix Web Vulnerability Scanner

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

Username : test

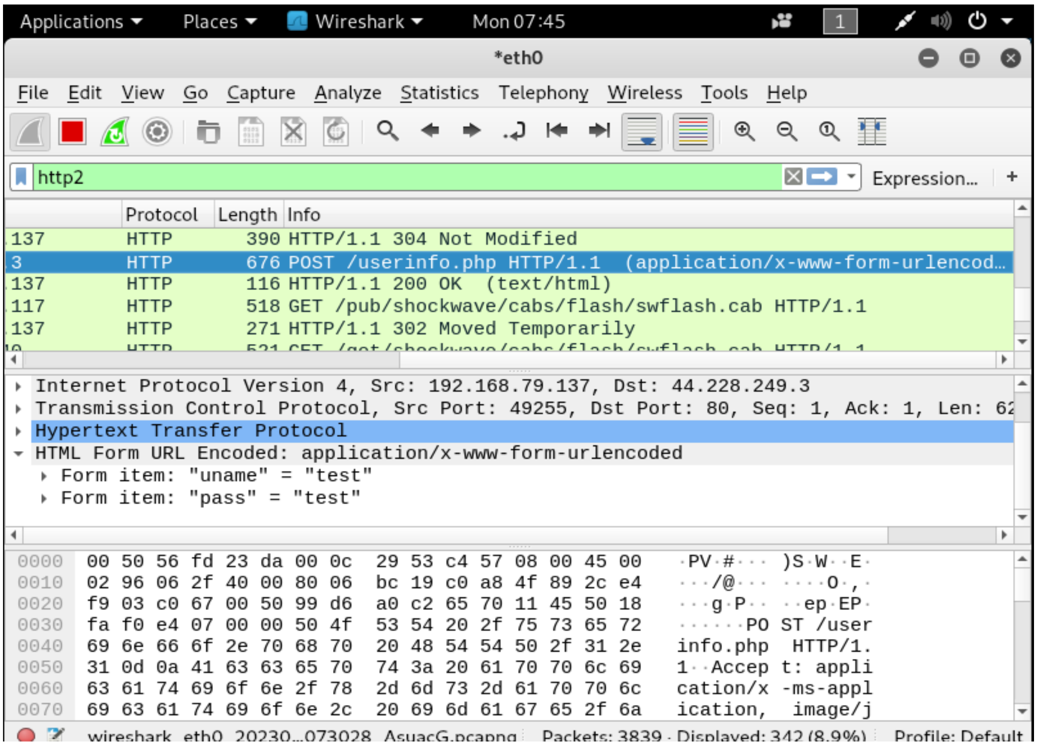
Password : ••••

You can also signup here.
Signup disabled. Please use the username test and the password test.

Now if you check the Wireshark on Attacker machine and apply the filter http, you can view http packets.

Search for the POST request, as we have give data into the website it goes as a POST request.

If you select the packet and view HTML form you can view the Login credentials you have entered.


Applications
Places
Wireshark
Mon 07:45
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http2
Expression…
+

```
         Protocol  Length Info
137      HTTP         390 HTTP/1.1 304 Not Modified
3        HTTP         676 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencod…
137      HTTP         116 HTTP/1.1 200 OK  (text/html)
117      HTTP         518 GET /pub/shockwave/cabs/flash/swflash.cab HTTP/1.1
137      HTTP         271 HTTP/1.1 302 Moved Temporarily
10       HTTP         521 GET /get/shockwave/cabs/flash/swflash.cab HTTP/1.1
```

```
▸ Internet Protocol Version 4, Src: 192.168.79.137, Dst: 44.228.249.3
▸ Transmission Control Protocol, Src Port: 49255, Dst Port: 80, Seq: 1, Ack: 1, Len: 62
▸ Hypertext Transfer Protocol
▾ HTML Form URL Encoded: application/x-www-form-urlencoded
   ▸ Form item: "uname" = "test"
   ▸ Form item: "pass" = "test"
```

```
0000  00 50 56 fd 23 da 00 0c  29 53 c4 57 08 00 45 00   ·PV·#···  )S·W··E·
0010  02 96 06 2f 40 00 80 06  bc 19 c0 a8 4f 89 2c e4   ···/@···  ····O·,·
0020  f9 03 c0 67 00 50 99 d6  a0 c2 65 70 11 45 50 18   ···g·P··  ··ep·EP·
0030  fa f0 e4 07 00 00 50 4f  53 54 20 2f 75 73 65 72   ······PO  ST /user
0040  69 6e 66 6f 2e 70 68 70  20 48 54 54 50 2f 31 2e   info.php  HTTP/1.
0050  31 0d 0a 41 63 63 65 70  74 3a 20 61 70 70 6c 69   1··Accep  t: appli
0060  63 61 74 69 6f 6e 2f 78  2d 6d 73 2d 61 70 70 6c   cation/x  -ms-appl
0070  69 63 61 74 69 6f 6e 2c  20 69 6d 61 67 65 2f 6a   ication,  image/j
```

wireshark_eth0_20230…073028_AsuacG.pcapng | Packets: 3839 · Displayed: 342 (8.9%) | Profile: Default

This is how with the help of wireshark, we can sniff the network traffic and check the credentials added by the user in the login page.

3. Preventive measures:
   - Secure connections: A secure internet connection is your first line of defense. To that end, only visit websites with a secure HTTP connection using SSL (Secure Socket Layer) technology. The additional SSL protection prevents MITM attacks. Sticking with secure websites isn't the only important thing to do, but also avoid using any unsecured public Wi-Fi connections. With no security, these connections are easy for a criminal to hack and insert themselves between you and the websites you're using.

   - VPN: One of the best practices for network security is to use a VPN (virtual private network) when connecting online. A VPN encrypts the data you send online. This encryption stops the MITM attack from infiltrating your network traffic. Even if a criminal manages to access your network, the encrypted data blocks them from reading your messages or knowing which websites you're going to.

4. Bibliography:

   - https://www.imperva.com/learn/application-security/man-in-the-middle-attack-mitm/
   - https://www.advantageitm.com/blog/understanding-the-dangers-of-a-man-in-the-middle-attack

5. Github link:
   https://github.com/kashishsood11/Man-in-the-middle-attack