

LAB 5

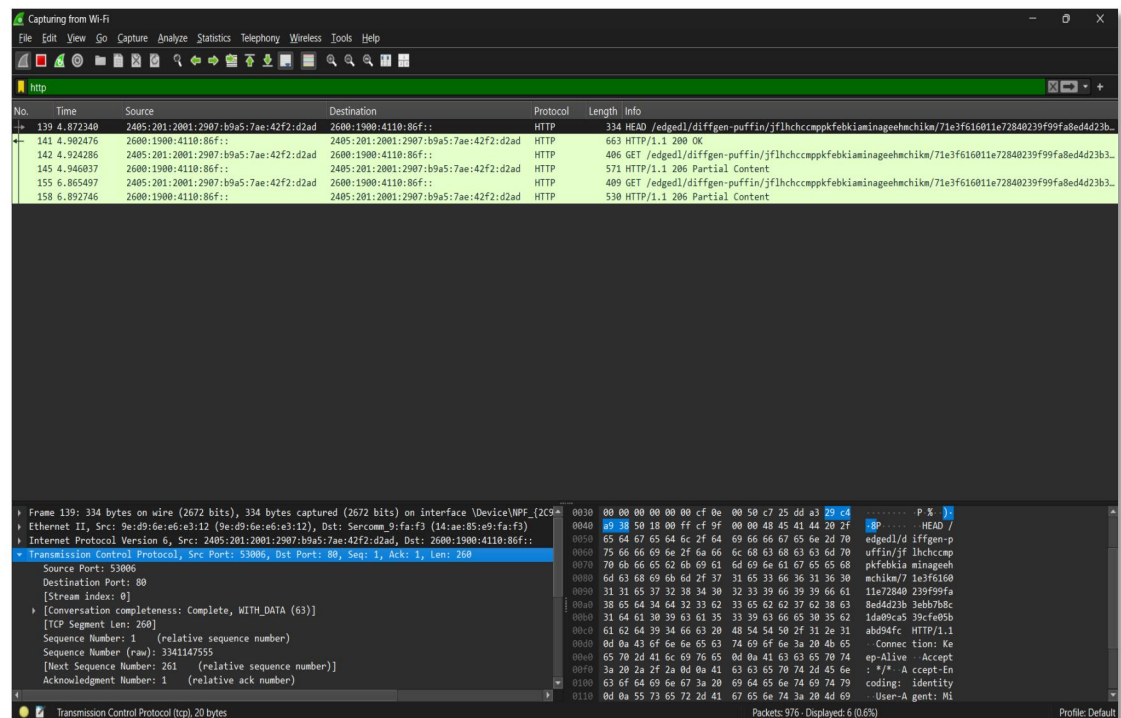
AIM: Using tools like Wireshark and tcpdump to capture and analyze mobile and IoT network traffic.

Introduction:

Wireshark and tcpdump are two of the most widely used tools for network traffic analysis. Wireshark, known for its powerful graphical interface and comprehensive packet analysis capabilities, is often favored for detailed, in-depth inspections of network data. Tcpdump, a command-line tool, is lightweight and highly effective for quick, real-time packet captures, making it ideal for mobile and IoT network environments where resources may be limited or where rapid diagnostics are needed.

By capturing and analyzing network traffic from mobile and IoT devices, these tools provide valuable insights into the data flow, protocols, and potential vulnerabilities in the communication between devices and servers. They are indispensable in ensuring that these networks run smoothly, efficiently, and securely. This analysis is especially crucial as the volume of connected devices continues to rise, and the complexity of mobile and IoT networks grows.

WireShark Capturing:



Tcpdump Capturing:

tcpdump -i any -c5 -nn port 80:

```

root@kali: /home/punit
[~] root@kali: /home/punit
# tcpdump -i any -c5 -nn port 80
tcpdump: WARNING: any: That device doesn't support promiscuous mode
(Promiscuous mode not supported on the "any" device)
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
23:12:33.006781 eth0 Out IP 192.168.40.128.35416 > 49.44.192.219.80: Flags [S], seq 619901991, win 64240, options [mss 1460,sackOK,TS val 1223532071 ecr 0,nop,wscale 7]
, length 0
23:12:33.011109 eth0 Out IP 192.168.40.128.35418 > 49.44.192.219.80: Flags [S], seq 525599053, win 64240, options [mss 1460,sackOK,TS val 1223532075 ecr 0,nop,wscale 7]
, length 0
23:12:33.012523 eth0 In IP 49.44.192.219.80 > 192.168.40.128.35416: Flags [S.], seq 1217277775, ack 619901992, win 64240, options [mss 1460], length 0
23:12:33.012567 eth0 Out IP 192.168.40.128.35416 > 49.44.192.219.80: Flags [.], ack 1, win 64240, length 0
23:12:33.012750 eth0 Out IP 192.168.40.128.35416 > 49.44.192.219.80: Flags [P.], seq 1:432, ack 1, win 64240, length 431: HTTP: POST / HTTP/1.1
5 packets captured
14 packets received by filter
0 packets dropped by kernel

[~] root@kali: /home/punit
#

```

tcp -c 20 -I eth0:

```

root@kali: /home/punit
[~] root@kali: /home/punit
# tcpdump -c 20 -I eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:20:44.566971 IP 192.168.40.128.59782 > 192.168.40.2.domain: 46584+ A? mozilla.cloudflare-dns.com. (44)
23:20:44.567122 IP 192.168.40.128.59782 > 192.168.40.2.domain: 33018+ AAAA? mozilla.cloudflare-dns.com. (44)
23:20:44.573322 IP 192.168.40.2.domain > 192.168.40.128.59782: 46584 2/0/0 A 172.64.41.4, A 162.159.61.4 (76)
23:20:44.580201 IP 192.168.40.2.domain > 192.168.40.128.59782: 33018 2/0/0 AAAA 2803:f800:53::4, AAAA 2a06:98c1:52::4 (100)
23:20:44.583603 IP 192.168.40.128.33666 > 172.64.41.4.https: Flags [S], seq 1110148024, win 64240, options [mss 1460,sackOK,TS val 1991640665 ecr 0,nop,wscale 7], length 0
23:20:44.583804 IP 192.168.40.128.33672 > 172.64.41.4.https: Flags [S], seq 253871562, win 64240, options [mss 1460,sackOK,TS val 1991640666 ecr 0,nop,wscale 7], length 0
23:20:44.583923 IP 192.168.40.128.33688 > 172.64.41.4.https: Flags [S], seq 370629416, win 64240, options [mss 1460,sackOK,TS val 1991640666 ecr 0,nop,wscale 7], length 0
23:20:44.602955 IP 172.64.41.4.https > 192.168.40.128.33672: Flags [S.], seq 160897904, ack 253871563, win 64240, options [mss 1460], length 0
23:20:44.602994 IP 192.168.40.128.33672 > 172.64.41.4.https: Flags [.], ack 1, win 64240, length 0
23:20:44.603666 IP 172.64.41.4.https > 192.168.40.128.33688: Flags [S.], seq 1153045515, ack 370629417, win 64240, options [mss 1460], length 0
23:20:44.603667 IP 172.64.41.4.https > 192.168.40.128.33666: Flags [S.], seq 723789063, ack 1110148025, win 64240, options [mss 1460], length 0
23:20:44.603679 IP 192.168.40.128.33688 > 172.64.41.4.https: Flags [.], ack 1, win 64240, length 0
23:20:44.603703 IP 192.168.40.128.33666 > 172.64.41.4.https: Flags [.], ack 1, win 64240, length 0
23:20:44.607398 IP 192.168.40.128.33672 > 172.64.41.4.https: Flags [P.], seq 1:675, ack 1, win 64240, length 674
23:20:44.607803 IP 172.64.41.4.https > 192.168.40.128.33672: Flags [.], ack 675, win 64240, length 0
23:20:44.608146 IP 192.168.40.128.33688 > 172.64.41.4.https: Flags [P.], seq 1:675, ack 1, win 64240, length 674
23:20:44.608408 IP 172.64.41.4.https > 192.168.40.128.33688: Flags [.], ack 675, win 64240, length 0
23:20:44.608704 IP 192.168.40.128.33666 > 172.64.41.4.https: Flags [P.], seq 1:675, ack 1, win 64240, length 674
23:20:44.608959 IP 172.64.41.4.https > 192.168.40.128.33666: Flags [.], ack 675, win 64240, length 0
23:20:44.628718 IP 172.64.41.4.https > 192.168.40.128.33688: Flags [P.], seq 1:3223, ack 675, win 64240, length 3222
20 packets captured
57 packets received by filter
0 packets dropped by kernel

[~] root@kali: /home/punit
#

```

Conclusion:

Using tools like Wireshark and tcpdump to capture and analyze mobile and IoT network traffic offers significant advantages for network administrators, security professionals, and developers. These tools allow for the detailed examination of the data exchanged between devices and servers, helping to diagnose issues, monitor performance, and ensure network security.