CCIDF Experiment No .11

DOP -10/10/2025
DOS- 17/10/2025

11.1 Aim: Network Forensics: CLI Utilities, OUI Lookup, Examining Network Packets

Example Scenario:
● You are working as a trainee in a cybersecurity lab. During a network audit, you detect multiple wireless clients connected to your office Wi-Fi. Some are known devices (Cisco Access Point, TP-Link router, Netgear extender), while one D-Link wireless client shows suspicious traffic patterns.
● Your task is to use command-line tools to inspect the network, perform an OUI lookup in Wireshark to identify the manufacturer of connected devices, and analyze captured packets to determine whether the D-Link client is behaving abnormally.

Tasks:
Check Network Configuration using CLI Utilities:
● Open Command Prompt (Windows) or Terminal (Linux).
● Run the following commands:
○ ipconfig / ifconfig → View IP and MAC details.
○ arp -a → List all connected IP and MAC addresses.
○ netstat -ano → Show all active connections and ports in use.
○ tracert or traceroute → Trace the path of data packets.
● Note any unfamiliar or unknown IP/MAC entries.

Perform OUI Lookup using Wireshark:
● Start Wireshark on your forensic workstation.
● Connect multiple wireless clients (Cisco router, TP-Link access point, Netgear extender, and D-Link client) to the same Wi-Fi network.
● Begin packet capture on your Wi-Fi interface for a few minutes.
● Stop capture and locate MAC addresses from captured frames.
● Wireshark automatically decodes the OUI (Organizationally Unique Identifier) in the packet details pane and displays the device manufacturer (e.g., Cisco Systems, TP-Link Technologies, Netgear Inc., D-Link Corporation).
● Verify the D-Link device manufacturer details (OUI example: 00:1C:F0 → D-Link Corporation).
● Note these results for documentation.

Examine Network Packets Using Wireshark:
● Apply filters to focus on specific devices, for example:
○ wlan.addr == <D-Link_MAC> → Filter traffic from the D-Link client.
○ ip.addr == <suspicious_IP> → Isolate specific communication.
● Review protocol details (DNS, HTTP, ICMP, TCP, UDP, ARP).
● Check for unusual patterns such as:
○ Repeated DNS lookups to unknown domains
○ High data transfer volume
○ Attempts to connect to suspicious IPs
● Export relevant packets as a .pcap file for further forensic analysis.
Analyze and Verify Findings:
● Summarize all captured data (source/destination IPs, ports, protocols).
● Compare OUI data to confirm manufacturer identities.
● Identify any suspicious or unauthorized communication by the D-Link device.
● Use hash values (MD5/SHA1) to verify the integrity of the packet capture.
Documentation:
● Include screenshots of:
○ CLI command outputs
○ Wireshark OUI lookup (highlighting Cisco, TP-Link, Netgear, D-Link)
○ Packet capture showing D-Link traffic
● Record all findings clearly in your lab report.

11.2 Lab Outcome :

● To learn how to identify and analyze wireless network devices using OUI lookup and packet analysis in Wireshark. You will also gain hands-on experience in recognizing manufacturer details (e.g., Cisco, TP-Link, Netgear, D-Link) and detecting suspicious communication using forensic techniques.

11.3Learning Objectives:

● To understand how network forensics helps in identifying and tracking wireless devices.
● To use CLI tools for analyzing network connections and IP configurations.
● To perform OUI lookups in Wireshark to determine device manufacturers.
● To capture and analyze wireless packets for evidence collection.
● To interpret findings and verify data authenticity with hash functions.

11.4 Requirement:

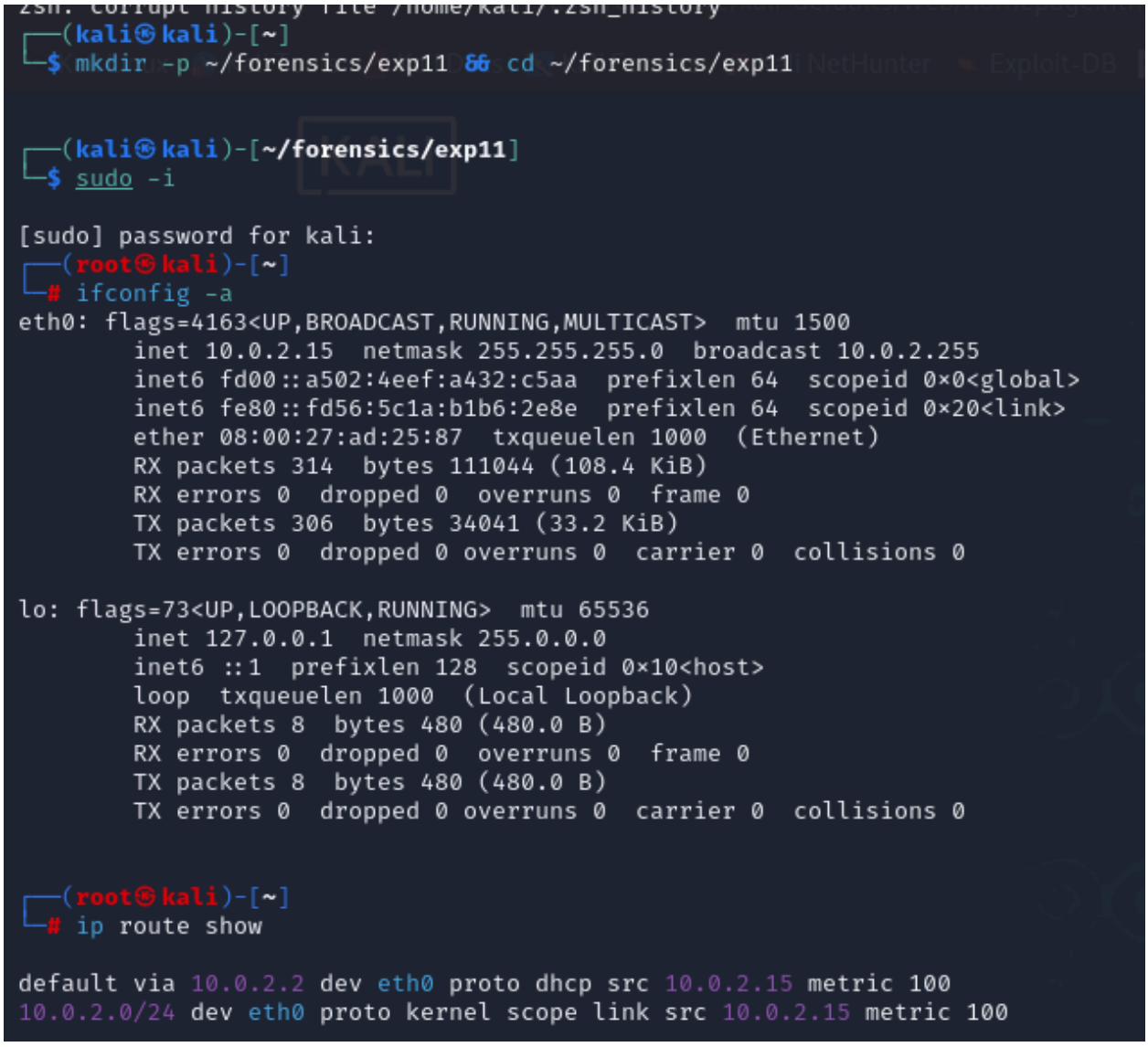
Hardware:
● Forensic workstation or laptop
● Wireless network setup (Cisco / TP-Link / Netgear devices)
● D-Link wireless client (example device)
● Internet connection
Software Tools:
● Command Prompt / Terminal – for CLI utilities
● Wireshark – for OUI lookup and packet analysis
● Hash calculator (MD5/SHA1) – for verifying evidence integrity
● Text editor or lab report template – for documentation

11.5 Related Theory :
Network Forensics:
Network forensics focuses on capturing and analyzing traffic data to detect and investigate cyber incidents. It helps identify the source and nature of unauthorized activities.
CLI Utilities:
Basic command-line tools provide quick insights into the network:
● ip config / ifconfig – shows IP and MAC address information.
● netstat – lists active connections and ports.
● arp -a – displays IP-to-MAC mappings.
● tracert / traceroute – shows the route taken by data packets.

Wireshark automatically resolves these OUIs and displays the vendor name in the packet details pane, helping investigators link network traffic to specific device brands.
Packet Analysis (Wireshark):
Wireshark captures all network packets and allows filtering by IP, MAC, or protocol. It helps detect malicious activity such as:
● Data exfiltration
● Unauthorized access
● Malware communication
● Network scanning attempts

Hashing and Chain of Custody:
When saving packet captures, hash values (MD5/SHA1) are generated to ensure the data has not been modified. This helps maintain forensic integrity and admissibility in investigations.

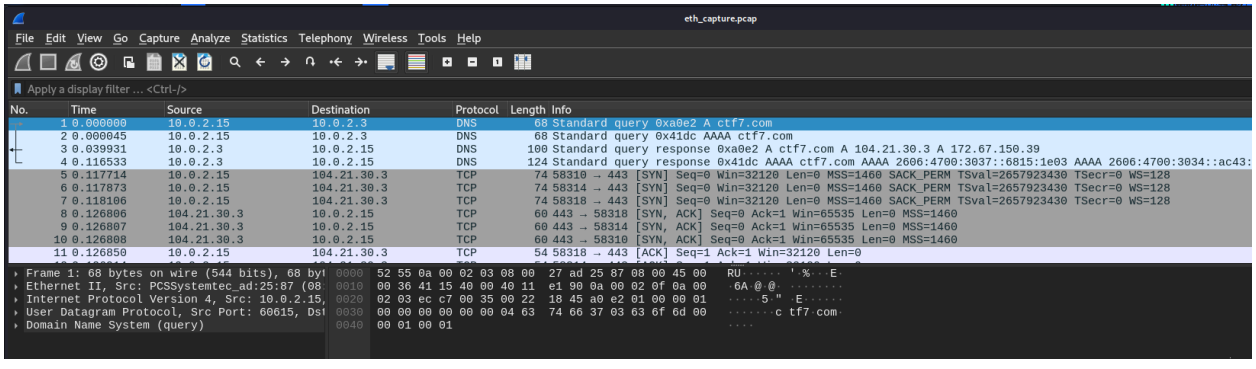
11.6 Output:


zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ mkdir -p ~/forensics/exp11 && cd ~/forensics/exp11
┌──(kali㉿kali)-[~/forensics/exp11]
└─$ sudo -i
[sudo] password for kali:
┌──(root㉿kali)-[~]
└─# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fd00::a502:4eef:a432:c5aa prefixlen 64 scopeid 0x0<global>
inet6 fe80::fd56:5c1a:b1b6:2e8e prefixlen 64 scopeid 0x20<link>
ether 08:00:27:ad:25:87 txqueuelen 1000 (Ethernet)
RX packets 314 bytes 111044 (108.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 306 bytes 34041 (33.2 KiB)
TX errors 0 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 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 8 bytes 480 (480.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8 bytes 480 (480.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
┌──(root㉿kali)-[~]
└─# ip route show
default via 10.0.2.2 dev eth0 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15 metric 100

```
┌──(root㉿kali)-[~]
└─# arp -a
m
? (10.0.2.3) at 52:55:0a:00:02:03 [ether] on eth0
? (10.0.2.2) at 52:55:0a:00:02:02 [ether] on eth0
m: command not found

┌──(root㉿kali)-[~]
└─# ss -tunap
Netid               State               Recv-Q              Send-Q                           Local Address:Port                           Peer Address:Port
udp                 ESTAB               0                   0                          10.0.2.15%eth0:68                            10.0.2.2:67
tcp                 LISTEN              0                   128                              0.0.0.0:22                                  0.0.0.0:*
tcp                 ESTAB               0                   0                             10.0.2.15:39212                         34.107.243.93:443
tcp                 ESTAB               0                   0                             10.0.2.15:58488                         34.36.137.203:443
tcp                 LISTEN              0                   128                                 [::]:22                                     [::]:*

┌──(root㉿kali)-[~]
└─# traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  10.0.2.2 (10.0.2.2)  0.716 ms  0.692 ms  0.365 ms
```

```
┌──(root㉿kali)-[~]
└─# sudo tcpdump -i eth0 -w ~/forensics/eth_capture.pcap

tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C3493 packets captured
3493 packets received by filter
0 packets dropped by kernel
```

```
┌──(root㉿kali)-[~]
└─# ls ~/forensics/

eth_capture.pcap

┌──(root㉿kali)-[~]
└─# wireshark ~/forensics/eth_capture.pcap

 ** (wireshark:10966) 10:11:17.642221 [GUI WARNING] -- QStandardPaths: XDG_RUNTI
```



```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ capinfos ~/forensics/test_capture.pcap

File name:           /home/kali/forensics/test_capture.pcap
File type:           Wireshark/tcpdump/ ...  - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 262144 bytes
Number of packets:   3
File size:           292 bytes
Data size:           220 bytes
Capture duration:    0.005337 seconds
First packet time:   2025-10-19 10:23:40.408943
Last packet time:    2025-10-19 10:23:40.414280
Data byte rate:      41 kBps
Data bit rate:       329 kbps
Average packet size: 73.33 bytes
Average packet rate: 562 packets/s
SHA256:              6bbfc060c0ae6dc5817d1dcd28f1b6adf67236dcc9cabf0956cd651b201f1089
SHA1:                18fda44b99e52ca6f745a6904105dbeb3adb0e3f
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 262144
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 3
```

```
┌──(kali㉿kali)-[~]
└─$ tshark -r ~/forensics/test_capture.pcap -T fields -e eth.src -e eth.dst | tr '\t' '\n' | grep -v '^$' | sort -u

08:00:27:ad:25:87
52:55:0a:00:02:02
```

```
┌──(kali㉿kali)-[~]
└─$ md5sum ~/forensics/test_capture.pcap
sha1sum ~/forensics/test_capture.pcap

0317b3e2d6ca1fbf82164acf4abb18b3  /home/kali/forensics/test_capture.pcap
18fda44b99e52ca6f745a6904105dbeb3adb0e3f  /home/kali/forensics/test_capture.pcap
```

11.7 Conclusion:

In this experiment, network forensics techniques were applied to identify and analyze devices on a wireless network. Using CLI utilities and packet captures, all connected devices were enumerated, and the D-Link client was isolated for deeper analysis. OUI lookup in Wireshark confirmed the manufacturer of each device, and filtered packet inspection revealed potential suspicious activity, such as repeated DNS queries and unusual outbound connections. The integrity of all captured evidence was verified using MD5 and SHA1 hashes, demonstrating the importance of forensic procedures in monitoring network security and identifying unauthorized or abnormal behavior.