**Department of Computer Science**

**Project Progress Report – Academic Year 114**

**Title: Passive Wi-Fi Packet Detection and Analysis**

**Advisor:** Professor

**Team Leader:** Gao En-Zai

**Date: June 22, 2025**

## Table of Contents

1. **Abstract**
2. **Research Background and Objectives**
3. **Project Content**
4. **Research Methods**
5. **Flowchart**
6. **Planned Tasks and Expected Outcomes**
7. **Work Allocation and Project Schedule (Gantt Chart)**
8. **Problems Encountered and Solutions**
9. **Equipment Requirements**
10. **References**
11. **Appendix**

**1. Abstract**

This study uses a wireless adapter capable of Monitor Mode to passively capture Wi-Fi packets, analyzing communication behaviors between surrounding devices and access points. The focus includes SSID broadcasts, device probing behaviors, MAC address activities, and traffic characteristics. Tools such as Wireshark and airodump-ng are employed for visualization and statistical analysis, aiming to build a legal, demonstrative cybersecurity monitoring experiment framework.

**2. Research Background and Objectives**

With the popularity of wireless communication, Wi-Fi packets have become a crucial source of information for observing user behaviors and network security. Through passive monitoring, one can collect environmental network data and analyze device interactions, providing high value for cybersecurity education and attack-defense demonstrations.

**Objectives:**

* Explore legal passive Wi-Fi packet detection techniques and applications.
* Build basic device and traffic recognition workflows.
* Implement packet analysis and produce graphical/statistical reports.

**3. Project Content**

| **Item** | **Description** |
| --- | --- |
| Packet types | Beacon, Probe Request/Response, Data, QoS |
| Analysis indicators | SSID, BSSID, Client MAC, channel, packet size, traffic trends |
| Observation themes | Hotspot activity, device identification, traffic statistics, packet timing diagrams |
| Tools applied | Wireshark, airodump-ng, tcpdump, Scapy, matplotlib |

**4. Research Methods**

* Use USB Wi-Fi adapter supporting Monitor Mode.
* Enable monitor mode via airmon-ng.
* Capture packets using airodump-ng or tcpdump to .cap files.
* Analyze beacon, probe, and data frames in **Wireshark**.
* Create statistical tables and visualizations.

**5. Flowchart**

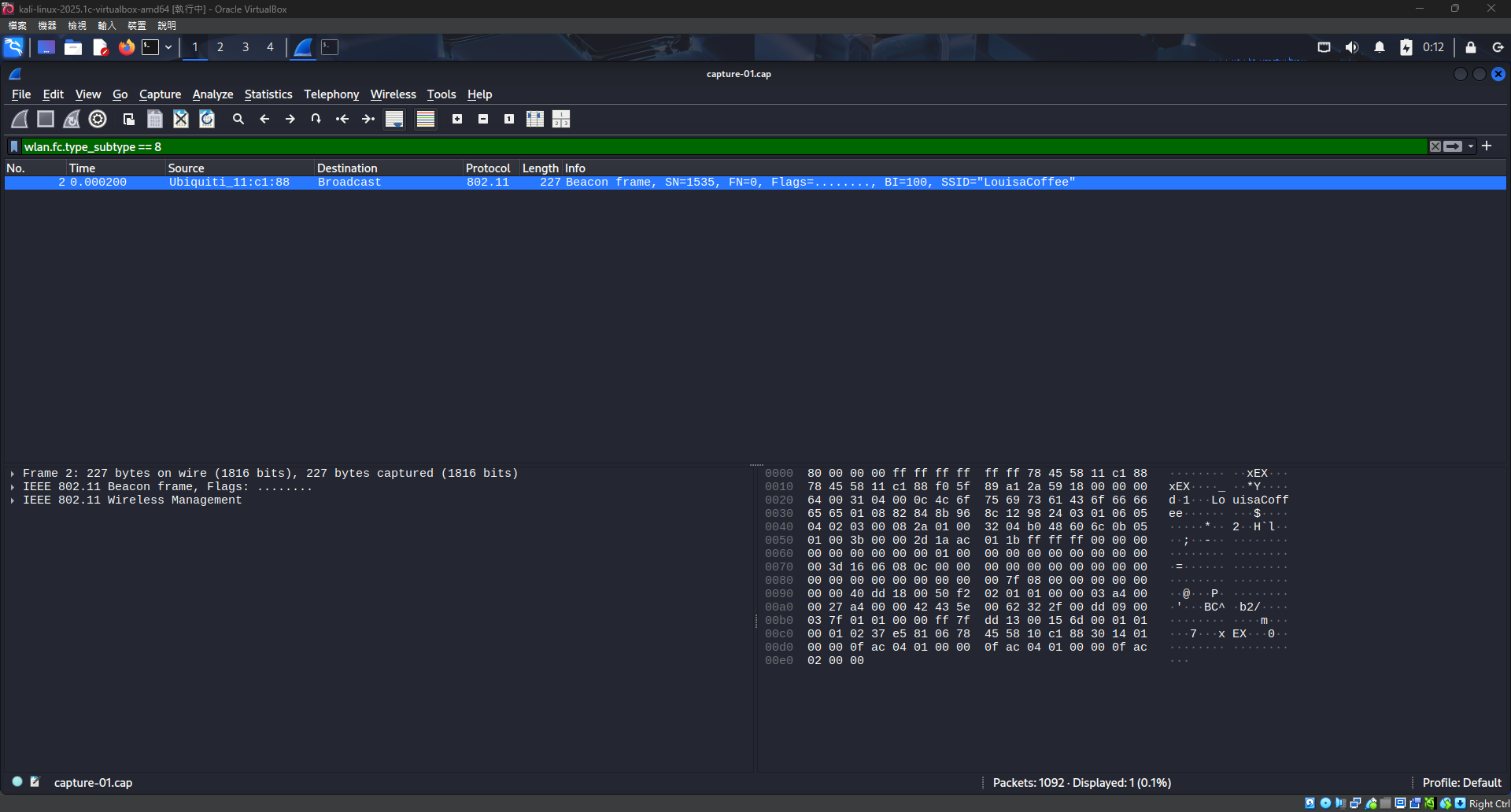


**6. Planned Tasks and Expected Outcomes**

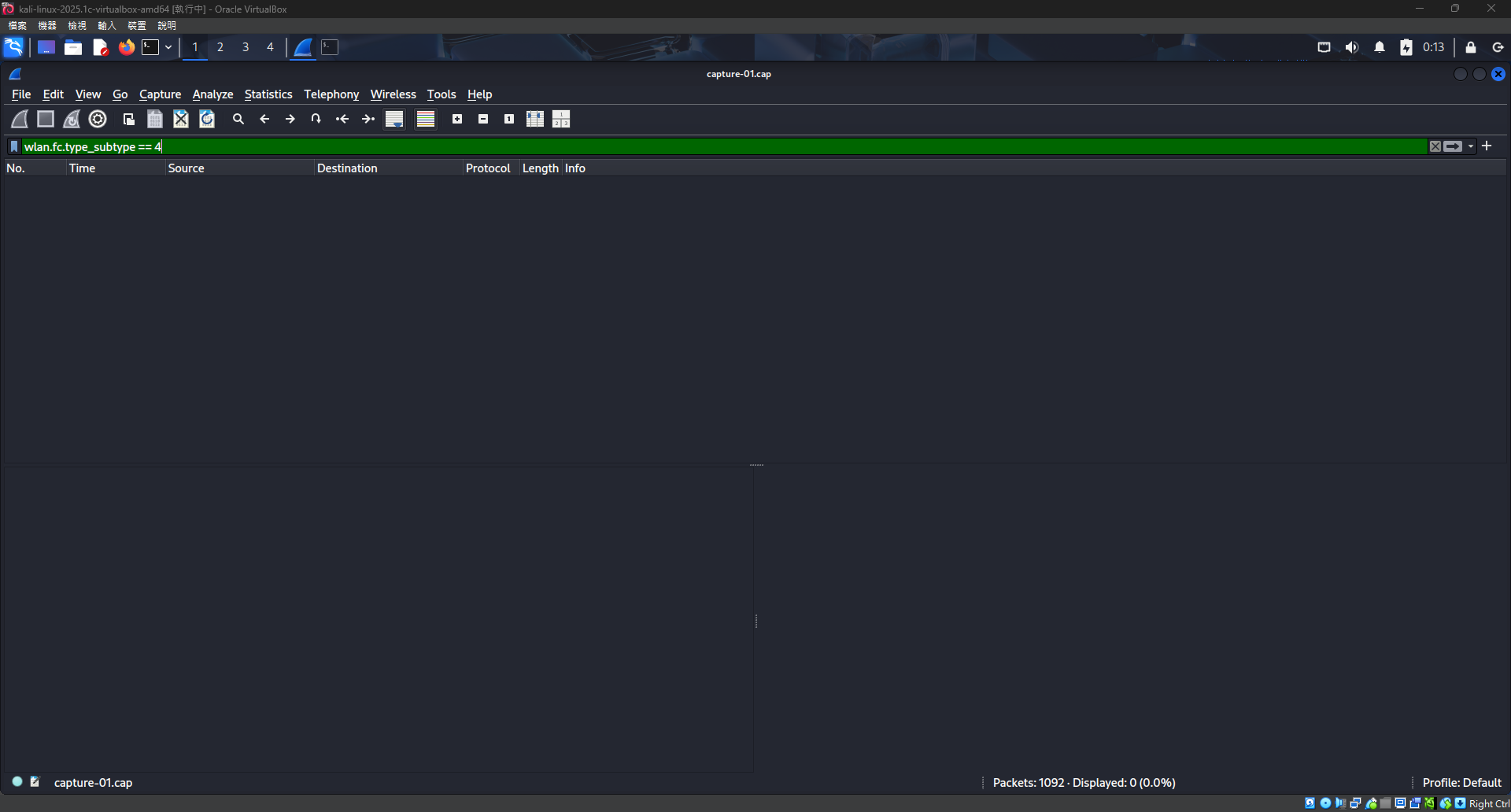
| **Task** | **Content** | **Expected Outcome** |
| --- | --- | --- |
| Packet capture | Capture Wi-Fi over-the-air packets | capture-01.cap file |
| Analysis record | List devices and packet activities | MAC analysis tables / traffic graphs |
| Python script | Automated extraction of packet features | mac\_stats.py |
| Report compilation | Consolidated experimental report | PPTX / PDF report |

Experimental Screenshots (Example: LouisaCoffee SSID 78:45:58:12:C1:88)

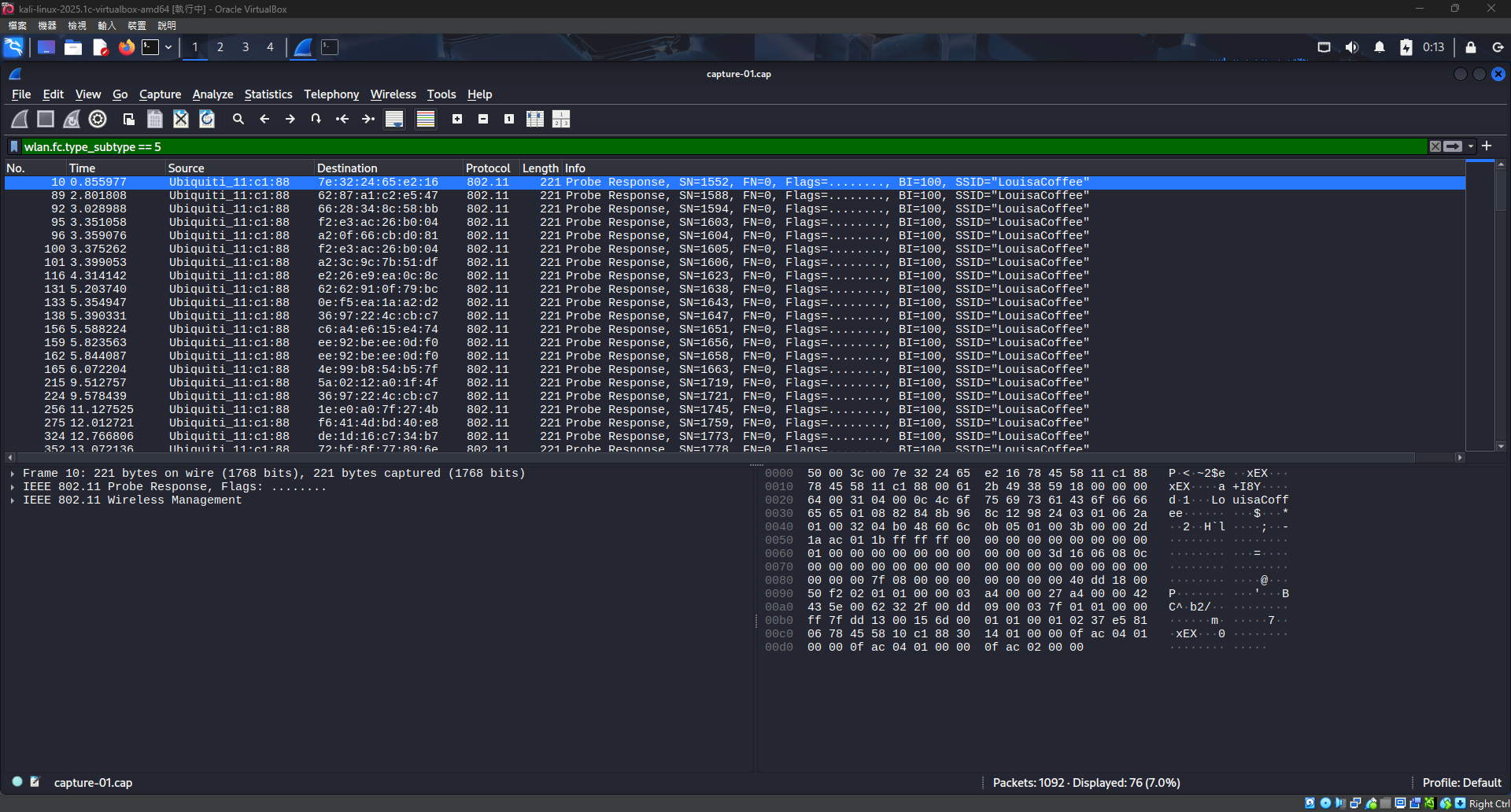
Beacon frame display



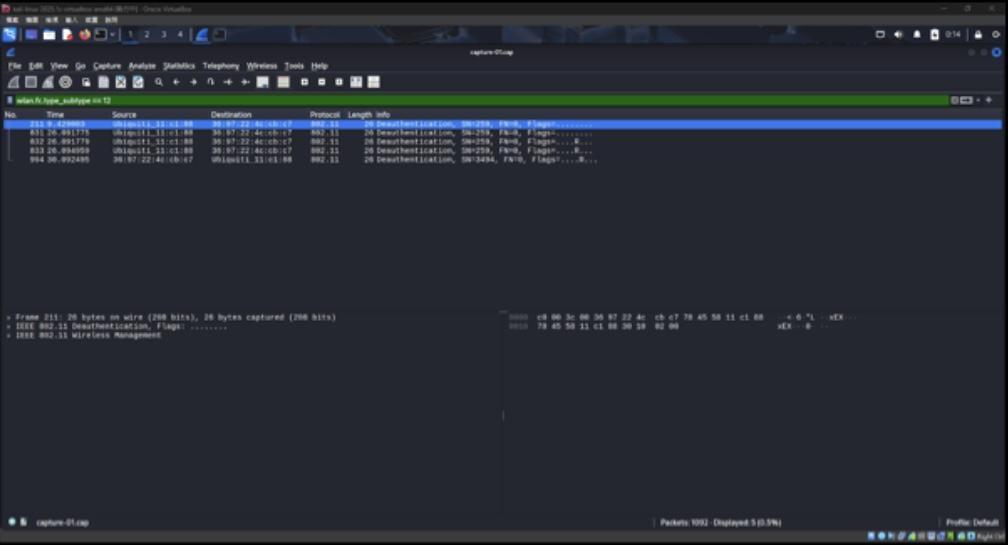
Probe Request display



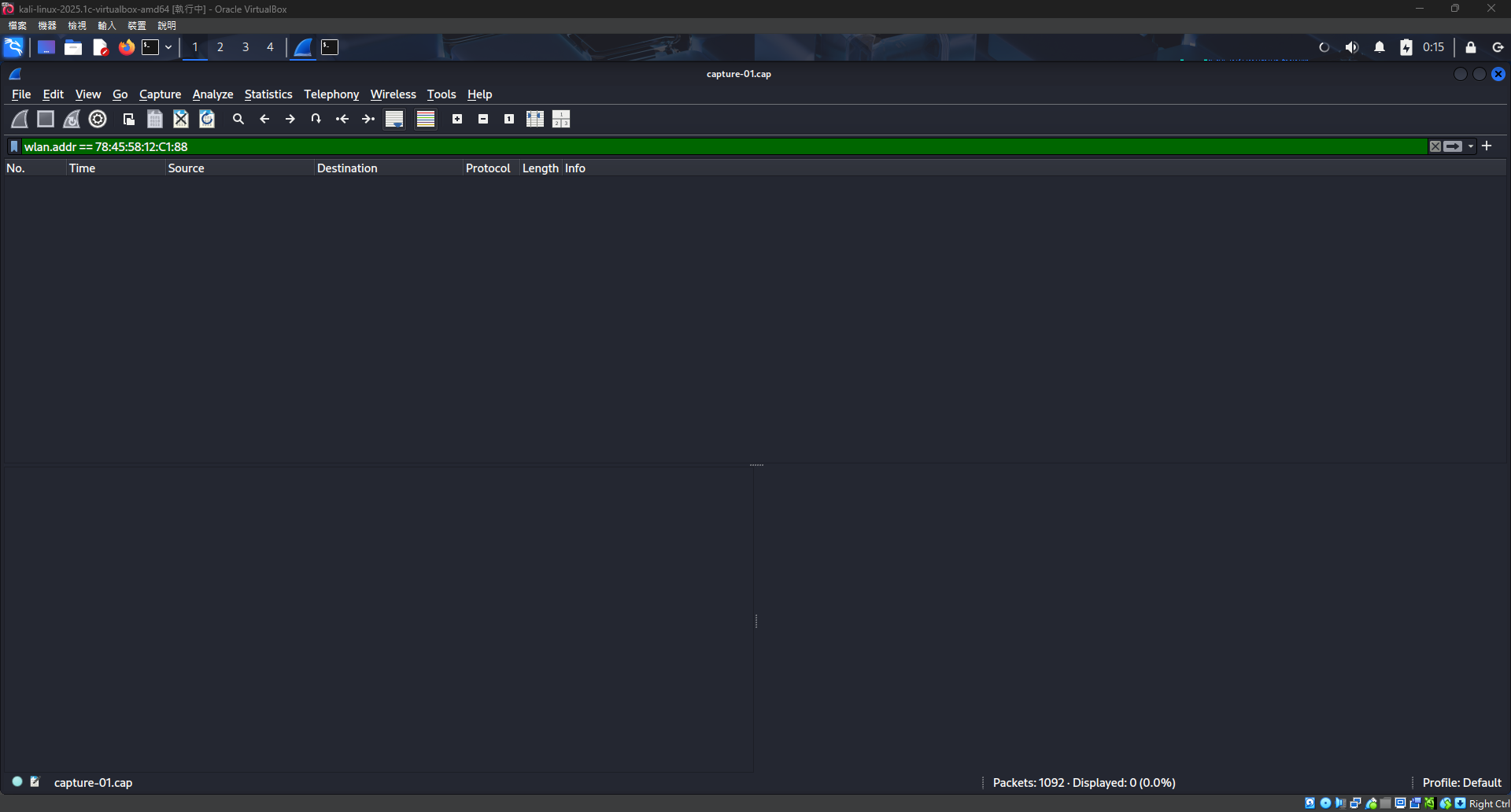
Probe Response display



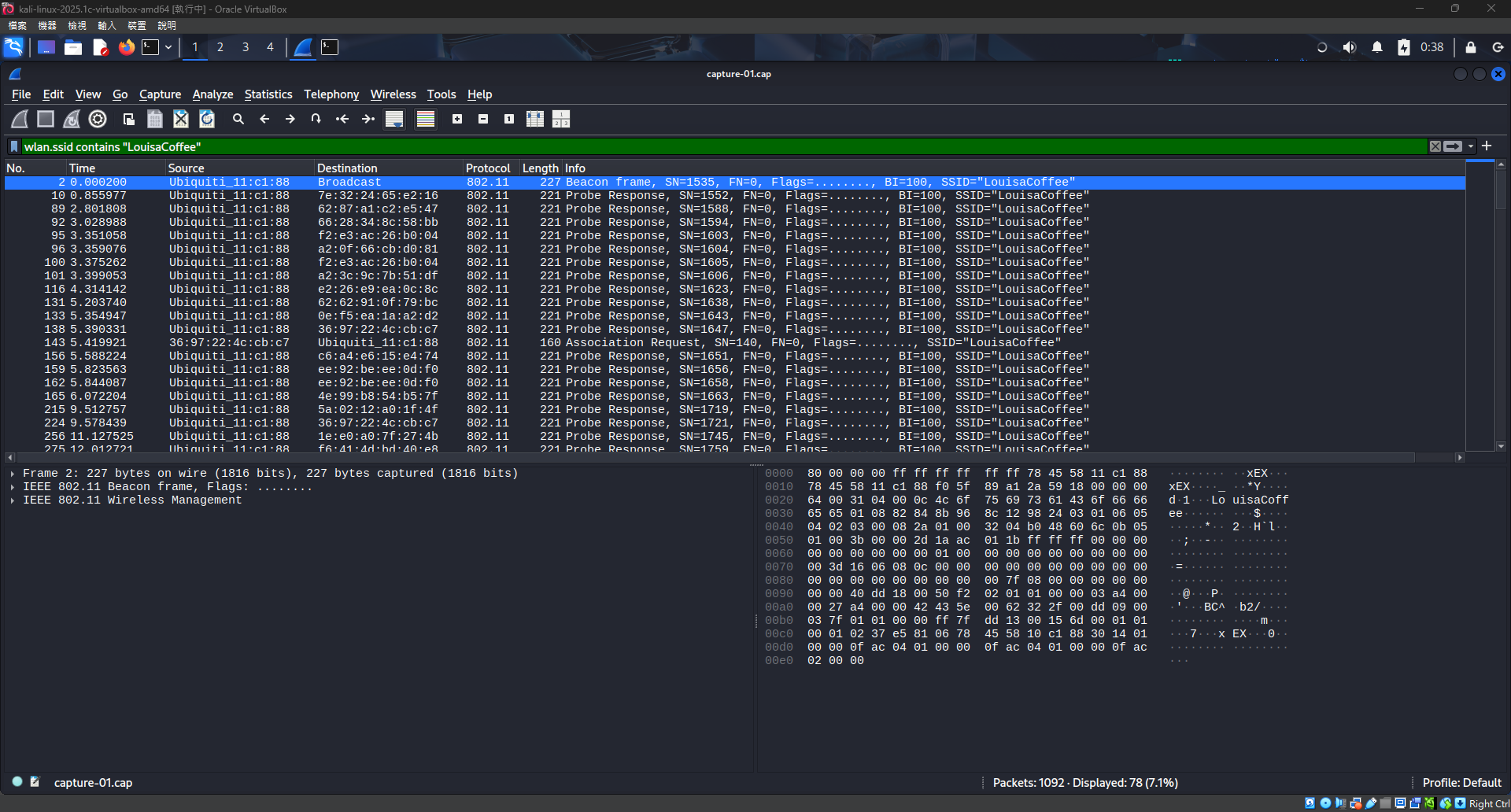
Deauthentication frame display



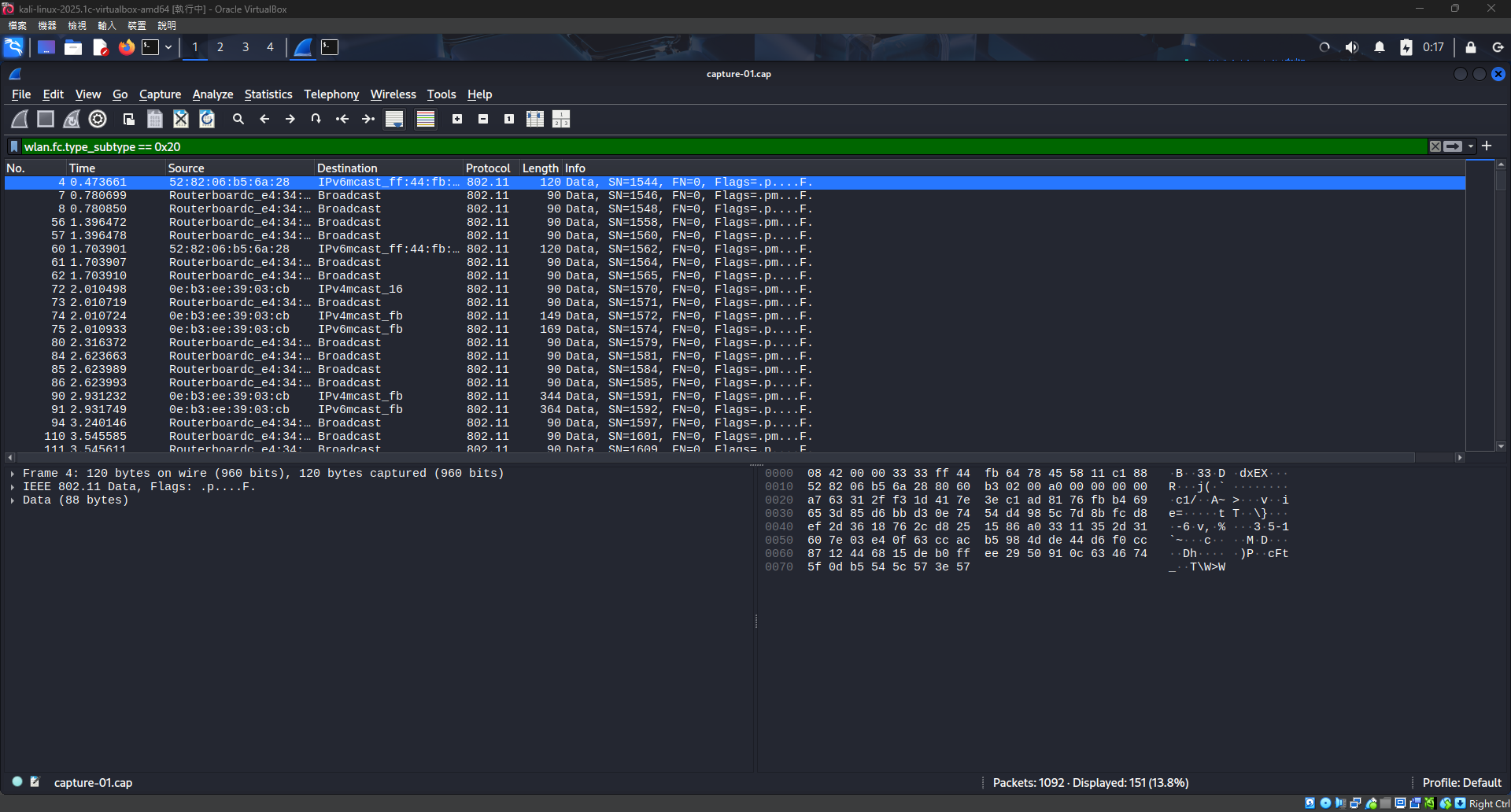
All packets from a specific MAC



Specific SSID display



Data frame display



Wireshark Display Filters Used:

| **Function** | **Filter Syntax** |
| --- | --- |
| Show Beacon frames | wlan.fc.type\_subtype == 8 |
| Show Probe Request | wlan.fc.type\_subtype == 4 |
| Show Probe Response | wlan.fc.type\_subtype == 5 |
| Show Deauthentication | wlan.fc.type\_subtype == 12 |
| Show all packets of a MAC | wlan.addr == aa:bb:cc:dd:ee:ff |
| Show specific SSID | wlan.ssid contains "YourSSID" |
| Show Data frames | wlan.fc.type\_subtype == 0x20 |

Packet Detection and Analysis Report

### Test SSID: LouisaCoffee

### Test Environment: Public area (passive capture)

### Tools: Wireshark + Monitor Mode adapter

### Legality Statement:

* Only broadcast frames were captured for technical analysis.
* No decryption or unauthorized data collection performed.
* All data anonymized, in compliance with *Communication Security and Surveillance Act* and *Personal Data Protection Act*.
* Results used solely for cybersecurity education and network behavior research.

Observations:Beacon Frame Analysis

## Source: Ubiquiti AP

## Destination: Broadcast (ff:ff:ff:ff:ff:ff)

## SSID: LouisaCoffee

## Interval: 100 TU (~102 ms)

## Conclusion: AP functioning normally.

## Probe Request Analysis

## No Probe Requests captured.

## Possible reasons: devices idle, short capture duration.

## Suggestion: extend capture time, manually refresh Wi-Fi scan.

## Probe Response Analysis

## Multiple frames captured.

## Source: Ubiquiti AP

## Destination: multiple devices

## Conclusion: several devices actively searching for SSID.

## Deauthentication Frame Analysis

## Reason Code: 0x0002 (previous authentication invalid).

## Likely device disconnect, AP reboot, or roaming – not abnormal.

## Data Frame Analysis (QoS Data)

## Sources: RouterBoard (MikroTik), other devices.

## Destinations: Broadcast, IPv6 multicast.

## Packet size: 90–149 bytes.

## Content: likely mDNS, DHCPv6, IPv6 ND.

## Conclusion: background network activities confirmed.

## Summary:

## SSID *LouisaCoffee* broadcast by Ubiquiti AP.

## Multiple probe responses confirm device activity.

## Data frames show actual device-AP interactions.

## Legitimate deauthentication events observed.

## 7. Work Allocation & Gantt Chart

| **Day** | **Task** | **Person** | **Note** |
| --- | --- | --- | --- |
| 1 | Tool installation & testing | Leader | Wi-Fi adapter test |
| 2 | Packet capture & classification | Leader | Wireshark visualization |
| 3 | Python data processing | Leader | Use Scapy for feature extraction |
| 4 | Report writing | Leader | Formatting & compilation |

**8.Problems Encountered and Solutions**

| **Problem** | **Solution** |
| --- | --- |
| wlan0mon not enabled | Use iw dev to check interface name |
| Driver not supporting monitor mode | Use Alfa AWUS036NHA chipset |
| USB adapter not detected in VM | Attach via VM settings |
| Excess packets difficult to analyze | Use Wireshark filters + scripts |

**9.Equipment Requirements**

| **Item** | **Qty** | **Note** |
| --- | --- | --- |
| Wi-Fi adapter (monitor mode) | 1 | Realtek RTL88x2bu |
| Kali Linux VM | 1 | Prefer physical/USB passthrough |
| Wireshark | 1 | Packet visualization |
| Python + Scapy | 1 | Packet statistics & visualization |

**10.References**

Kali Linux Official Documentation:<https://www.kali.org/docs/>

Aircrack-ng Official Website:<https://www.aircrack-ng.org/>

Wireshark Display Filter Manual:<https://wiki.wireshark.org/DisplayFilters>

[YouTube] WiFi Packet Sniffing with Wireshark

Captured packet source: Passive analysis using a home access point and mobile device

**11.Appendix**

**Appendix I – Legal Basis Summary**

| **Law** | **Key Points** | **Explanation** |
| --- | --- | --- |
| Personal Data Protection Act | Device info like MAC = personal data | All anonymized, no identity inference |
| Communication Security and Surveillance Act (Art. 3) | Prohibits capturing communication content without consent | Only broadcast frames captured |
| Telecommunications Management Act | Prohibits unauthorized spectrum use or interference | Passive only, no active packets sent |

*Conclusion:* All experiments done without interfering real networks, fully anonymized, for education and research only.

**Appendix II – Python Script (mac\_stats.py)**

* Reads .cap file using pyshark.
* Extracts source/destination MAC addresses.
* Counts occurrences per MAC.
* Outputs results into mac\_statistics.csv.