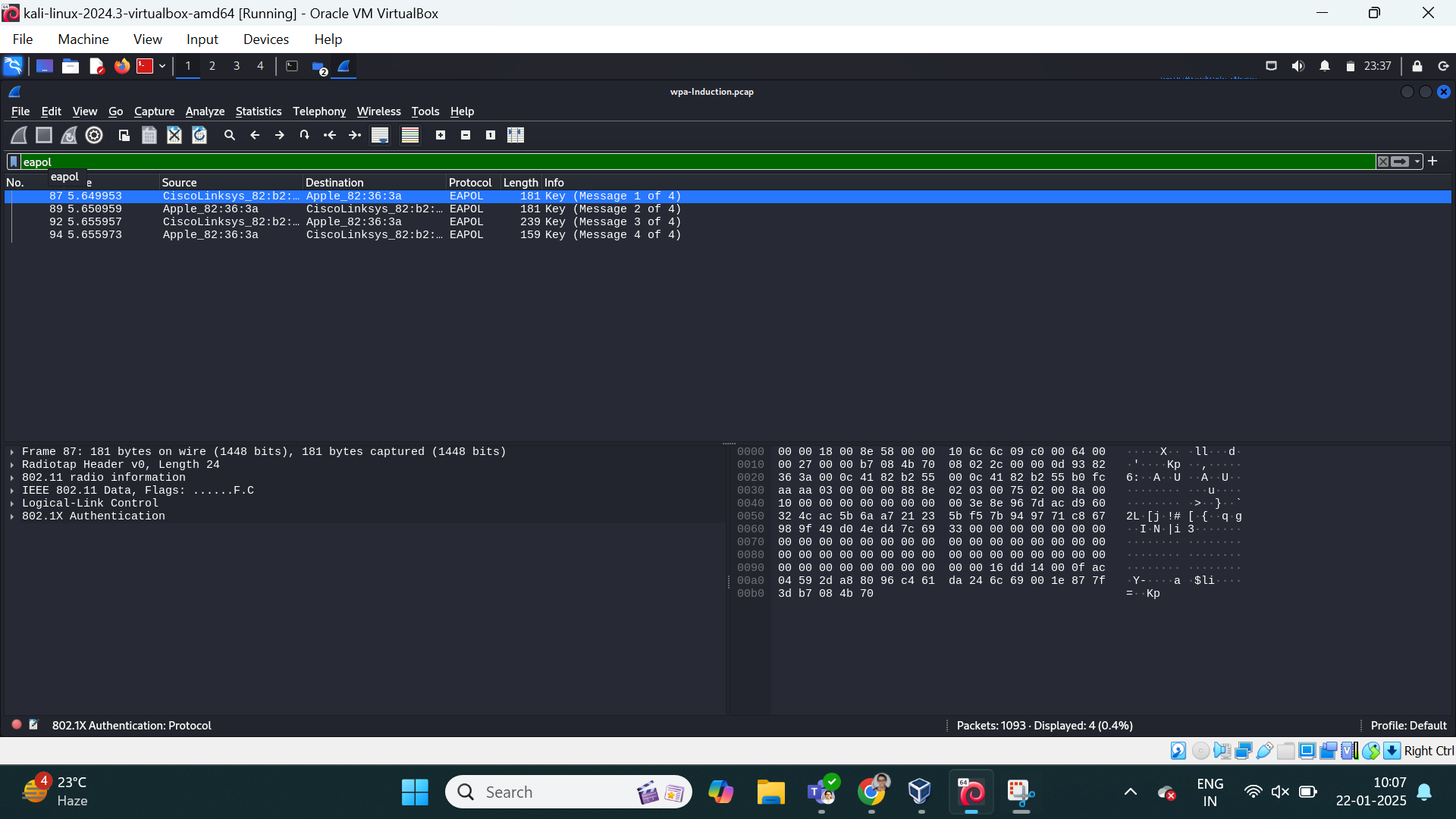
## \*\*Lab Exercises\*\*

### \*\*Exercise 1: Analyzing Wireless Traffic Using Wireshark\*\*

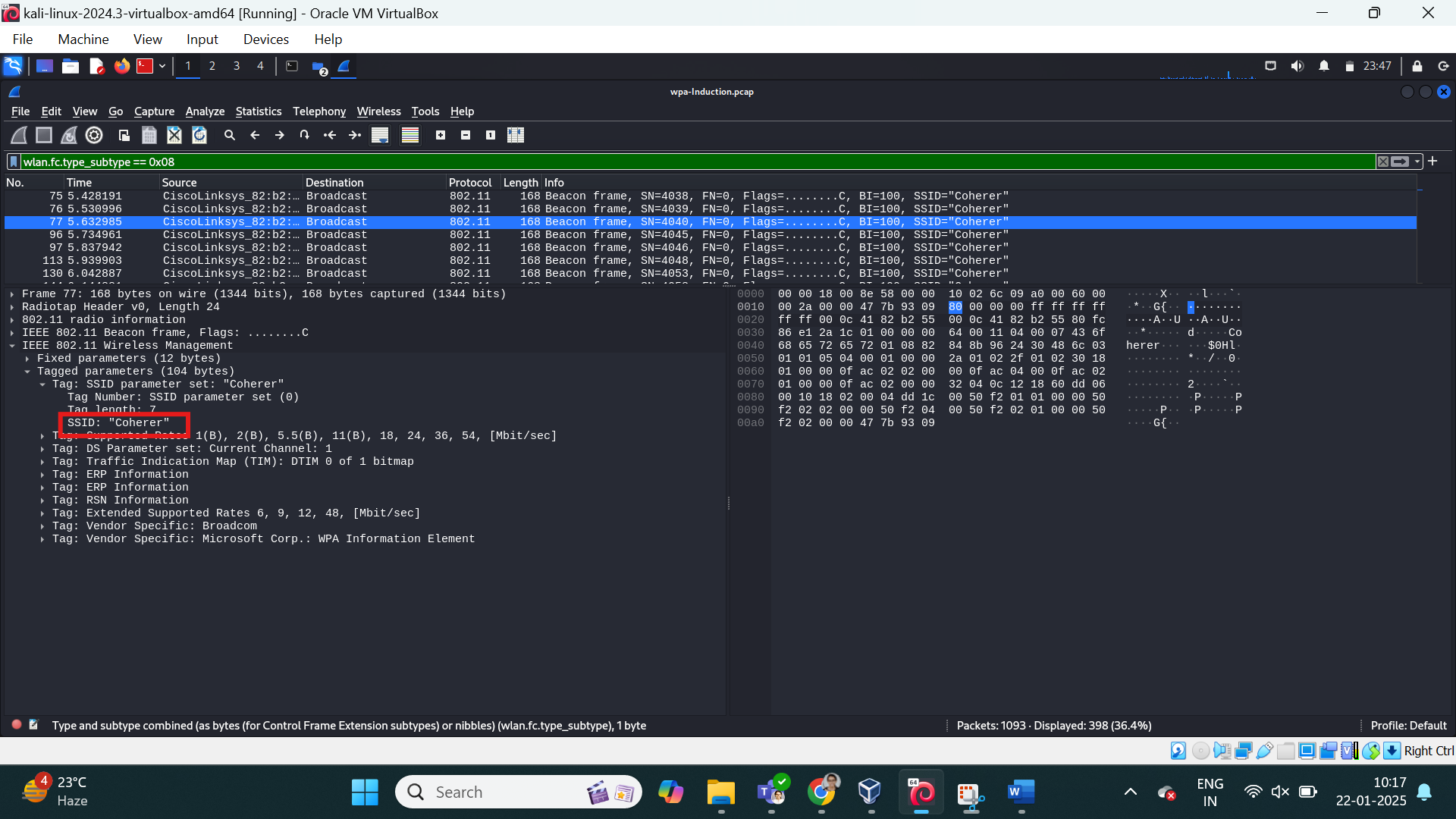
#### \*\*Objective\*\*:  
Learn how to open and analyze pre-captured wireless network traffic.

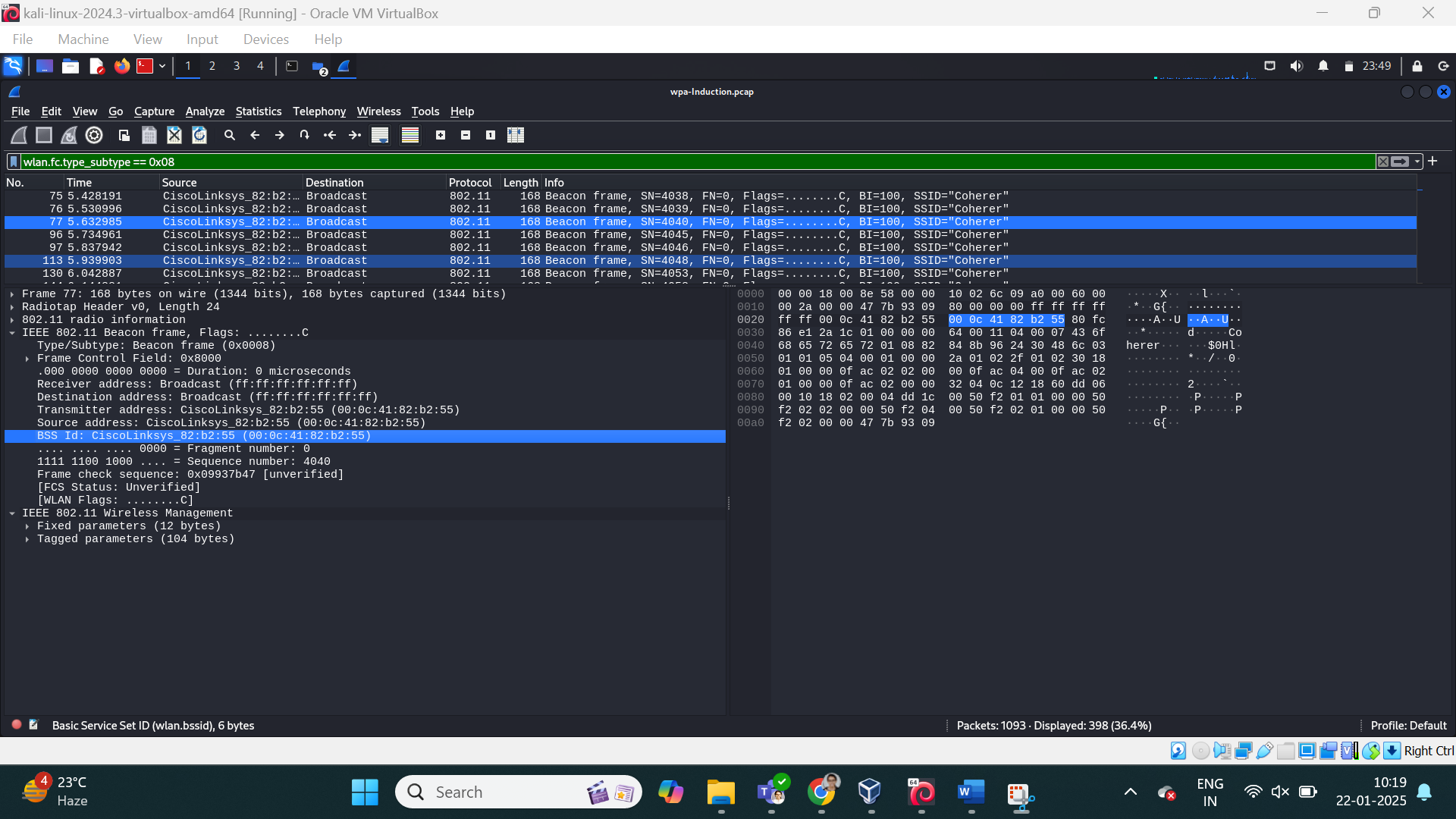
#### \*\*Steps\*\*:  
1. \*\*Open Wireshark\*\*:  
   - Launch Wireshark on your computer.  
   - Load a pre-captured file (e.g., WPA handshake):  
     - Go to `File` > `Open` > Select the `.pcap` file.

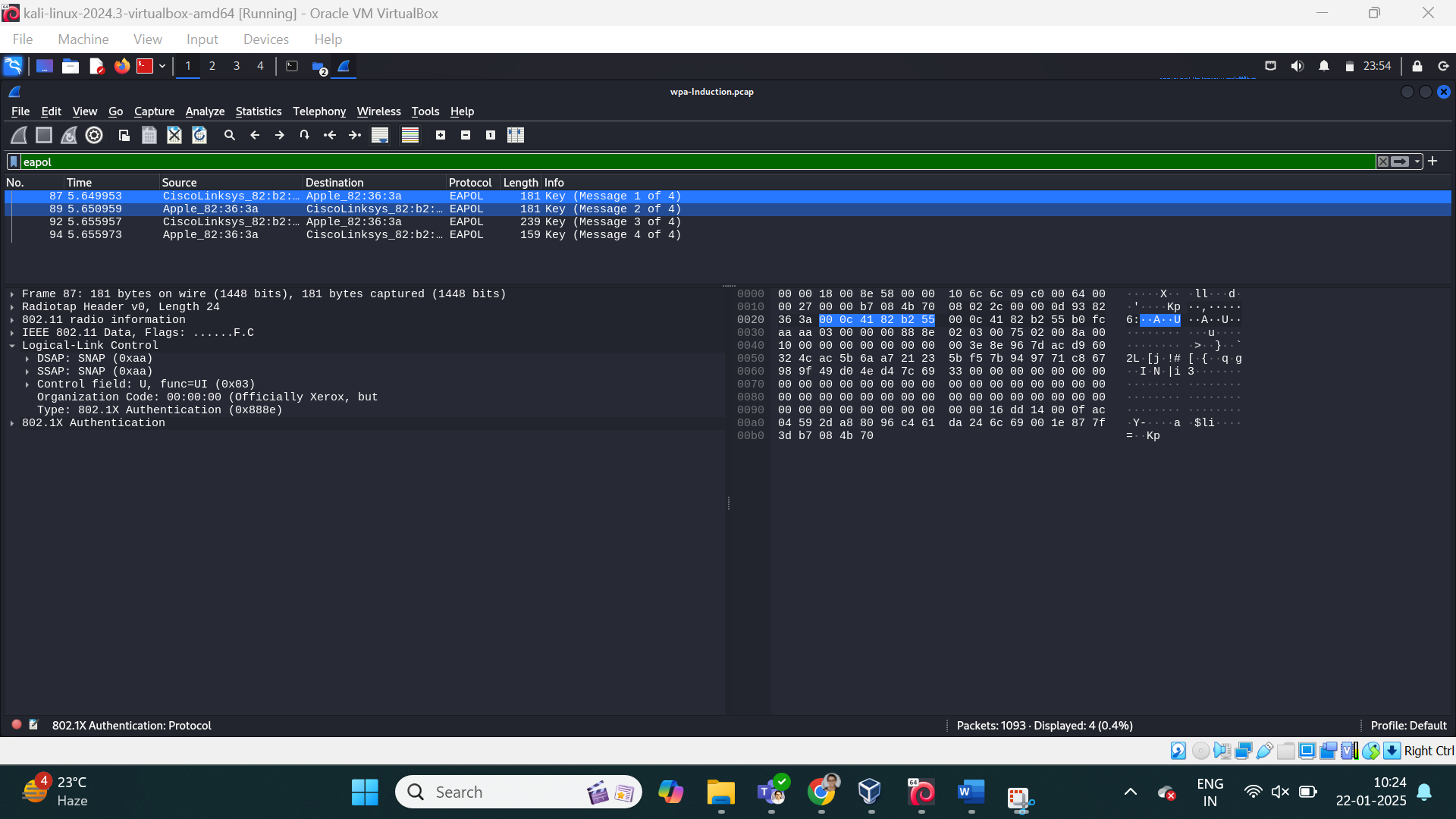
2. \*\*Filter Wireless Traffic\*\*:  
   - Use filters to isolate wireless packets:  
     - For WPA handshake: `eapol`.



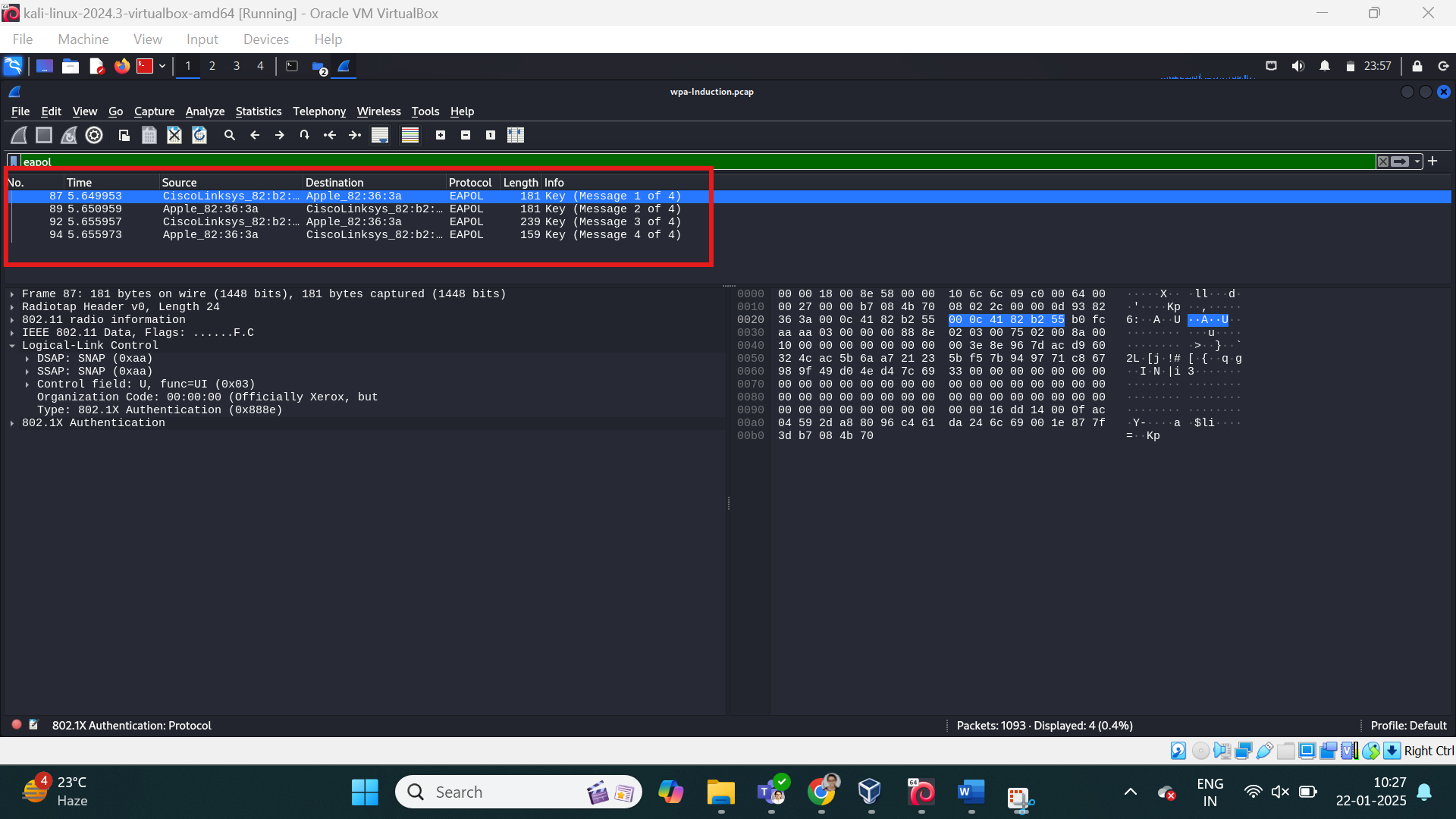
3. \*\*Analyze Packets\*\*:  
   - Identify the following components in the capture:  
     - \*\*SSID\*\*: The network name.

  
     - \*\*BSSID\*\*: The MAC address of the access point.

  
     - \*\*Handshake Packets\*\*: Look for EAPOL packets.



4. \*\*Document Observations\*\*:  
   - Note the time, source, destination, and protocol of key packets.



---

### \*\*Exercise 2: Cracking WEP Encryption Using Pre-Captured Files\*\*

#### \*\*Objective\*\*:  
Simulate cracking a WEP-encrypted network using Aircrack-ng.

#### \*\*Steps\*\*:  
1. \*\*Download a WEP Capture File\*\*:  
   - Obtain a `.cap` file containing WEP packets (e.g., from GitHub).

2. \*\*Analyze the Capture\*\*:  
   - Open the file in Wireshark to ensure it contains WEP-encrypted traffic.

3. \*\*Use Aircrack-ng\*\*:  
   - Launch Kali Linux.  
   - Run Aircrack-ng on the capture file:  
     ```bash  
     aircrack-ng -b [BSSID] -w /usr/share/wordlists/rockyou.txt wep\_capture.cap  
     ```

4. \*\*Crack the Key\*\*:  
   - If successful, Aircrack-ng will display the WEP key in plaintext.

#### \*\*Expected Outcome\*\*:  
Understand how WEP encryption is vulnerable and why it is deprecated.

---

### \*\*Exercise 3: Cracking WPA/WPA2 Encryption Using a Pre-Captured Handshake\*\*

#### \*\*Objective\*\*:  
Simulate cracking a WPA/WPA2-encrypted network using a captured handshake and dictionary attack.

#### \*\*Steps\*\*:  
1. \*\*Download a WPA Handshake File\*\*:  
   - Obtain a `.cap` file with a valid handshake (e.g., from GitHub).

2. \*\*Verify the Handshake\*\*:  
   - Open the file in Wireshark.  
   - Use the filter `eapol` to ensure handshake packets are present.

3. \*\*Convert the Capture File\*\*:  
   - Use `aircrack-ng` to convert the `.cap` file to `.hccapx` format for Hashcat (optional):  
     ```bash  
     aircrack-ng handshake.cap -J handshake  
     ```

4. \*\*Run Aircrack-ng\*\*:  
   - Attempt to crack the handshake using a dictionary file:  
     ```bash  
     aircrack-ng -w /usr/share/wordlists/rockyou.txt handshake.cap  
     ```

5. \*\*Run Hashcat (Optional)\*\*:  
   - Use GPU acceleration for faster cracking:  
     ```bash  
     hashcat -m 2500 handshake.hccapx /usr/share/wordlists/rockyou.txt  
     ```

#### \*\*Expected Outcome\*\*:  
Understand the importance of strong passwords in securing WPA/WPA2 networks.

---

### \*\*Exercise 4: Simulating an Evil Twin Attack (Theoretical)\*\*

#### \*\*Objective\*\*:  
Understand the concept of a rogue access point and its impact on network security.

#### \*\*Steps\*\*:  
1. \*\*Setup Overview\*\*:  
   - Learn about tools like `Fluxion` or `Airgeddon` for setting up fake APs.  
   - Note: Without Wi-Fi hardware, this exercise remains theoretical.

2. \*\*Captive Portal\*\*:  
   - Understand how users are tricked into providing credentials via a rogue AP.

3. \*\*Defensive Measures\*\*:  
   - Emphasize the importance of HTTPS and VPNs in mitigating Evil Twin attacks.

#### \*\*Expected Outcome\*\*:  
Awareness of social engineering risks associated with rogue access points.

---