



RIPHAH
INTERNATIONAL UNIVERSITY

Wi-Fi Deauther Using NodeMCU ESP8266

Final Project Report

Submitted By:

Muhammad Husnain – 62056

Ameer Hamza – 63582

Mudansuru Auwalu Garba – 66292

Abubakar Ali – 62782

Syed Mujtaba Zaidi – 62081

Supervisor:

Hasnat Ali

Course:

Digital Logic Design

Submission Date:

May 26, 2025

Wi-Fi Deauther Project Report Using NodeMCU ESP8266

Table of Contents

1. Introduction
2. Team Members
3. Project Objectives
4. Background and Motivation
5. Literature Review
6. Tools and Components
7. Methodology 7.1 Firmware Selection and Flashing 7.2 Hardware Setup 7.3 Network Connection and Web Interface Access 7.4 Scanning and Target Selection 7.5 Execution of Deauthentication Attack
8. Implementation 8.1 Step-by-Step Process 8.2 Observations and Results
9. Technical Overview 9.1 802.11 Protocol and Deauthentication Frames 9.2 ESP8266 and Web Interface Functionality
10. Security Implications and Ethical Considerations
11. Logic Diagram
12. Challenges Faced
13. Conclusion
14. References

1. Introduction

This report documents the development and execution of a Wi-Fi Deauther project using the NodeMCU ESP8266 microcontroller. The project explores the vulnerabilities in Wi-Fi networks through the demonstration of a deauthentication (deauth) attack, using open-source firmware. This tool is intended strictly for educational and ethical use, aimed at raising awareness about wireless security weaknesses.

2. Team Members

- Muhammad Husnain (62056)
- Ameer Hamza (63582)
- Mudansuru Auwalu Garba (66292)
- Abubakar Ali (62782)
- Syed Mujtaba Zaidi (62081)

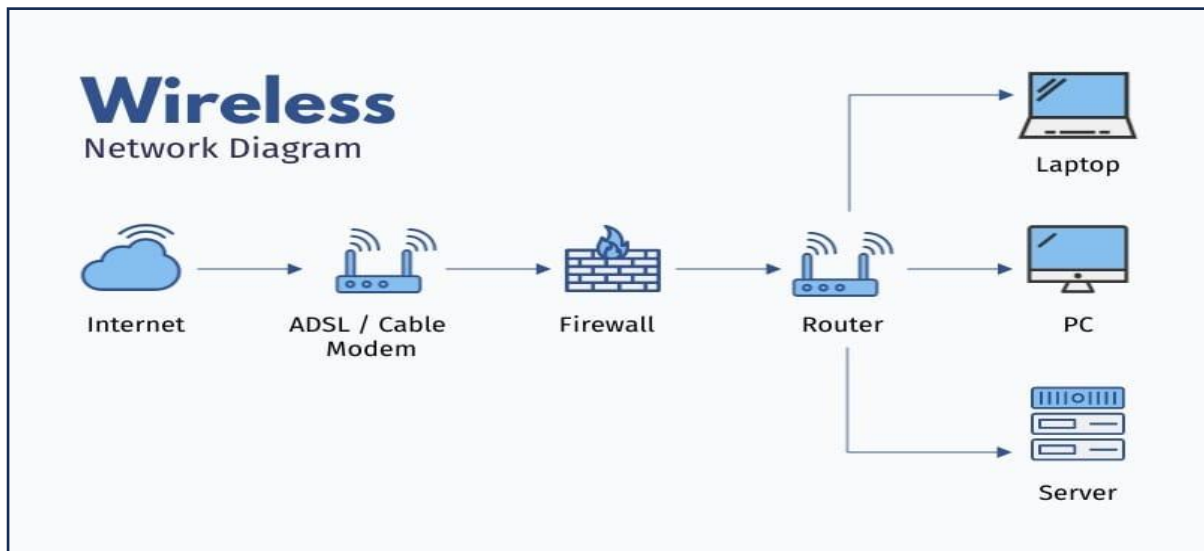
3. Project Objectives

- Develop a functional Wi-Fi Deauther using NodeMCU ESP8266
- Understand and demonstrate 802.11 deauthentication vulnerabilities
- Educate peers on Wi-Fi security threats and best practices
- Emphasize the importance of ethical use and legal constraints

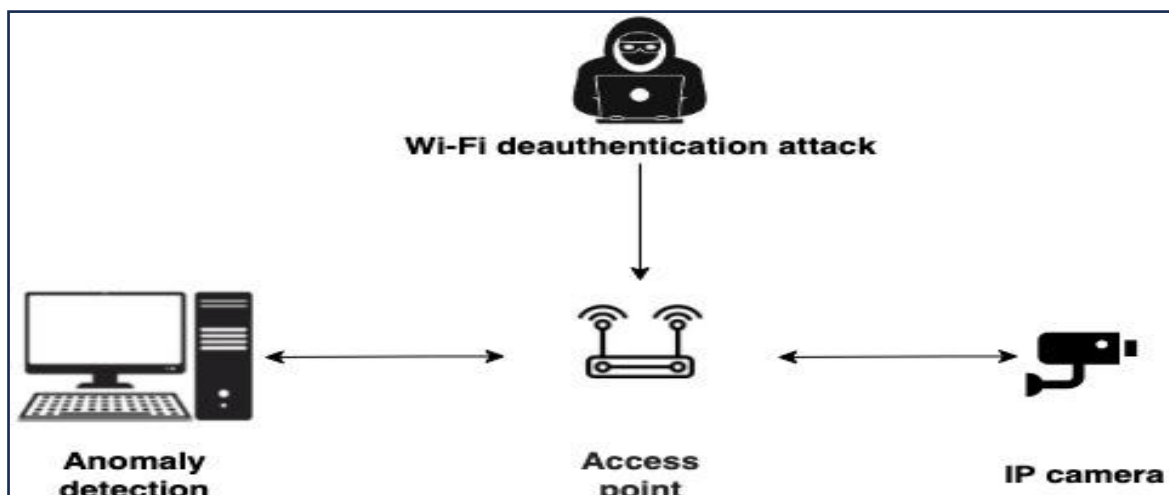
4. Background and Motivation

Modern wireless networks are susceptible to numerous threats, one of the most prominent being denial-of-service (DoS) attacks via deauthentication frames. The 802.11 standard allows unauthenticated deauth packets to be sent, potentially leading to connectivity loss. The purpose of this project is to highlight such vulnerabilities through a controlled, educational demonstration using affordable hardware.

- A general diagram of Wi-Fi network structure (Access Point → Clients).



- Visual showing how typical deauthentication occurs in a wireless network.



5. Literature Review

Prior works and research in wireless penetration testing emphasize the ease with which deauthentication attacks can be executed. Tools like Aircrack-ng and firmware projects such as Spacehuhn's Deauther showcase the risks associated with poor wireless configurations. Literature also emphasizes the importance of encryption (e.g., WPA3) and user education to mitigate such attacks.

6. Tools and Components

- NodeMCU ESP8266 Board
- USB Data Cable
- Laptop
- Arduino IDE or NodeMCU Flasher Tool
- ESP8266 Deauther Firmware (Spacehuhn)
- Web Browser

7. Methodology

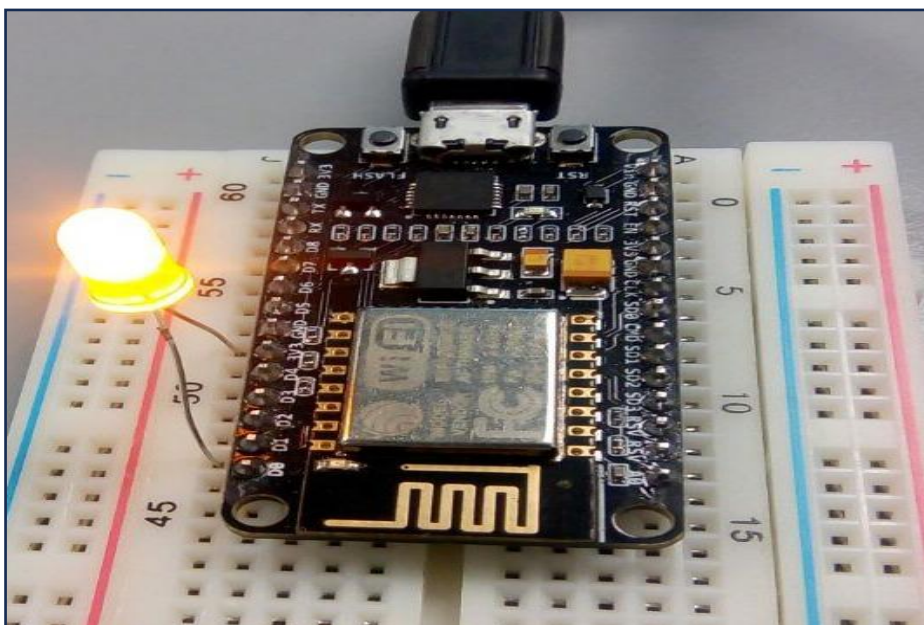
7.1 Firmware Selection and Flashing

The open-source Spacehuhn Deauther firmware was selected for its ease of use and strong documentation. The firmware was flashed to the ESP8266 using the NodeMCU Flasher tool.

7.2 Hardware Setup

Wires were connected to ensure stable power and communication. The NodeMCU was linked to a laptop via USB.

NodeMCU ESP8266 setup (wired and connected):



- Screenshot of the NodeMCU Wi-Fi network:



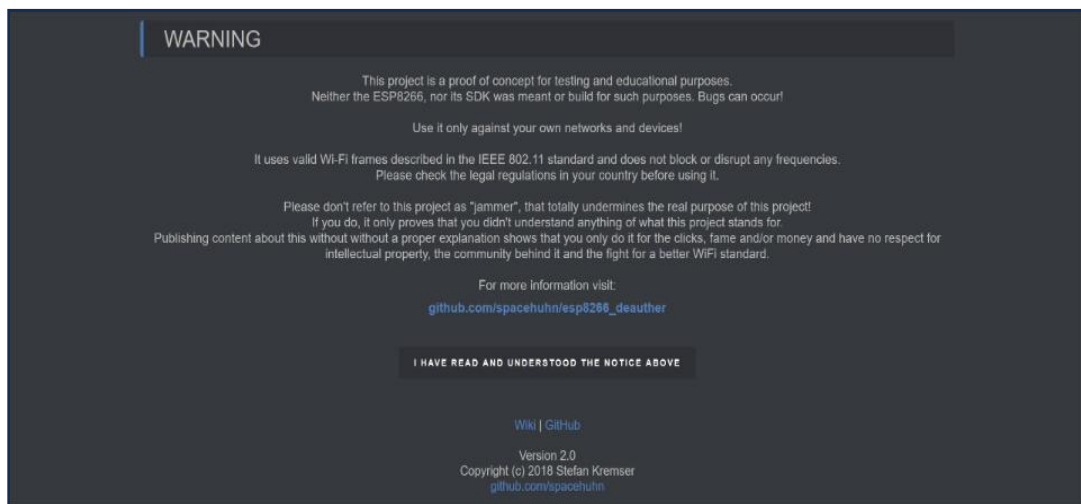
Screenshot of the web interface used to scan networks.

Access Points: 6							
SSID	Name	Ch	RSSI	Enc	MAC	Vendor	
0 Don't	--SpaceRouter!--	6	-57	WPA2	f4:6b:de:da:8d:95	Spacehuhn	<input type="checkbox"/>
1 call	<input type="button" value="ADD"/>	1	-60	-	cc:cf:1e:d5:5b:2b	SpaceLtd	<input checked="" type="checkbox"/>
2 It	<input type="button" value="ADD"/>	6	-81	WPA*	5c:37:3b:f7:67:be	SpaceBox	<input type="checkbox"/>
3 a	<input type="button" value="ADD"/>	8	-82	WPA2	cd:ce:1e:0a:4e:9e	SpacEEE	<input type="checkbox"/>
4 jammer	<input type="button" value="ADD"/>	8	-83	WPA2	c7:0e:14:95:a1:3b	Chicken!	<input type="checkbox"/>
5 Don't call it a Jammer! DON'T !!	<input type="button" value="ADD"/>	8	-90	WPA2	c8:0e:14:95:a1:3b	Huhn	<input type="checkbox"/>
<input type="button" value="SELECT ALL"/> <input type="button" value="DESELECT ALL"/>							

7.3 Network Connection and Web Interface Access

Once powered, the ESP8266 hosted its own access point (AP). A laptop was connected to this AP, and the web interface was accessed via browser at:

<https://deauther.com/docs/usage/web>.



7.4 Scanning and Target Selection

Using the interface, nearby Wi-Fi networks were scanned. One target test network (with explicit permission) was selected for the demonstration.

Access Points: 6

SSID	Name	Ch	RSSI	Enc	MAC	Vendor		
0 Don't	--SpaceRouter!--	6	-57	WPA2	🔒 f4:6b:de:da:8d:95	Spacehuhn	<input type="checkbox"/>	x
1 call	ADD	1	-80	-	cc:cf:1e:d5:5b:2b	SpaceLtd	<input checked="" type="checkbox"/>	x
2 It	ADD	6	-81	WPA*	🔒 5c:37:3b:f7:67:be	SpaceBox	<input type="checkbox"/>	x
3 a	ADD	8	-82	WPA2	🔒 od:ce:1e:0a:4e:9e	SpacEEE	<input type="checkbox"/>	x
4 jammer	ADD	8	-83	WPA2	🔒 c7:0e:14:95:a1:3b	Chicken!	<input type="checkbox"/>	x
5 Don't call It a Jammer! DONT !!	ADD	8	-90	WPA2	🔒 c8:0e:14:95:a1:3b	Huhn	<input type="checkbox"/>	x

SELECT ALL **DESELECT ALL**

7.5 Execution of Deauthentication Attack

The interface allowed the sending of continuous deauth frames to the selected network, effectively disconnecting clients from the access point temporarily.

Scan SSIDs Attacks Settings Info

Attacks

INFO:

- You might lose connection when starting an attack!
- You need to select a target for the deauth attack.
- You need a saved SSID for the beacon and probe attack.
- Click reload to refresh the packet rate.

In case of an unexpected error, please reload the site and look at the serial monitor for further debugging.

You might lose connection when starting the attack!

RELOAD

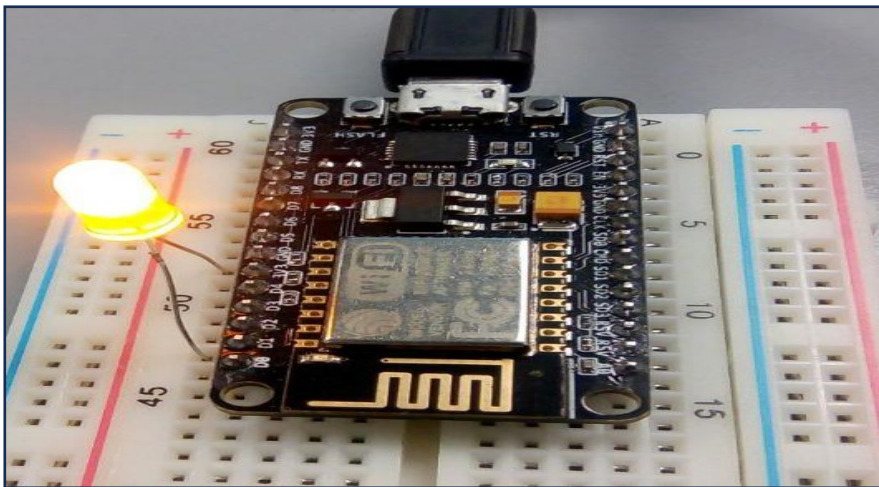
Attacks	Targets	Pkts/s	START / STOP
Deauth	0	0/0	START
Beacon	0	0/0	START
Probe	0	0/0	START

8.1 Step-by-Step Process

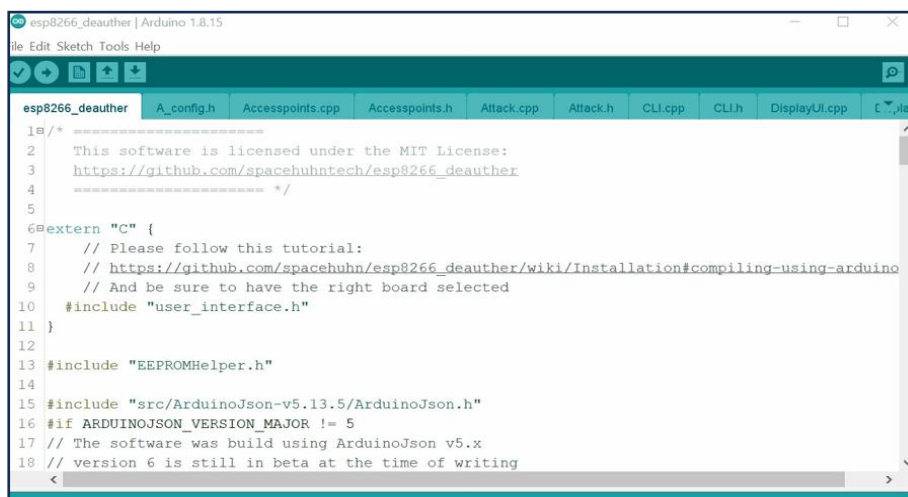
1. **Wire Connection:** Properly connect GPIO pins if needed (not required for basic setup).



2. **USB Connection:** Connect the NodeMCU to a laptop via USB.



3. **Firmware Flashing:** Use NodeMCU Flasher or Arduino IDE to upload the Spacehuhn firmware



```
esp8266_deauther | Arduino 1.8.15
File Edit Sketch Tools Help

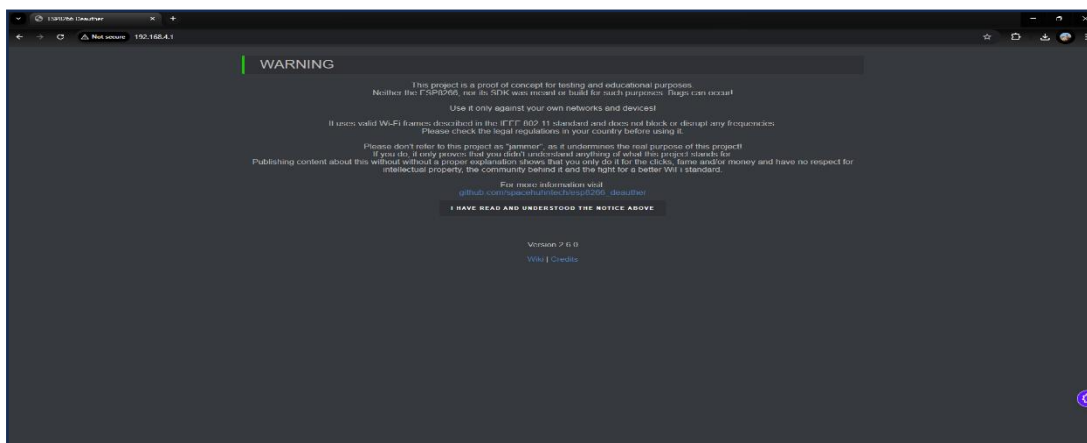
esp8266_deauther  A_config.h  Accesspoints.cpp  Accesspoints.h  Attack.cpp  Attack.h  CLI.cpp  CLI.h  DisplayUI.cpp  ...

1 // /*
2  // This software is licensed under the MIT License:
3  // https://github.com/spacehuhn/esp8266_deauther
4  //
5  //
6 #extern "C" {
7  // Please follow this tutorial:
8  // https://github.com/spacehuhn/esp8266_deauther/wiki/Installation#compiling-using-arduino
9  // And be sure to have the right board selected
10 #include "user_interface.h"
11 }
12
13 #include "EEPROMHelper.h"
14
15 #include "src/ArduinoJson-v5.13.5/ArduinoJson.h"
16 #if ARDUINOJSON_VERSION_MAJOR != 5
17 // The software was build using ArduinoJson v5.x
18 // version 6 is still in beta at the time of writing
19 <
```

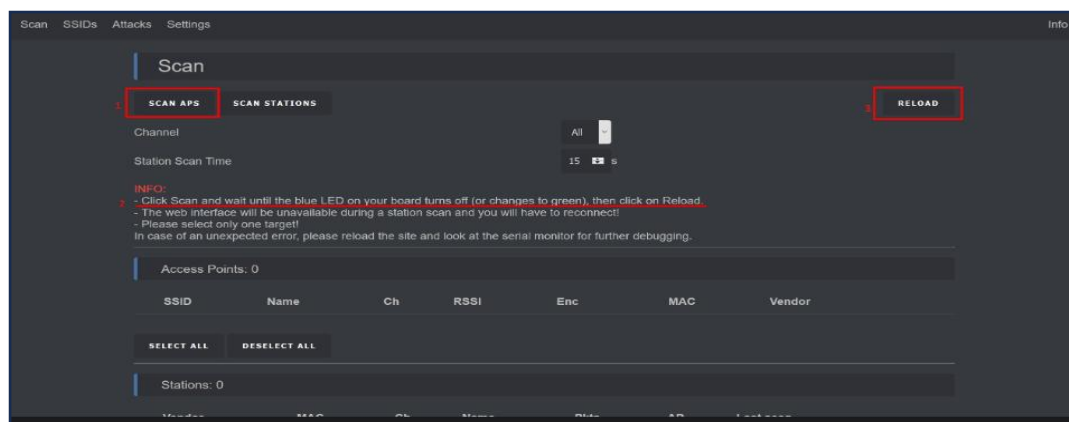
4. **Access Device Network:** From laptop Wi-Fi settings, connect to the AP broadcasted by NodeMCU (e.g., "pwned")



5. **Access Interface:** Open browser and navigate to <http://192.168.4.1> (default IP).



6. **Scan Networks:** Click on the Scan button to list nearby networks.



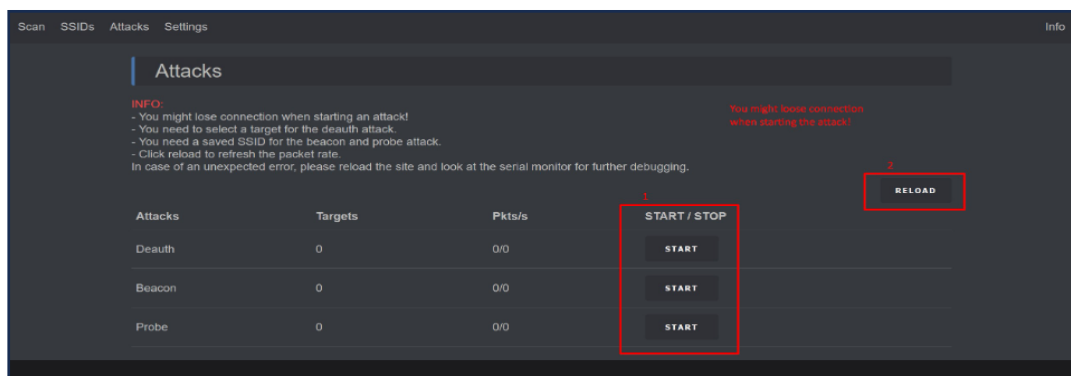
7. **Select and Attack:** Choose a network and begin deauthentication (for testing purposes only).

	SSID	Name	Ch	RSSI	Enc	MAC	Vendor		
0	Don't	--SpaceRouter!--	6	-57	WPA2	f4:6b:de:da:8d:95	Spacehuhn	<input type="checkbox"/>	x
1	call	ADD	1	-80	-	cc:cf:1e:d5:5b:2b	SpaceLtd	<input checked="" type="checkbox"/>	x
2	it	ADD	6	-81	WPA*	5c:37:3b:f7:67:be	SpaceBox	<input type="checkbox"/>	x
3	a	ADD	8	-82	WPA2	cd:ce:1e:0a:4e:9e	SpacEEE	<input type="checkbox"/>	x
4	jammer	ADD	8	-83	WPA2	c7:0e:14:95:a1:3b	Chicken!	<input type="checkbox"/>	x
5	Don't call it a Jammer! DONT !!	ADD	8	-90	WPA2	c8:0e:14:95:a1:3b	Huhn	<input type="checkbox"/>	x

SELECT ALL **DESELECT ALL**

8.2 Observations and Results

The attack successfully disconnected test devices from the selected Wi-Fi. The disconnection occurred rapidly and repeatedly until the attack was stopped. No permanent damage was caused. This reinforced the ease and potential disruption caused by such tools.



9. Technical Overview

9.1 802.11 Protocol and Deauthentication Frames

The 802.11 standard allows management frames to be sent without encryption. Deauthentication frames are used to terminate connections between clients and APs. Attackers can spoof MAC addresses and send forged frames, resulting in disconnection. WPA2 and WPA3 do not encrypt these management frames unless protected by 802.11w.

9.2 ESP8266 and Web Interface Functionality

The ESP8266 uses its internal Wi-Fi module to scan and transmit 802.11 packets. The firmware provides an HTML interface through which users can execute attacks, scan networks, or monitor activity. This functionality simulates a rogue AP or attacker in a real-world penetration testing scenario.

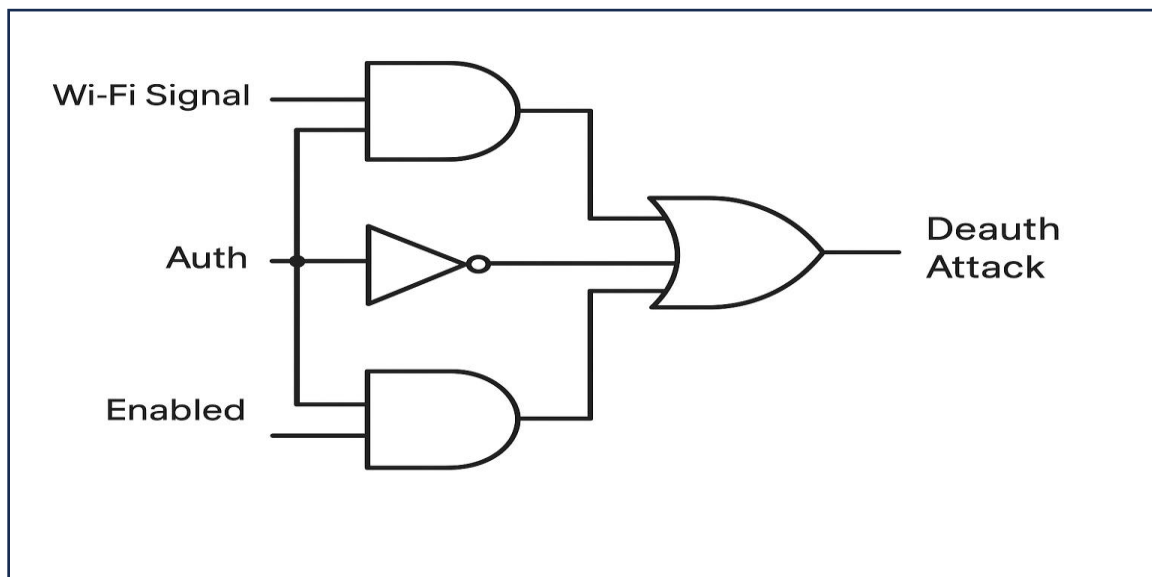
10. Security Implications and Ethical Considerations

This project was conducted with strict adherence to ethical guidelines. Only networks we owned or had explicit permission to test were targeted. Wi-Fi deauthentication is illegal without consent and can lead to criminal prosecution. The purpose of this project is purely academic: to raise awareness of network vulnerabilities and promote secure wireless configurations such as:

- Using WPA3 encryption
- Enabling 802.11w Management Frame Protection
- Regular firmware updates for routers
- Network segmentation and device isolation

DISCLAIMER: This project is for educational purposes only. Unauthorized use of deauthentication tools is illegal.

11. Logic Diagram:



Logic-Level Representation of Deauthentication Process

To better understand the internal decision-making logic of the Wi-Fi Deauther system, we have translated the core concept into a logic circuit diagram using basic gates (AND, OR, NOT). This abstract model simplifies the conditions under which a deauthentication attack is initiated.

Inputs:

1. **Wi-Fi Signal:** Represents whether a Wi-Fi access point is detected and a connection exists.
2. **Auth (Authentication):** Indicates whether the current device is authenticated with the target network.
3. **Enabled:** Reflects whether the deauther attack feature is manually enabled by the user through the web interface.

Logic Flow:

- The **Auth** signal passes through a **NOT gate**, converting an authenticated state (1) into a deauthenticated state (0) and vice versa. This simulates the deauthentication goal of the project.
- The **Wi-Fi Signal** is combined with the **Enabled** signal using **AND gates** to ensure that the attack can only proceed when both are active (i.e., a Wi-Fi network is present and the user has explicitly activated the attack).
- The output of the NOT (inverted Auth) and the two AND gates are fed into an **OR gate**.
- The final output, **Deauth Attack**, becomes high (1) when any of the logical conditions are true:
 - There is a Wi-Fi signal and attack is enabled, or
 - The device is not authenticated (i.e., suitable for sending a deauth packet).

Interpretation:

This logic model ensures that a deauthentication attack is only initiated when:

- A valid Wi-Fi signal is detected,
- The user has enabled the attack, and
- The target device is not already authenticated or the intention is to forcibly disrupt it.

This simplified circuit illustrates how conditions must align to trigger a responsible, controlled action. Though abstract, it reflects the structured checks performed in actual firmware logic.

12. Challenges Faced

- Initial difficulty flashing firmware due to driver issues
- Compatibility concerns with USB ports and cable types
- Intermittent stability of the NodeMCU AP
- Legal and ethical boundaries limiting the scope of testing

13. Conclusion

The Wi-Fi Deauther project demonstrated how vulnerable common Wi-Fi setups can be to simple deauthentication attacks. Using affordable and accessible hardware, we replicated a denial-of-service attack and gained valuable insight into network security risks. This project highlights the critical need for secure wireless practices and serves as a hands-on educational tool for cybersecurity learning.

14. References

1. IEEE 802.11 Wireless LAN Standards
2. Spacehuhn Technologies - ESP8266 Deauther Documentation:
https://github.com/SpacehuhnTech/esp8266_deauther
3. Arduino IDE Official Documentation
4. NIST SP 800-153: Guidelines for Securing Wireless Local Area Networks
5. Wi-Fi Alliance - WPA3 Overview
6. Deauther Web Usage: <https://deauther.com/docs/usage/web>