Software Requirements Specification (SRS)

Project Name: Wi-Fi Attack Simulator (Educational Use Only)
Prepared by: *Harsh , Varun , Sarit , Dikshansh*

---

1. Introduction

1.1 Purpose

This document specifies the software requirements for the **Wi-Fi Attack Simulator**, a tool designed for educational use to simulate common wireless network attacks like deauthentication, Evil Twin AP setup, WPA/WPA2 handshake capture, and basic packet sniffing.

1.2 Scope

The simulator helps cybersecurity students and researchers understand how wireless attacks work in a controlled lab/test environment.
Key features include:

- Deauth attack using Scapy
- Evil Twin access point:
    - Using ESP8266 NodeMCU
    - Using Kali Linux tools (hostapd + dnsmasq)
- Optional modules (in development): WPA handshake capture, packet sniffer

1.3 Intended Audience

- Cybersecurity students
- Ethical hackers (in labs)
- Network security instructors
- Admins testing network resilience

---

2. Overall Description

2.1 Product Perspective

This is a standalone educational simulator with:

- CLI-based Python tools (Kali Linux)
- Arduino/PlatformIO ESP8266 AP code
- Bash config for Kali-based fake AP

2.2 Product Functions

- Simulate deauthentication attacks
- Clone access points (Evil Twin)
- Capture WPA handshakes (future)
- Demonstrate phishing login via fake portal

2.3 User Classes and Characteristics

User Type	Description
Learner	Uses CLI or web interface to launch attacks
Instructor	Demonstrates attacks in a safe lab
Admin/Analyst	Tests detection & monitoring setups

---

3. System Features

3.1 Deauthentication Attack

- Sends spoofed deauth frames to disconnect Wi-Fi clients.
- Requires monitor mode and packet injection capability.

3.2 Evil Twin Attack – Kali Linux

- Clones SSID and hosts a phishing login page.
- Uses hostapd and dnsmasq.

3.3 Evil Twin Attack – ESP8266

- Creates an open AP with a spoofed name (e.g., Free_Public_WiFi).
- Can be extended to serve a captive portal.

---

4. External Interface Requirements

4.1 Hardware Interfaces

- USB Wi-Fi adapter with monitor mode (for Kali-based scripts)
- ESP8266 NodeMCU board (for microcontroller-based AP)

4.2 Software Interfaces

- Python 3.x
- Arduino IDE / PlatformIO
- Tools: aircrack-ng, hostapd, dnsmasq, scapy, apache2/nginx

4.3 User Interfaces

- Command-line input for Python/Bash scripts
- Arduino code upload via USB
- Optional: Web dashboard (Flask – future module)

---

5. Non-Functional Requirements

5.1 Performance

- Must run on modern Linux systems with root access
- Real-time packet injection and logging

5.2 Safety and Ethics

- Clearly marked for **educational use only**
- Includes disclaimer in all files
- Must be used only on test networks

5.3 Security

- Fake login pages do not store data
- Password forms only simulate user input for teaching

---

6. Constraints

- Not to be used on real networks without permission
- Requires hardware support for monitor mode (e.g., Alfa cards)

- ESP8266 version has limited memory and range

---

7. Assumptions and Dependencies

- User has Linux and networking basics
- User has sudo or root access
- ESP8266 drivers installed for flashing code
- Attack modules run with proper permissions

---

8. Future Enhancements

- Flask web dashboard to run attacks via GUI
- Logging with MAC/IP/time tracking
- WPA handshake capture & cracking module
- Integration with Wireshark or Kismet
- ML-based Evil Twin detection