

Project: WiFiGuard – Wireless Penetration Testing Toolkit

Instructor: Muhammad Ali Naseer Shah

Course: Ethical Hacking

Section: BSCYS-6A

Members:

• Sami-Ur-Rehman🡪 K224673

• Ayan Mohammad Zakriya🡪 K224728

• Hamza Yousuf 🡪K224748

**🔹 Abstract**

WiFiGuard is a modular wireless penetration testing toolkit developed in Python. It enables users to simulate real-world Wi-Fi attacks such as deauthentication, MAC address spoofing, rogue access point creation, phishing via captive portals, and WPA/WPA2 password cracking. The toolkit integrates well-known utilities like aircrack-ng, hostapd, and dnsmasq, providing an educational platform for ethical hacking and wireless security analysis in controlled environments.

**🔹 Introduction**

Wireless networks are widely used but often poorly secured. Attackers can exploit weak configurations, outdated protocols, or user negligence to gain unauthorized access. WiFiGuard was created to demonstrate such attack techniques in an ethical lab environment. Built using Python scripts and Linux-based tools, the toolkit provides hands-on experience in understanding and testing the security of Wi-Fi networks.

**🔹 Literature Review**

Many open-source tools like Aircrack-ng, Wifite, and Bettercap are used in Wi-Fi penetration testing. These tools show how vulnerabilities in WPA/WPA2, WPS, and management frames can be exploited. WiFiGuard incorporates similar **methodologies:**

* **Deauthentication Attacks:** Based on IEEE 802.11 management frame abuse to disconnect clients.
* **Rogue Access Points:** Used to impersonate trusted networks and capture credentials.
* **Captive Portal Phishing:** Simulates real-world social engineering attacks.
* **MAC Address Spoofing:** Helps bypass basic access control mechanisms.
* **WPA/WPA2 Cracking:** Demonstrates brute-force attacks using captured handshakes.

The project builds on these foundations to provide a step-by-step, user-guided interface for educational use.

**🔹 Methodology**

WiFiGuard is divided into multiple Python modules, each responsible for a different attack or task. Below are the core functionalities:

**🔸 Monitor Mode & MAC Spoofing**

* **enable\_monitor\_mode.py, disable\_monitor\_mode.py:** Toggle monitor mode.
* **change\_mac\_user\_input.py:** Randomizes MAC address with macchanger.

**🔸 Scanning & Recon**

* **wifi\_scanner.py:** Uses airodump-ng to detect networks.
* **wifi\_attack.py:** Captures WPA/WPA2 handshakes while running deauth attacks.

**🔸 Rogue Access Point & Captive Portal**

* **rogue\_ap.py:** Creates a fake Wi-Fi using hostapd, dnsmasq, and static IP routing.
* **captive\_portal.py:** Serves a phishing page (login.html) that stores credentials in creds.txt.

**🔸 Password Cracking**

* **wifi\_password\_cracker.py:** Builds wordlists using crunch, then cracks handshakes using aircrack-ng.

**🔸 Tool Installation**

* **tools\_installer.py:** Installs core tools like aircrack-ng, wifite2, bettercap, reaver, pixiewps, ghost-phisher.

**🔸 Main Launcher**

* **eth.py:** Provides a menu-driven CLI interface to access all tools and attacks in one place.

**🔹 Tools Used**

* **aircrack-ng –** Cracks WPA/WPA2 passwords using captured handshakes and wordlists.
* **airodump-ng –** Scans Wi-Fi networks and captures handshakes and client information.
* **aireplay-ng –** Sends deauthentication packets to disconnect clients from a Wi-Fi network.
* **hostapd –** Creates a rogue access point to simulate a real Wi-Fi network.
* **dnsmasq –** Provides DHCP and DNS services for the rogue access point.
* **macchanger –** Changes the MAC address to anonymize the attacker's device.
* **crunch –** Generates custom password wordlists based on defined rules.
* **reaver –** Performs brute-force attacks on WPS-enabled routers.
* **pixiewps –** Launches offline WPS PIN attacks using Pixie Dust vulnerability.
* **bully –** Another tool for attacking WPS, useful against locked APs.
* **wifite2 –** Automates Wi-Fi attacks including handshake capture and WPS cracking.
* **bettercap –** Supports advanced MITM attacks over Wi-Fi and Bluetooth.
* **ghost-phisher –** GUI tool for creating rogue APs and phishing pages.
* **iptables –** Sets up NAT and forwards traffic from the fake AP to the internet.
* **Flask –** Hosts the fake login page used in captive portal phishing.
* **xterm –** Opens terminal windows to run tools like aireplay-ng during attack.

**🔹 Techniques Used**

* **Monitor Mode Activation –** Enables packet sniffing by setting Wi-Fi card to monitor mode.
* **MAC Spoofing –** Hides the real device identity by changing its MAC address.
* **Wi-Fi Scanning –** Detects surrounding access points, clients, and signal strength.
* **Deauthentication Attack –** Disconnects devices to force them to reconnect and expose handshakes.
* **Handshake Capture –** Records WPA/WPA2 authentication data for later cracking.
* **Rogue Access Point –** Sets up a fake Wi-Fi network that victims may connect to.
* **Captive Portal Phishing –** Tricks users into entering credentials on a fake login page.
* **Password Cracking (Brute Force) –** Attempts to guess passwords using generated wordlists.
* **WPS PIN Exploitation –** Bypasses Wi-Fi passwords by attacking weak WPS configurations.
* **Automated Attack Flow –** Offers a simple menu system to run attacks step by step.

**🔹 Conclusion**

WiFiGuard serves as a practical and educational toolkit for learning Wi-Fi security. It demonstrates key vulnerabilities in wireless networks and shows how attackers exploit them. The modular structure and simplified menu system make it ideal for students, ethical hackers, and trainers. All components work together to simulate real attacks, making the toolkit a valuable learning aid for wireless penetration testing.

**🔹 References**

1. IEEE 802.11 Wireless LAN Standard
2. Aircrack-ng Documentation – <https://www.aircrack-ng.org/>
3. Kali Linux Wireless Tools – https://tools.kali.org/
4. Bettercap – <https://www.bettercap.org/>
5. Hostapd Guide – https://w1.fi/hostapd/