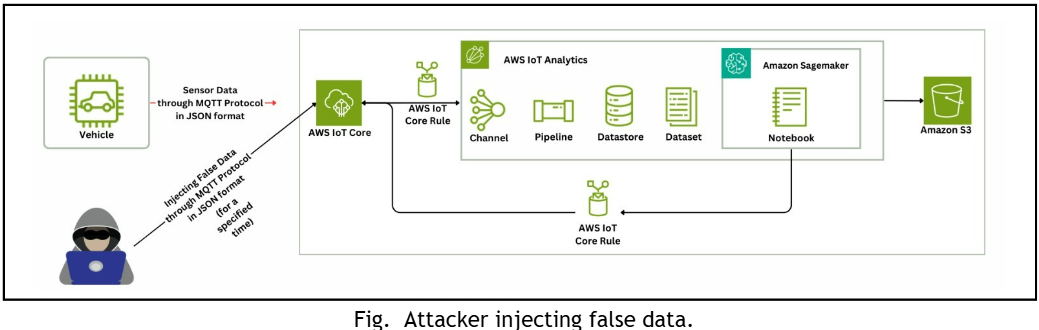
|  |  |
| --- | --- |
| **1.** | **ATTACK SCENARIO** |

In this project, we simulate an attack to introduce false data into the communication channel between a legitimate autonomous vehicle and the cloud. The goal is to test our machine learning model deployed on Amazon SageMaker, which is designed to detect any such anomalies.The architecture diagram of the attack scenario is depicted in the below figure.



For the attack scenario, we assume the presence of an SDK file containing the false data that we aim to inject. To facilitate the attack, we use a network adaptor with monitor mode capabilities, specifically the TP-LINK TL-WN722N which we used here.We leverage a suite of advanced wireless network auditing tools—airmon-ng, airodump-ng, and aireplay-ng—to facilitate our simulation and testing process. These tools are integral to executing specific tasks within the attack scenario, ensuring precision and efficiency at every stage.

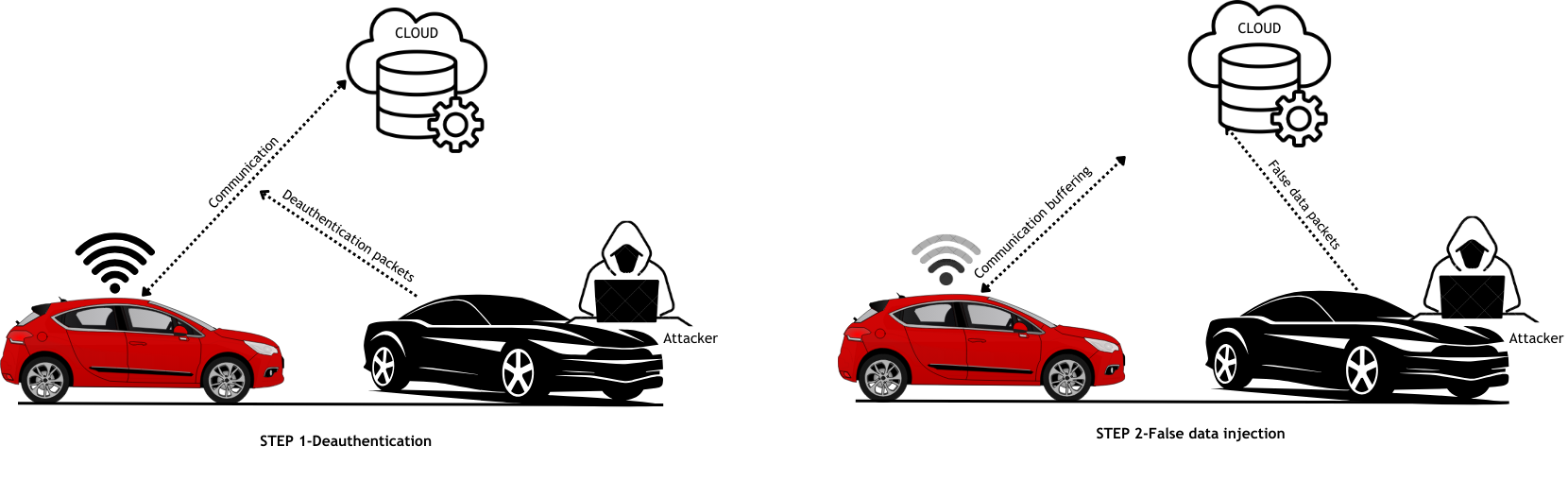


Fig 2. Two steps in attack scenario

Initially, we scan for available networks in the vicinity using the airodump-ng tool[1]. Once we identify the desired network employing WPA2 (Wi-Fi Protected Access) security.Although WPA3 has been introduced as a more secure protocol, WPA2 remains the predominant security standard in the automotive sector.Then we proceed to target that network using the airodump-ng tool along with the BSSID of the network. This allows us to isolate the legitimate vehicle connected to it.

Subsequently, we initiate a deauthentication attack(STEP-1) using the aireplay-ng tool, temporarily disconnecting the legitimate car by sending a specified number of deauthentication packets.

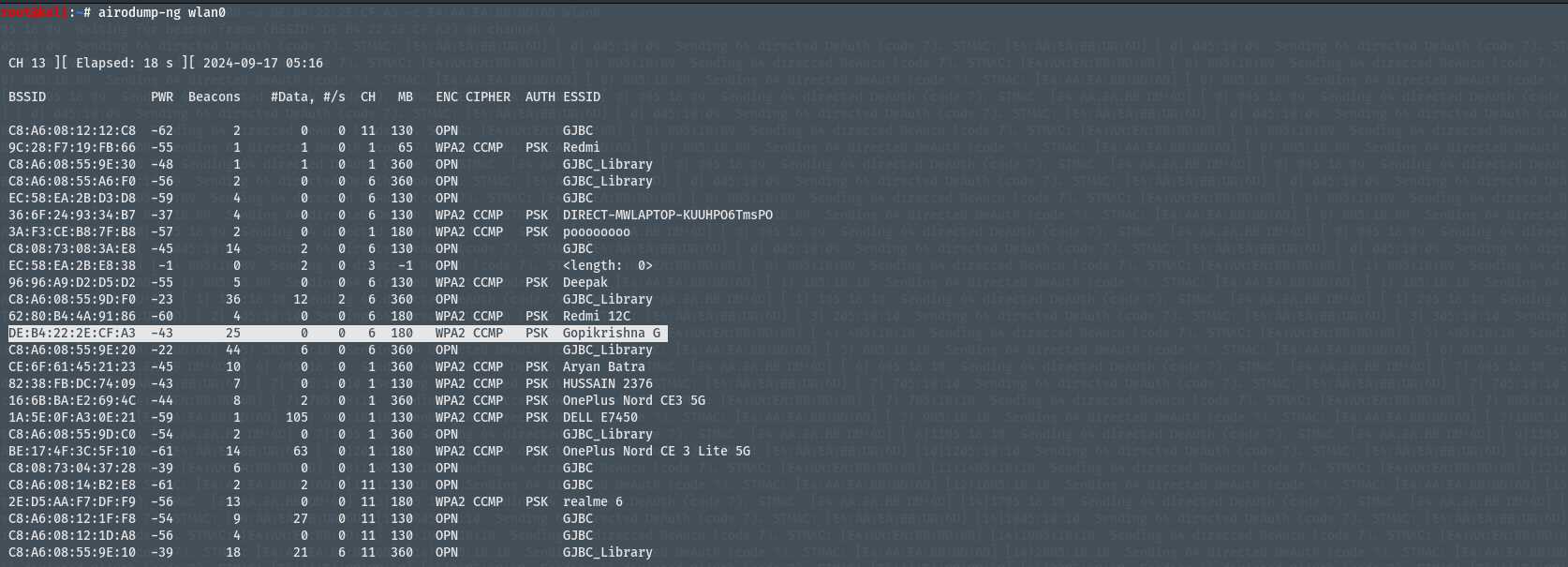
During this brief window, typically under 30 seconds, we inject the false data(STEP-2) into the network via the SDK file. Once the injection is complete, the deauthentication attack is terminated, restoring normal communication. Due to the speed and precision of this process, the legitimate vehicle remains unaware of the intrusion.

**In STEP-1 :**

**Deauthentication**

**1.airodump-ng wlan0**(wlan0 is the name of the wireless router of the device).

This command displays all the network devices available nearby.



### Fig 2. All nearby networks shown

### **airodump-ng -d** BSSID [router’s BSSID] –channel [router’s channel]

### wlan0

This command is used to show the devices connected to specific network.

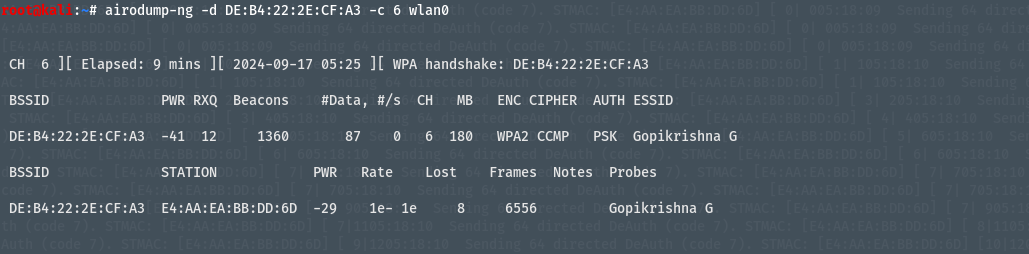


Fig 3.The specific network to which the victim is connected is shown in detail

### **a**ireplay-ng–deauth 100000[no. of deauth requests] -a[mac

### address of target network] -c[mac address of victim] wlan0

This command is used to deauth the specified device from specific network.

### 

Fig 4.Deauthenticating the device from specified network.

**IN STEP -2**

**Data injection**

Here we are sending the false data file (v5.csv), which includes manipulated parameters such as speed, time, distance, and others (totaling 14 parameters), through PowerShell,using MQTT communication.

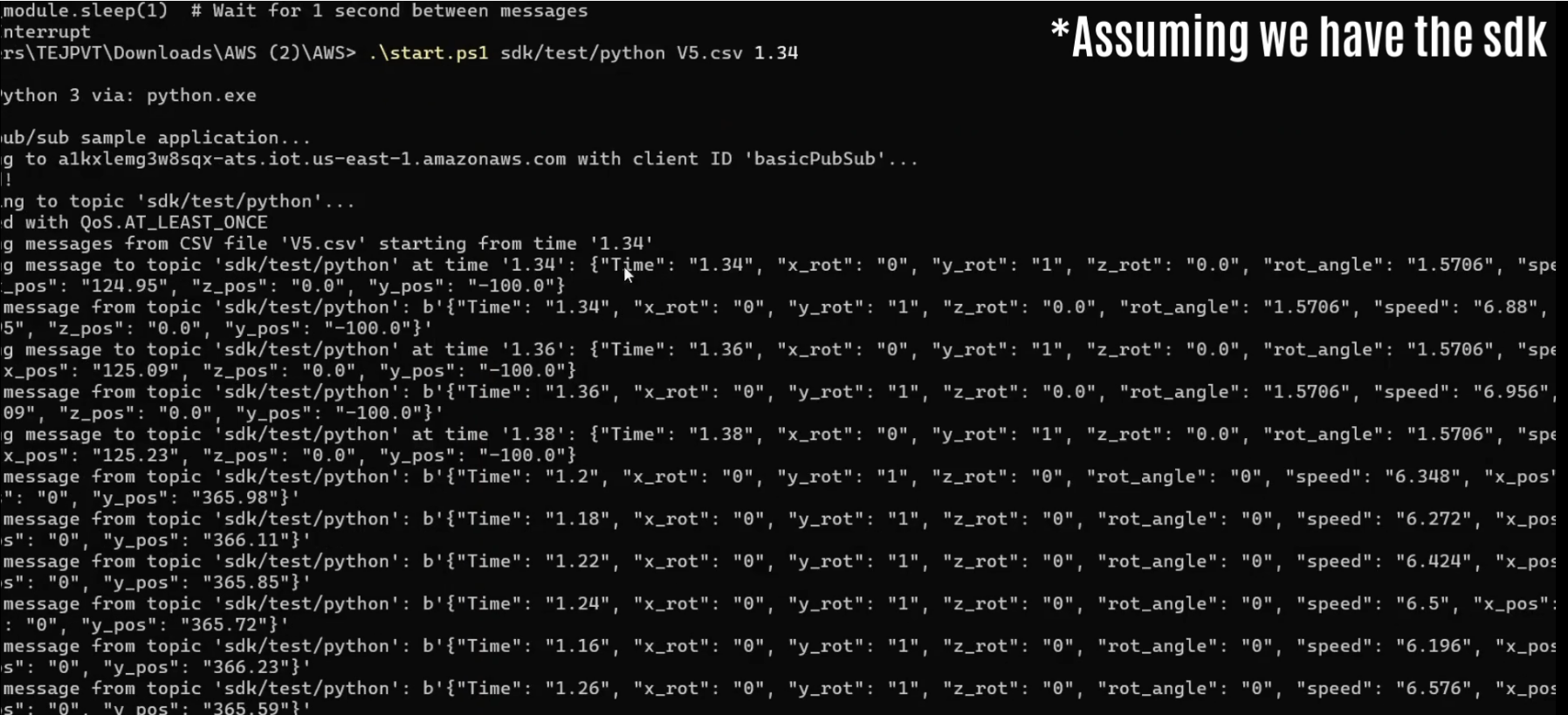


Fig 5. Injecting the false data .

It is important to emphasize that the primary focus of this project is not on executing the attack but rather on detecting and validating whether an attack has occurred. This testing approach serves purely as a means to evaluate and validate the effectiveness of the detection mechanis.

References:

1. T. Rakhra, A. Kaushal, S. Tanwar, P. Datta and A. Rana, "De Authentication Attack: A Review," 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), Gunupur Odisha, India, 2020, pp. 1-6, doi: 10.1109/iSSSC50941.2020.9358889. keywords: {Computer hacking;Web and internet services;Authentication;Companies;Security;Faces;Wireless fidelity;Cyber-attacks;Hacking;Ethical Hacking;Malicious Hackers;DE authentication attacks},