Smartphone Location Spoofing Attack in Wireless Networks

Chengbin Hu[1(✉)], Yao Liu[1], Zhuo Lu[2], Shangqing Zhao[2], Xiao Han[1], and Junjie Xiong[1]

[1] Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA
{chengbin,yliu21,xiaoh,junjiexiong}@usf.edu
[2] Department of Electrical Engineering, University of South Florida, Tampa, FL 33620, USA
{zhuolu,shangqing}@usf.edu

Abstract. GPS-free outdoor localization becomes popular because of the expanding scale of WiFi deployments in metropolitan areas. As a substitution or complement to the Global Positioning System (GPS), WiFi localization systems provide very accurate results in WiFi-rich area. However, the current WiFi localization systems are not robust to WiFi external signal attack. In this study, we implement a reverse engineering model to decode the Android WiFi localization system output. With the aid of reverse engineering mode, we implement both static and dynamic external signal attacks to make the smartphone believing it is located in another location or moving along the attacker's designed route using a portable programmed IoT device ESP8266. We also demonstrate that the WiFi based localization and navigation are vulnerable to external signal attacks by testing this attack on Android smartphone. Finally, we discuss the possible defense solutions and the future work. Our study indicates the smartphone is vulnerable to external signal attacks and there is an urgent need for defense solutions.

Keywords: Location security · Wireless signal · External signal attack

1 Introduction

Without a doubt, smartphones have become one of the most widely used technologies today. Billions of users in the world are relying on mobile positioning and navigation services on a daily basis [1]. Applications range from map navigation, service suggestion, to social media, all require accurate and authentic location data. In the meantime, the popularity of these location-based applications has become a honey pot for attackers. Those attackers aim to manipulate the location data, which is reported by smartphones, for criminal activities. In the past, the traditional positioning system relies on GPS signals. However, with the rapid development of WiFi technology, the WiFi-based positioning system has been widely used, which is not only as a complementary location-based service of GPS but also as a major positioning system [2].

J. Garcia-Alfaro et al. (Eds.): SecureComm 2021, LNICST 399, pp. 295–313, 2021.
https://doi.org/10.1007/978-3-030-90022-9_15

Because most positioning signals are not encrypted, the current positioning system is vulnerable to spoofing attacks where the attackers can send falsified wireless signals to change the smartphone location result. The previous research shows that a static attack that spoofs the smartphone location system to a false position is easy to achieve. But the dynamic attack that making the smartphone believe it is moving along the designed fake route of the attacker is limited [3]. In this paper, we exploit the current vulnerabilities of the Android wireless location system and design both static and dynamic attacks against it. We prove that it is feasible to attack the current smartphone location system and it is imperative to find a defense solution for this type of attack.

Numerous studies have attempted to explain the risks of location spoofing attack [4] and have created GPS, WiFi spoofers to alternate the positioning result of smartphones. Previous studies have shown that the positioning system is vulnerable to external signal attacks by broadcasting false GPS and WiFi positioning signals [5]. However, the current attacks still have a lot of limitations such as they require a specific shape of the route for a dynamic route or they can only spoof the device to a static position. Thus, it is important to study the possibility of the external signal attacks of current location systems and access the robustness of current location systems against these attacks.

In this paper, we first systematically study the current positioning system by designing a machine learning model using reverse-engineering to mimic the current Android positioning algorithm. Second, we design an external wireless signal spoofer using a portable IoT device. Third, we design the attacking algorithms on the device and evaluate the attack results using experiments. We also discuss defense solutions and future work. In summary, the key contributions of this paper include:

- We propose a machine learning model for building a reverse-engineering model of the Android WiFi positioning system. This proposed positioning model can mimic the positioning system and can help us to understand the attack and defense of the positioning system.
- We implement the external signal attack on a low-cost portable IoT device.
- We evaluate the attack in the real-world scenario. We prove the current smartphone is vulnerable to the external signal attack.

The rest of the paper is organized as follows. Section 2 introduces the background and threat model. Section 3 describes the WiFi positioning algorithm reverse-engineering. Section 4 shows the implementation of the attack. Section 5 demonstrates the result and evaluation of a real-world attack. Section 6 discusses the defense solutions and the future directions of our work. Finally, Sect. 7 concludes this paper.

2 Background and Threat Model

In this section, we start by providing the background of the smartphone positioning system, smartphone positioning system, positioning-based smartphone

apps, and the external signal attacks. We then describe the unique challenges in attacking the smartphone positioning system.

Smartphone Positioning System. The current state-of-art smartphone positioning system uses multiple signal sources, including GPS, and WiFi as the input to determine the position of the device [6]. The Global Positioning System (GPS) is a traditional positioning system using satellite signals. Since the GPS positioning signal may not be available under certain conditions such as bad weather, dense buildings or intentionally blocking signal, the smartphone positioning system uses other Wireless signals such as the WiFi as a substitution or complement to the GPS. In this paper, we focus on Google geolocation API which is used for an Android-based WiFi localization system. In brief, the Android-based WiFi localization service relies on a central reference database as shown in Fig. 1 [7]. The central reference database is maintained by the location service company and stores most geolocation information of WiFi APs around the world. When a user device such as a smartphone initiates a request of location, the device starts to scan all the WiFi APs around it. Once the device gets all the BSSIDs/MAC/RSSI information of these APs, it starts to query the central reference database with these data. If the central reference database can find it, it will return a MAC: location pair for each MAC. Finally, the positioning algorithm computes the positioning result based on the data received from the central database. This positioning result can be reported alone or as a complement to increase the accuracy of GPS results.

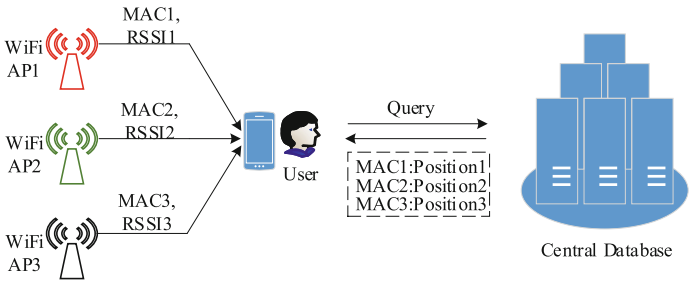

Fig. 1. WiFi localization service.

Positioning-Based Smartphone Apps. The use of location service represents one of the biggest advancements in smartphone technology. With the location service, the smartphone app can deliver online content to users based on their physical location. The popular location service-based smartphone apps include map navigation, service suggestion, social media, and game. These apps can provide users location-related content such as a positioning marker on the map and surrounding environment details. With the development of smartphone technology, the location service market grows rapidly. 55% of smartphone users use location-based services of the smartphone and the research study shows the young population more prefers to use the location services [8].

WiFi External Signal Attacks. The unencrypted civilian GPS signal is vulnerable to spoofing attacks. Many studies are showing that using a radio simulator or a satellite simulator to send false GPS data can successfully launch a location spoofing attack to GPS system [9,10]. In [9], The authors use a satellite simulator to attack a navigation system to create a dynamic route. However, their attack requires a new attack route that has the same shape as the real route drive, which is limited in many areas.

Some attacks are effective but not stealthy. In [11], the attack needs to inject malicious code into the smartphone application or modify the smartphone software system. This type of false GPS signal attack can be detected by the malware defense mechanism. Because the WiFi signal plays an important role in the positioning system, some studies use manipulated WiFi signals to spoof the positioning system. As reported in [12], the iPhone location can be relocated when attackers sending WiFi beacons to other locations. In addition, the Skylift project uses a low-cost ESP8266 WiFi chip to successfully manipulate the current location of an iPhone [13]. We notice that these WiFi signal-based attacks are limited to spoof the device to a single location in a static way. Therefore, it is urgent to test whether the attacker can spoof the device in a dynamic way that can let the device believe it is moving along a false route using WiFi signals.

2.1 Threat Model

In this paper, we design a novel attack against the smartphone positioning system.

The attacker, which can be hidden devices installed on the victim's car, uses a radio simulator to block the legitimate GPS signal. The attacker then sends the spoof WiFi beacon frame packets to the victim's device. The spoofing WiFi signals let the victim's device believe it is located on a spoofed location or moving according to a route designed by the attacker.

In our threat model, the attacker only relies on the external signal and has no access to the internal software or hardware of the victim's smartphone. Compared to the spoofing positioning system via the GPS signal reported in previous work, there are two unique challenges to cheat the WiFi positioning system.

Accuracy of Spoofing Result: First, to successfully attack the WiFi positioning system, an attacker needs to keep the error of the spoofing result to a minimum. This includes two parts, the final spoofing result should be close to the designed spoofing result. The positioning error should be minimal. In a common attack scenario, when a target device is moving on route A, the attacker wants to let the target device believe it is moving on route B by broadcasting the WiFi signals collected from route B. When there are a lot of legitimate WiFi access points, it is very hard to spoof the device without jamming these legitimate APs. To solve this challenge, Kumar, P., et al. [14] use an override method that mimics much more spoofing APs than legitimate APs (more than five times). These false AP signals can successfully attack the location-based apps on Android and iPhone devices. Therefore, the attacker can broadcast stronger WiFi signals and

more SSIDs to increase the false input of the WiFi positioning algorithm, which can achieve a more accurate positioning result. Our study aims to solve how to increase the accuracy of the positioning result using the low power profile IoT device ESP8266.

Device Choice and Synchronization: Second, the attacker needs to find the best portable device that supports the 802.11 protocol. The attacker needs either a powerful device that is capable to broadcast all the required SSIDs or a more portable device that is easy to hidden. Although a common laptop can send the designed Wireless packets, it is too big for launching a secret attack. According to our attack scenario, the attacker prefers a small and quiet device that is easy to be overlooked. Therefore, a portable 802.11 g/n ESP8266 is used for sending WiFi beacon frames to simulate the WiFi APs [15]. The ESP8266 is a low-cost and small device that has the full function of sending 802.11 frames. According to our bandwidth test and the online test report, this device has a stable max transmission rate of 2.7 Mbps [16]. Since a legitimate AP beacon deliveries a beacon frame at about 0.018 Mbps, the theoretical capacity of the number of WiFi APs that one ESP8266 can send is around 150. However, during a WiFi external attack the ESP8266 needs more resources to execute our attack algorithm, this number drops to less than 60 in our experiment. Because in some WiFi density areas, there can be more than 60 WiFi APs in the environment. The attacker needs to find an optimal way to divide the dataset and synchronize the portable devices that are working together. The attacker can divide the total dataset D into several subsets for each portable device. Therefore, each subset contains a part of SSIDs in D. Then, these devices have to work coordinately to send the total dataset.

3 WiFi Positioning Algorithm Reverse-Engineering Model

As we know from the background, a WiFi positioning algorithm needs the data of WiFi APs and a central reference database as input. The output is GPS coordinates with longitude and latitude. There are four steps used in the reversing engineering of the current WiFi positioning algorithm: Data Capture, Preprocessing, Positioning Modeling, and Algorithm Evaluation shown as Fig. 2.

3.1 Data Capture

As we introduced, the WiFi positioning algorithm relies on the querying result of the central database. Therefore, the data capture includes a reconstruction of the central database. In addition, we also need to collect the training/testing data for geolocations that the attacker is using.

To reconstruct a central database, we need to provide the longitude and latitude for each WiFi AP. Therefore, the format of a record in the central database is shown as MAC: longitude, latitude. The Google geolocation API

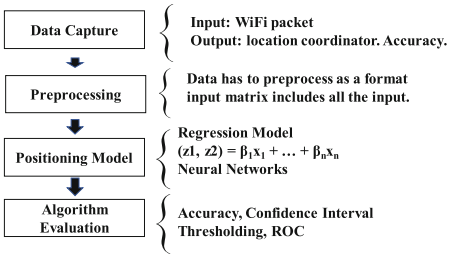

Fig. 2. Reverse-engineering model

can return a longitude and latitude pair based on the input GSM/WiFi data. Thus, we query the Google geolocation API for every WiFi AP for its associated longitude, latitude.

To collect the geolocation training/testing data, an Android phone Nexus 6 with WiFiManager API [17] is used for scanning WiFi SSIDs along a planned route. The WiFiManager API collects all the essential information about the SSIDs it discovers, such as RSSI, MAC, GPS positioning coordinates, timestamps, etc. We collect the WiFi SSIDs by driving through the route and record the SSIDs once per second. Among these data, RSSI and MACs are used as input data. The GPS positioning coordinates are used as the ground truth of the outcome data when observing certain SSIDs.

In this study, we collect three routes that represent 10 WiFi-rich areas, 10 WiFi-medium-rich areas, and 10 WiFi-poor areas. For each positioning (a certain longitude, latitude), the average number of WiFi APs is 26.8. We consider one position with its associated WiFi APs as a data record. There are 32142 data records in our total data set.

3.2 Preprocessing

As we discussed, the input data of training/testing includes a set of WiFi APs along with their RSSI. Each data record w is a set of WiFi APs shown as $\{mac_1, rssi_1, mac_2, rssi_2, \ldots, mac_n, rssi_n\}$. However, this format of the data record cannot be directly used as the input of the WiFi positioning algorithm. We need to first query the central database and transfer each MAC address information to longitude and latitude pair.

In addition, the longitude and latitude system is a global system which is very sensitive in a minor change. It is not good to build up a machine learning model using a sensitive input that has very little change. Therefore, we need to convert the longitude/latitude to (x,y) coordinates. The input records are therefore converted to the Cartesian coordinate system using this formula [18].

$$x = R \times cos(lat) \times cos(lon)$$

$$y = R \times cos(lat) \times sin(lon)$$

R is the radius of the Earth, we use the approximate value 6378137 m. The result (x, y) usually is very big for most longitude/latitude pairs. Therefore, we select an original point as (0, 0) and normalize all the points to this coordinate by subtracting the coordinate of this original point. Finally, the record of input data is shown as $\{x_1, y_1, rssi_1, x_2, y_2, rssi_2, \ldots, x_n, y_n, rssi_n\}$.

3.3 Positioning Modeling

Weighted Linear Regression Model. As we discussed before, the input data is a set of WiFi APs and their RSSI. After preprocessing, each record contains a set of coordinates and RSSI. The input data X therefore can be presented by a $n \times 3$ matrix where there are 3 columns of features of longitude, latitude, and RSSI.

$$X = \begin{bmatrix} r_1 \times x_1 & r_1 \times y_1 & r_1 \\ r_2 \times x_2 & r_2 \times y_2 & r_2 \\ \cdots & \cdots & \cdots \\ r_n \times x_n & r_n \times y_n & r_n \end{bmatrix}$$

The outcome data $y' = [x, y, z]$ is GPS coordinates with accuracy of the measurement. A weighted multivariate linear regression model can be represented as (1) here.

$$y' = [x, y, z] = \begin{bmatrix} \beta_1 \ \beta_2 \cdots \beta_n \end{bmatrix} \times \begin{bmatrix} r_1 \times x_1 & r_1 \times y_1 & r_1 \\ r_2 \times x_2 & r_2 \times y_2 & r_2 \\ \cdots & \cdots & \cdots \\ r_n \times x_n & r_n \times y_n & r_n \end{bmatrix} + [b1, b2, b3] \quad (1)$$

From the dimension of y' and X, we could know the parameter vector is a $1 \times n$ vector.

Neural Network Model. The Weighted Linear Regression Model (1) assumes a simple model that each input element will be multiplied by one parameter β_i. But this assumption may not help us get the most accurate model. Therefore, we implement a fully connected Neural Network Model for our dataset and set the number of parameters as flexible.

Here we define a $m \times 3n$ hidden layer weight matrix θ.

$$\theta = \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_m \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \cdots & w_{13n} \\ w_{21} & w_{22} & w_{23} & \cdots & w_{23n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ w_{m1} & w_{m2} & w_{m3} & \cdots & w_{m3n} \end{bmatrix}$$

And we flat our input matrix as a $1 \times 3n$ vector. Thus, the output of first layer can be represented as (2).

$$h(1) = g\left(\begin{bmatrix} w_{11} & w_{12} & w_{13} & \cdots & w_{13n} \\ w_{21} & w_{22} & w_{23} & \cdots & w_{23n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ w_{m1} & w_{m2} & w_{m3} & \cdots & w_{m3n} \end{bmatrix} \times \begin{bmatrix} x_1 \\ y_1 \\ r_1 \\ x_2 \\ y_2 \\ r_2 \\ \cdots \\ x_n \\ y_n \\ r_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_m \end{bmatrix}\right) \tag{2}$$

Here g is an activation Relu function that we choose. The output of this layer $h(1)$ is a $1 \times m$ vector. We can therefore add the next layer after this layer using a $l \times m$ weight matrix to get a $1 \times l$ vector. And so on we can add more layers as (3) [19].

$$h(n) = [g\left(\Sigma_m \theta_{nm}\left[\cdots\left[g\left(\Sigma_j \theta_{kj}[g(\Sigma_i \theta_{ji} \cdot x_i + b_i)] + b_k\right)\right]\cdots\right] + b_n\right)] \tag{3}$$

4 Attack Overview

4.1 Goal of the Attacker

Here, we explore a novel attack against a smartphone positioning system by altering WiFi signals. The attack can be realized for two specific types. **Static Attack.** In a static attack, the attacker aims to spoof the smartphone positioning system to get a false position that is different from the real-world position. During this attack, the positioning result suggests a wrong location which is very far away from the user. **Dynamic Attack.** The attacker aims to send a false route data and the smartphone positioning system reports a false moving behavior of the current device. This may cause the malfunction of map applications and the user cannot go to the correct destination.

4.2 Attack Details

Data Collection: The static positioning data of other cities such as Paris, London are collected from the Wireless Geographic Logging Engine (WiGLE https://wigle.net/) website database through its public API. To collect the dynamic route data, an Android phone with WiFiManager API [17] is used for scanning WiFi SSIDs along a planned route B. The WiFiManager collects all the essential information about the SSIDs it discovers, such as RSSI, MAC, GPS positioning coordinates, timestamps, etc. The attacker collects the WiFi SSIDs by driving through the route B and record the SSIDs. Among these data, RSSI

and MACs are mainly used for the attack. The GPS positioning coordinates are used as the ground truth of the current position when observing certain SSIDs. We collected four sets of static positioning data with each has 100 WiFi APs from WiGLE. For the dynamic attack data, we repeatedly collect the driving route 10 times using an Android phone while driving a car. This route contains both WiFi dense and sparse areas with an average number of 42.3 WiFi APs in each location of this route. The average driving time for this route is 3 min and 32 s. There are a total of 8967 WiFi APs in this dataset. Each WiFi AP record has corresponse RSSI, MAC, and GPS positioning information related to its SSID.

Beacon Frame Packet Design: We use specifically designed beacon frame packets as the attack signal to represent the spoofing WiFi APs. As shown in Fig. 3, the beacon frame packet we designed is a 51-bytes concise packet without any optional parameters (e.g. DS Parameter Set, CF Parameter Set, IBSS Parameter Set). We also use only a null byte for SSID information therefore the WiFi SSID is hidden and will not be reported by the smartphone if the user is searching for nearby available WiFi networks. Thus, timestamp, source address, and BSSID field are the only 3 fields that need to be changed. During the attack, the MAC information of each spoofing WiFi AP is written in the source address and the BSSID field of the beacon frame packet, and the timestamp is updated according to the system timer.

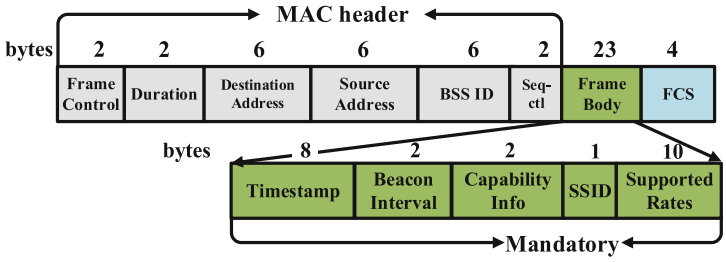

Fig. 3. The structure of spoofing beacon frame packet.

GPS Signal Jamming: Since GPS signal is primarily used for the positioning system, the legitimate GPS signal will affect the positioning system, which makes the system trust GPS location. The attacker first needs to block legitimate GPS signals. A GPS signal jammer is deployed for blocking legitimate GPS signals. The attacker uses a Universal Software Radio Peripheral (USRP) X300 for broadcasting the noise signal. The attacker uses a Gaussian noise signal to jam the GPS signal at 1575.42 MHz, which is the central frequency of Coarse/Acquisition (C/A) code of GPS information for civilian use.

Device Set Up: The attack is set up as shown in Fig. 4. The attacker first hides the GPS jammer (shown in Fig. 4 left part) in the car. Before launching the attack, the attacker places these portable ESP8266 devices around the victim and hides them (shown in Fig. 4 middle part). A typical attack scenario is to hide the device in the car as shown in Fig. 4 right part. A USB 5 V power is required for these devices, which can be easily set up by a portable USB charger. Firstly, the attacker uses his laptop to synchronize the timer of the attacker's devices using a synchronize algorithm as shown in Algorithm 1. After getting the same system timer, the attacker starts to broadcast the attack signals according to the location sequence as we mentioned above.

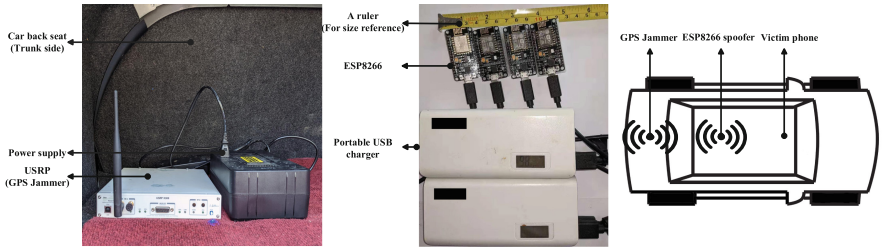

Fig. 4. The experiment setups for GPS jammer and ESP8266 spoofer

4.3 Attack Algorithm Design

The attack algorithm contains two key components: time synchronization for multiple spoofing devices and attack signal deploying. For multiple spoofing devices to work at the same pace, we design a time synchronization algorithm for multiple spoofing devices to send a part of the spoofing signal. After time synchronization, we send the attack signal through each ESP8266 device.

In our attack model, we use multiple ESP8266 devices to send out APs along with each point of the route. Each device sends a part of APs at each point. So every device will mimic a full route for the dynamic spoofing attack. To construct a dynamic route, we consider a WiFi AP set $\mathbb{W}$ is the total WiFi APs for the dynamic route. The full set $\mathbb{W}$ can be divided into a sequence of set $\mathbb{W}_1, \mathbb{W}_2, \ldots, \mathbb{W}_n$ according to the time sequence of WiFi APs. Thus, every subset is all the WiFi APs the device detect at one specific moment. In order to allocate all the WiFi APs to m available devices during an attack, we further divide each subset to m sets. Therefore, at time point 1, $\mathbb{W}_1$ will be divided into $\mathbb{W}_{11}, \mathbb{W}_{12}, \ldots, \mathbb{W}_{1m}$. Device 1 then sends WiFi beacon frames of $\mathbb{W}_{11}$; Device 2 sends WiFi beacon frames of $\mathbb{W}_{12}$ and so on. So at the next time point, Device 1 will send $\mathbb{W}_{21}$ and so on. To achieve this, every device needs to synchronize the time point to reconstruct the subsets $\mathbb{W}_1, \mathbb{W}_2, \ldots, \mathbb{W}_n$.

In order to synchronize the time for each device so that every device can send the data from the same point, we send the current epoch time number to ESP8266s from a laptop as described in Algorithm 1. At first, the device is

marked as not received and not synchronized. The attacker first broadcasts the current time e to every ESP8266. If the status of the device is not received and not synchronized, the device will first record the e and then compute the base time b that the device is turned on by e minus the time elapsed since the device turned on. This is because the ESP8266 has a built-in time counter that can count how long has the device turned on. Therefore, the device can always use a base time to synchronize its step with other devices.

Algorithm 1. External signal attack algorithm

Input: **e**: Start epoch time from a laptop
$\mathbb{W}$: Attack data set
n: Attack time length
Output: The current time of each device

Broadcasting e to each attacker's device
For each one of m devices **do**
$m \leftarrow$ device number
    If Not received
        Read e from laptop broadcasting
        received = True
    End If
    If Not synchronized and received
        $c \leftarrow$ time elapsed since the device turned on.
        $b \leftarrow e - c$
        synchronized = True.
    End If
    While synchronized is True
        $t \leftarrow b +$ time elapsed since the device turned on
        If $b > n$
            Break While
        End If
        send out $\mathbb{W}_{tm}$
    End While
End For
Return $b +$ time elapsed since the device turned on

With the above synchronizing algorithm, the attacker may launch the attack by using several ESP8266s. As we mentioned, the attacker collects the spoofing dataset by real-world driving or directly from the WiGLE database. To successfully attack the smartphone, the attacker needs the verification from the reverse-engineering model to make sure the victim's device will get the desire positioning result. Therefore, for a list of attacking data A, we design an attack signal deploying algorithm to verify each part of the data and broadcast the verified data as shown in Algorithm 2. When the attacking data A is loaded, the attacker first checks each part of A and goes through the reverse-engineering

model to verify the positioning result. If the positioning result is correct, the signal will be sent. Otherwise, that part of the data will be dropped. The attack can use this method to avoid some abnormal positioning results and being blocked by some defense mechanisms.

Algorithm 2. Attack Signal Deploying

Input: A, a sequence of collected WiFi APs
Output: Confirmed positioning location result

```
Pointer pt = start of A
While pt not meet the end of A
    location result lr = ReverseEng(pt− > aps)
    If lr == collected gps position
        deploy pt− > aps through ESP8266 according to the timestamp
    else
        drop current lr and aps
    End If
    pt = pt-¿next
End While
```

5 Attack Evaluation

Next, we evaluate the external signal attack on a smartphone-based setup on real-world data.

Attack Scenarios. We evaluate our attack by the following attack scenarios.

- *Static external signal attack.* This is the basic external signal attack. We want to spoof the positioning system of the smartphone to get another location that is different from the current position. In this case, we evaluate the relationship between the distance between the ground truth position and the spoofed location with the error of the positioning result.
- *Dynamic external signal attack.* This attack tries to spoof the smartphone and let it believe it is moving along a designed route while the smartphone is not following the route in the real world. In this attack scenario, we evaluate how well the smartphone following the designed route by the error of each designed point with the real positioning result.

Evaluation Metrics. The Wireless external signal attack aims to spoof the positioning system of the smartphone without raising an alert. The abnormal positioning result usually shows as a big error of positioning result and/or an impossible real-world location (e.g. A car is driving in the river). Therefore, how accurate the positioning result could be is an important metric to evaluate whether the attack can bypass the defense mechanism. We use the following metrics to evaluate the attack performance: (1) *Truth-designed distance.* This is

the distance between the ground truth location and the designed attack location. This is shown as the red arrow in Fig. 5. The ground truth location is the real-world location of the smartphone. The designed attack location is the position that the attacker wants the smartphone to believe it is at. (2) *Designed-result distance.* This is the distance between the designed attack location and the positioning result. This is shown as the yellow arrow in Fig. 5. (3) *Positioning result standard error.* This is the standard error reported by the positioning system for each result.

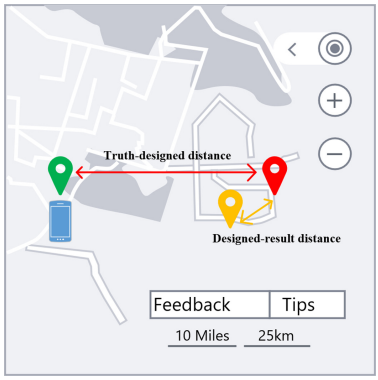

Fig. 5. Evaluation metrics: Truth-designed distance and Designed-result distance

Performance of the Static External Signal Attack: We conduct the static external signal attack using the spoofing signal collected from the WiGLE database. The attack is performed in a parking lot, the victim's smartphone is the author's phone and we place the attack devices in the author's car. As shown in Fig. 6, the smartphone positioning system is spoofed through our designed attack system and the user receives the false location results of other cities such as New York, Paris, London, and Berlin. The performance of the static external signal attack is shown in the left of Fig. 7. The y axis shows the designed-result distance and the error bar represents the positioning result standard error. As we can see the designed-result distance and positioning result standard error slightly increases when the truth-designed distance significantly increases (e.g. One country to another country). Compared to the control normal positioning error shown in the right of Fig. 6, both designed-result distance (mean 6.91 m) and positioning result standard error (mean 3.23 m) is within range of smartphone positioning result standard error control data (4.82 ± 5.93) m in 95% confidence interval which indicates the attack does not raise abnormal positioning result even when the attacker tries to spoof the smartphone to a very far location. Therefore, our study reveals the vulnerability that current smartphone apps will provide false location results to the user during the static external signal attack.

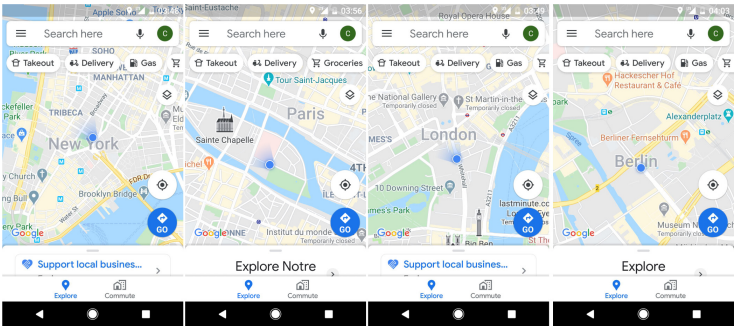

Fig. 6. Static external signal attack

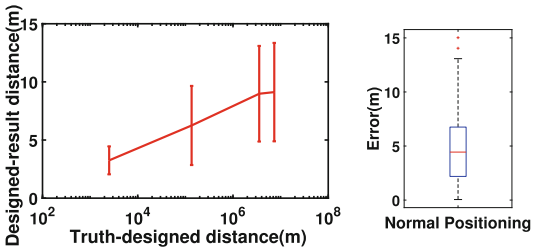

Fig. 7. Static external signal attack evaluation

Performance of the Dynamic External Signal Attack: In this attack, we collect the attack data as described in Sect. 3. The attack data contains all the WiFi APs in a 3 min 32 s length driving route as shown in Fig. 8 (A). Similar to the static attack environment, the attack is performed in the author's car. We choose a route with both WiFi dense and sparse areas. The average number of WiFi APs per location is 42.3 with a range from 6 to 100 as shown in the left of Fig. 9. We use four ESP8266s to synchronize this attack according to the algorithm 1. In this attack, an attacker is staying in the car at the start point without driving. The attacker blocks the GPS signal through GPS Jammer. He then uses the spoofing devices to broadcast the data of the route as shown in Fig. 8 (A).

During the attack, although the victim's smartphone is not moving, we can observe the positioning result shows the smartphone is moving according to the designed route of the attacker. In addition, to make sure the attack is successful, the spoofing WiFi signal RSSI is higher than the data we collected in the real world. This result can be observed from an RSSI histogram in Fig. 8 (B) and (C). We can observe a significant left shift of the RSSI peak, which indicates an increase of RSSI during the attack. Next, we want to evaluate the Designed-result distance, we notice that in the dynamic attack the number of WiFi spoofing signals is always changing and is highly related to the performance of the attack. We plot the number of WiFi APs with the Designed-result

distance in the left of Fig. 9. Our result shows a Designed-result distance in (3.35 ± 1.16) m in the 95% confidence interval. To evaluate this metric, we choose the normal driving data without attack to compare with the result as shown in the middle and right of Fig. 9. Since normal driving data has no designed-result distance, we set the designed-result distance as 0 and compute the confidence interval from the positioning error. The normal driving data could be represented as (0 ± 15.23) m in the 95% confidence interval. The t-test of these two groups shows $t = 1.639, df = 414, p = 0.102$, which indicates the two groups are not statistically significant. This result indicates attackers successfully spoof the smartphone positioning system and the positioning result is close to the attacker's designed location.

Next, in order to evaluate the positioning result standard error during a dynamic attack, we check the relation between WiFi spoofing signals and the positioning error. As shown in the left of Fig. 10, the positioning result standard error is inversely correlated with the number of WiFi APs used in the attack. The positioning result standard error is (9.26 ± 8.53) in 95% confidence interval, which is not significantly different with the normal control driving data (8.74 ± 6.44) m $t = 1.074, df = 414, p = 0.2834$ in the middle of Fig. 9. In addition, the CDF of dynamic attack data in the middle of Fig. 10 is similar to the normal control driving data in the right of Fig. 9. More than 90% of errors are below 15 m. This indicates during the attack, most positioning error is considered normal and the smartphone defense system may not notice that there is an attack. If we plot the error with the number of WiFi APs, the pattern suggests a logarithmic regression as shown in the right of Fig. 10. This result suggests that the attacker needs more WiFi APs to reduce the standard error of positioning results and achieve an accurate attack result.

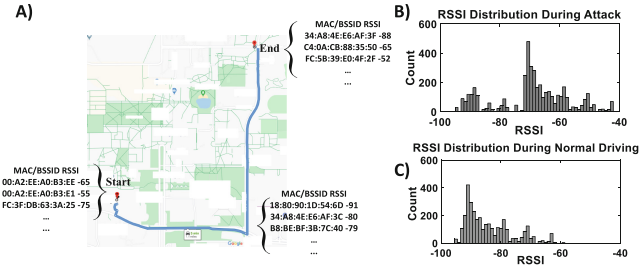

Fig. 8. Dynamic attack route and RSSI distribution

The current positioning system usually enables a position prediction model to predict the possible legit positioning result based on current position and speed. Therefore, we use a time-series long-short memory (LSTM) model to predict the legit position and compare it with the attacking result as described in [20]. The result is shown in Fig. 11. Here, the y-axis shows the distance travelled in prediction or the ground truth, which is based on the previous speed. The ground

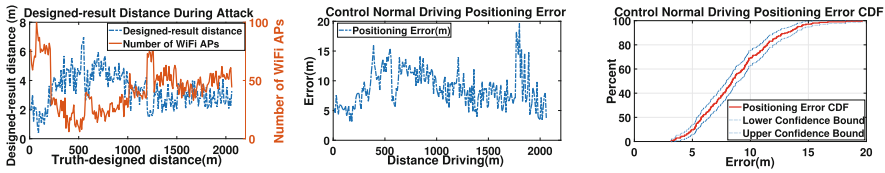

Fig. 9. Designed-result distance during attack and control normal driving positioning error

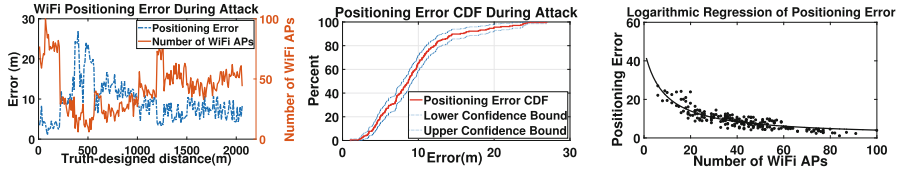

Fig. 10. Positioning result standard error during attack

truth matches the attack predicted with only an average error of 0.34 m. This error is much less than the threshold (5–10) m defined by GPS accuracy.

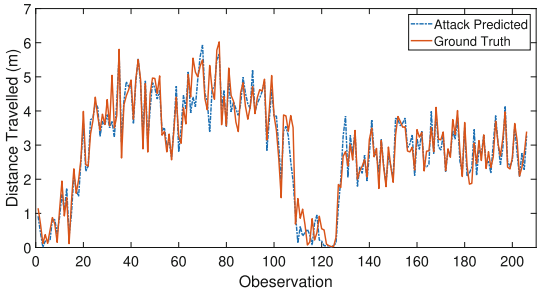

Fig. 11. The predicted position compared to the ground truth

6 Discussion and Defense Solutions

Our paper demonstrates the initial feasibility of manipulating the road navigation system through an external beacon frame spoofing attack. Based on our study, the threat becomes realistic as some users may not notice the navigation system when the user is not familiar with the current environment. Since the current smartphone navigation system is vulnerable to this external signal attack, it is important to explore the current smartphone system and find out an effective solution to this type of attack.

The main advantages of our proposed attack are the feasibility and the low-cost of the external dynamic signal attack. Our proposed attack requires only

four ESP8266s as the WiFi signal source, which only cost $2 each. Since the Skylift project already performed this attack in a static way, we further extend this attack in a dynamic way that can attack victims on a moving vehicle [13]. Thus, the attack can mimic both static and dynamic scenarios. Our attack uses WiFi AP beacon frames as the attack signal which can be very similar to legitimate signals. The current defense mechanism can hardly detect our attack. Thus our attack is also highly efficient. The main limitation of our attack is that we need to block the legitimate GPS signals. The legitimate GPS signals can alert the victim's device that there is a positioning attack. Our current attack uses a separated USRP to block the GPS signals. Yet, there are some more portable ways to perform this job using other devices such as HackRF One Software Defined Radio (SDR) [10]. Thus, future studies should find a more convenient way to perform the attack and block the legitimate signals. Another limitation of this study is that we use much stronger WiFi signals from the ESP8266 to overwhelm other legitimate WiFi signals. This is a low-cost way to perform the attack and it assumes that sources of legitimate WiFi signals are much far away from the attacker's ESP8266s. This is correct under our attack thereat model because the attacker hides ESP8266s around the victim. In order to fully eliminate the effect of other WiFi beacons, we can block most WiFi channels and only use one channel to attack the victim's phone. But this method will significantly increase the cost of the attack.

Recently, the Decimeter-level localization techniques including the angle of arrival (AoA), angle of departure (AoD), and relative time of flight (rToF) begin to apply to WiFi localization system [21]. However, during our experiments, we find that the Android smartphone has not applied these techniques and uses the Google geolocation API that is mainly based on the RSSI. Although there are many technical difficulties to spoof these systems, it is still possible that the attacker can extend our attack to these designed systems by carefully positioned the ESP8266s in different directions around the victims. Thus, further research to prove the feasibility of this attack to the AoA/AoD/rToF WiFi localization systems is needed.

The traditional defense mechanism focus on abnormal result detection. However, the state-of-art defense solution now enables the signal checking such as WiFi-hotspot tags [22]. The straightforward defense solution is to verify each WiFi AP and make sure it is a legitimate signal. This requires a legitimate signal filter installed on the smartphone. A device signal fingerprint can be used for verification. The fingerprint can include the frequency pattern of the WiFi AP based on the manufacturer shown in the MAC address in the WiFi packet, the historical WiFi AP beacon frame packet recorded in the database, and/or other physical information of the WiFi signal. However, the attacker may collect the real-world signal and try to mimic the fingerprint information and bypass this approach. In addition, the smartphone system may cross-check the data of the inertial measurement unit (IMU) sensors to confirm the actual movement of the smartphone and compare it with the geolocation data of the smartphone positioning system [10]. This defense method in general suffers from

accumulative IMU sensor errors and becomes ineffective as the time drifts. Also, [23] has demonstrated that IMU sensors are not sufficient to defend against location spoofing attacks under different scenarios.

7 Conclusions

In this paper, we explore the feasibility of external WiFi signal spoofing attacks the smartphone positioning systems. Real-world attack tests confirmed the attack effectiveness and vulnerability of the current smartphone system. We further discuss some possible defense solutions. We hope that our results can help in designing defense mechanisms for the current positioning system to the potential external WiFi signal spoofing attacks.

Acknowledgments. We appreciate constructive comments from anonymous reviewers. This work is supported by National Science Foundation (No 1553304).

References

1. Wahlström, J., Skog, I., Händel, P.: Smartphone-based vehicle telematics: a ten-year anniversary. IEEE Trans. Intell. Transp. Syst. **18**(10), 2802–2825 (2017)
2. Bell, S., Jung, W.R., Krishnakumar, V.: Wifi-based enhanced positioning systems: accuracy through mapping, calibration, and classification. In: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness, pp. 3–9. ACM (2010)
3. Zeng, K.C., Shu, Y., Liu, S., Dou, Y., Yang, Y.: A practical gps location spoofing attack in road navigation scenario. In: Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications, pp. 85–90 (2017)
4. Tippenhauer, N.O., Rasmussen, K.B., Pöpper, C., Čapkun, S.: Attacks on public wlan-based positioning systems. In: MobiSys. ACM, pp. 29–40 (2009)
5. Tippenhauer, N.O., Pöpper, C., Rasmussen, K.B., Capkun, S.: On the requirements for successful gps spoofing attacks. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, pp. 75–86. ACM (2011)
6. Paek, J., Kim, J., Govindan, R.: Energy-efficient rate-adaptive gps-based positioning for smartphones. In: MobiSys, pp. 299–314 (2010)
7. Alizadeh-Shabdiz, F., Jones, R.K., Morgan, E.J., Shean, M.G.: Location-based services that choose location algorithms based on number of detected access points within range of user device, 4 December 2007, uS Patent 7,305,245
8. Schmitz Weiss, A.: Exploring news apps and location-based services on the smartphone. Journalism Mass Commun. Quarterly **90**(3), 435–456 (2013)
9. Zeng, K.C., Shu, Y., Liu, S., Dou, Y., Yang, Y.: A practical gps location spoofing attack in road navigation scenario. In: HotMobile. ACM, pp. 85–90 (2017)
10. Zeng, K.C., et al.: All your gps are belong to us: towards stealthy manipulation of road navigation systems. In: USENIX Security 2018 (2018)
11. Cao, Y., Luo, Q., Liu, J.: Road navigation system attacks: a case on gps navigation map. In: ICC 2019–2019 IEEE International Conference on Communications (ICC), pp. 1–5. IEEE (2019)

12. Tippenhauer, N.O., Rasmussen, K.B., Pöpper, C., Capkun, S.: iphone and ipod location spoofing: attacks on public wlan-based positioning systems," Technical report/ETH Zürich, Department of Computer Science, vol. 599 (2012)
13. Harvey, A.: Skylift (2017). https://github.com/adamhrv/skylift
14. Kumar, P., Duraimurugan, G., Kumar, G.M., Logesh, R., MyvizhiPraveen, L.: Prevention and localization of mac address spoofing attacks in wireless networks (2016)
15. Grokhotkov, I.: Esp8266 arduino core documentation. ESP8266 (2017)
16. Susiripala, A.: Load testing an esp8266 (2017). https://arunoda.me/blog/load-testing-an-esp8266
17. Larimer, J., Root, K.: Security and privacy in android apps. Google Developers (2012)
18. Russell, C.T.: Geophysical coordinate transformations. Cosmic Electrodynamics **2**(2), 184–196 (1971)
19. Atanassov, K., Sotirov, S.: Index matrix interpretation of the multilayer perceptron. In: 2013 IEEE INISTA, pp. 1–3. IEEE (2013)
20. Rustamov, A., Gogoi, N., Minetto, A., Dovis, F.: Gnss anti-spoofing defense based on cooperative positioning. In: Proceedings of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020), pp. 3326–3337 (2020)
21. Soltanaghaei, E., Kalyanaraman, A., Whitehouse, K.: Multipath triangulation: decimeter-level wifi localization and orientation with a single unaided receiver. In: MobiSys, pp. 376–388 (2018)
22. Ye, A., Li, Q., Zhang, Q., Cheng, B.: Detection of spoofing attacks in wlan-based positioning systems using wifi hotspot tags. IEEE Access 8, 39 768–39 780 (2020)
23. Narain, S., Ranganathan, A., Noubir, G.: Security of gps/ins based on-road location tracking systems. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 587–601. IEEE (2019)