Fake GPS Defender: A Server-side Solution to Detect Fake GPS

Yu-Heng Chang
Department of Computer Science
and Information Engineering
National Central University
Taoyuan, Taiwan 32001
Email: bill0722001@gmail.com

Yan-Ling Hwang
School of Applied Foreign Languages
Chung Shan Medical University
Taichung, Taiwan 40201
Email: yanling_h@yahoo.com

Chih-Wen Ou
Department of Computer Science
and Information Engineering
National Central University
Taoyuan, Taiwan 32001
Email: chihwen.frankou@gmail.com

Chih-Lin Hu
Department of Communication Engineering
National Central University
Taoyuan, Taiwan 32001
Email: clhu@ce.ncu.edu.tw

Fu-Hau Hsu
Department of Computer Science
and Information Engineering
National Central University
Taoyuan, Taiwan 32001
Email: hsufh@csie.ncu.edu.tw

***Abstract*—Smart phones, tablets, and wearable devices are equipped with Global Positioning System (GPS) sensors in order to obtain devices geographical location. Many conventional network-based applications provide the specific content and service to users according to their locations. The correctness of the location provided by the device's GPS module is certainly important to these Location-Based Service (LBS) providers. However, most service providers are unable to effectively authenticate GPS values provided by their users. This becomes an issue because device users can manipulate GPS values with their desired latitude and longitude through installing the specific firmware on their devices. For a popular LBS game like *Pokemon GO*, fake GPS values bring negative impact on the system stability and the fairness among other service users. This issue is so called "Fake GPS" problem. In this paper, we propose a pure network-based detection solution for the LBS provider who has to verify the correctness of their users' GPS values. Our mechanism is based on Internet Control Message Protocol (ICMP), and is able to provide the detection precision to state/city-level by using location-IP mappings of devices' edge routers. As a server side solution, our approach makes the malicious GPS manipulators more difficult to perform the trick. According to the implementation and experiment, the major contribution of FGDefender is that it does have better detection precision. Its server-side nature of deployment is competitive as well.**

Keywords–Location; Fake GPS; Spoofing; Router.

I. INTRODUCTION

With the fast development of Internet and telecommunication technology, electronic devices are usually equipped with Global Positioning System (GPS) module, giving them the ability to perceive location of themselves. To catch up with the trend of Internet of Things (IoT), more and more network services rely on devices location, which forms a kind of network service: *Location-Based Service* (LBS). *Pokemon GO* [1] is a well-known example of LBS game for its heavily usage of the GPS sensor on mobile devices. This kind of network applications suffers from a specific data spoofing attacks. LBS servers accept all the GPS values obtaining from user devices by default, typically without additional mechanisms to check whether these GPS values have been manipulated or not. Such issue is usually referred as the Fake GPS problem. Lack of data authentication and integrity checking not only brings doubt about the system, but also gives chances to malicious users to spoof the location, which is known as the *Location Spoofing Attack* (LSA), to obtain illegal interests from the application. The Mobile Network Operator (MNO) may aware of where their subscribers are, but LBS providers still hardly acquire GPS information from the MNO, especially while the LBS provider is not a domestic regulated enterprise. Moreover, most of conventional IoT-related devices can assume that physical access, changing/reversing the firmware of the device, all require lots of effort and professional skills. Unlike conventional IoT-based applications, LBS applications running on smartphones, which usually belong to user themselves, only take relatively low effort for owners to manipulate the firmware on their devices. Many open and online resources are provided for smartphone owners capable of jailbreaking, rooting, flashing new custom ROMs on their smartphones. *LineageOS* [2] is a famous and easy-to-install third-party smartphone ROM for Android. For Apple's iPhone, those without the newest version of iOS can also be jailbreaked easily by just one click. All these conditions indicate that LSA can be conducted by malicious LBS users without too much cost. These methods allow GPS value manipulating before returning them to the LBS servers are listed below:

Android mock location: By surfacing developer options in Android, one can easily find the mock location option lying on the setting menu. Emulating GPS value is a partially build-in function in Android framework [3]. It is originally designed for the LBS application developer whose device is not equipped with GPS module. As a result, creating fake GPS result is not difficult since Android OS provides such function

 ISBN: 978-1-61208-658-3

for developing purpose.

Emulator: *BlueStacks* [4], and *Genymotion* [5] are well-known Android emulators running with custom Android ROMs. Those emulators are usually running on the desktop PC instead of the handheld device. Therefore, the build-in GPS module emulation is obviously necessary. Attackers who use these emulators to run LBS applications can easily spoof the location information by assigning desired GPS values to the emulator's configuration.

Xposed Framework [6]: Xposed is a framework that can change the behavior of the system and apps without touching any APKs. Xposed Framework is a famous tools using system call hooking techniques in order to control the system behavior. By replacing `/system/bin/app_process` in Android framework, most the sensor-related APIs can be intercepted and modified before returning it to the caller application. Other data provided by other sensors is also unconvincing.

Software Define Radio (SDR): Software defined radio [7] is a radio communication system implemented by means of software on a personal computer or embedded system. *HackRF* [8] is an open source SDR platform. Interfering GPS signals on hardware level helps malicious users to evade most of the software-based defense mechanisms for both Android and iOS devices. Due to the requirement for considerably resources and professional knowledge, few civilian cases other than military activities of such attack were found.

In this paper, a fake GPS defender (FGDefender) is proposed. It is a server-side, pure network-based LSA detect mechanism for the LBS provider who has to verify the correctness of their users GPS value. Most existing systems provide IP-location lookup by querying databases containing these IP registry data. This is static data mapping, and many IP address owners assign this field with the owners location. If the owner is MNO providing Internet access service across the country, the location is obviously different between the IP address owner and the practical user. Our mechanism will be based on Internet Control Message Protocol (ICMP), which is widely supported by most network equipments and systems. In addition, most modern detection methods can only authenticate GPS values with country-level precision. FGDefender can effectively increase the detection precision to state/city-level by using location-IP probings for the device's edge router. Because GPS values are authenticated by those features obtained from applications server side, It makes the GPS value manipulators (attacker) more difficult to perform the trick. Our study of FGDefender also analyzes several situations that may cause false positive and negative, and gives possible improvements for each case as well. The major contribution of FGDefender is that it has better detection precision and lower deployment cost. Meanwhile, it still also works along with other existing detection systems.

II. Related Work

Saroiu and Wolman proposed the location proofs [9] in 2009, a simple primitive that allows mobile devices to prove their locations to mobile applications and services. A location proof is issued by the wireless infrastructure, such as a Wi-Fi access point, to the device within the communication range. The proof can be transmitted by the device to the application that wishes to verify the location of that device. This approach verifies user location at the client side and depends on the deployment for the target infrastructure. The detailed comparison between this approach and ours is discussed in Section VI-A. Recent studies [10] [11] [12] deal with the fake GPS signal, which is one of the most common issues that is researched. Our approach aims at issues not restricted to the fake GPS signal. We also focus on the issue that users may use tools on the device to spoof GPS data for LBS applications.

To verify the location of a device, one potential idea is to check its IP-location mappings. For each IP address on Internet, there are databases maintaining registration data [13] [14] listing companies or organizations who owns these IP addresses. The registration data often includes the organization name, location, registration date, and administrators contact information. Users can obtain country, and state/city (probably), in the location field usually. Hence, most basic detect mechanisms to prevent LSA is to compare the location of the IP registration data with the location that the device claims. However, the registration data for IP address is not accurate, especially for IP address when the device operates inside the mobile network. As a case shown in Figure 1, a public IP address (114.137.156.248) of a device actually locates in Taoyuan. But, when we search the Whois database, this IP's location is marked as the location of the MNO's headquarter in Taipei. As the result, only the Country field in the registration database is relatively convincing. Such precision is not accurate enough for most LBS providers.

Figure 1. GeoIPtool database result for IP address 114.137.156.248

Some studies, like [15], are trying to use other sensors on the mobile devices to detect LSA. Devices with same moving path can represent similar pattern on gyroscope, accelerometer, magnetometer or other sensors. When it comes to such detection mechanisms, problems will occur: First, acquiring more sensors' information on the mobile device indicates more permissions required for accessing these sensors. In addition, since the GPS sensor can be manipulated by some system call hooking techniques provided by Xposed Framework, it means that values from other sensors can also be contaminated by the same skill. Due to the attackers ownership of the mobile device, our detection mechanism will not depend on any information provided by devices.

Another LSA detection [16] is trying to use devices that are adjacent to the target device. The main idea is that two

 ISBN: 978-1-61208-658-3

adjacent devices must be able to communicate with or detect some signals from the other if these two devices are close enough geographically. This can be done by making use of the random service set identifier on portal Wi-Fi hotspot, or the ability to communicate with other hosts under the same Network Address Translation (NAT) by a random private IP. Although turning on Wi-Fi hotspot on mobile devices can be achieved programmatically, it causes current network to be disconnected. Due to the fact that Wi-Fi can be functioned either on STA mode or AP mode, it means that users who are chosen to turn on the hotspots, are able to connect to Internet through Wi-Fi before the other devices find it. We believe that it results in lots of inconvenience to those affected users. Another disadvantages is that it is restricted by the Wi-Fi transmission range (which is often 10-50m with normal antenna). We have tried to use the Carrier-grade Network Address Translation (CGNAT) to identify whether two devices connect the same base station or the same CGNAT. Instead of using normal NAT established by Wi-Fi hotspots, we tried to apply the same method on telecommunication networks to eliminate these disadvantages. In the end, it came to a failure that most MNOs may not allow packets forwarded to a private IP address, even these packets are from the same base station.

III. System Design

FGDefender aims at effectively increasing the detection precision to state/city-level, and providing a server-side, pure network solution to LBS provider. As mentioned in Section I, traditional IP location database systems cannot guarantee the location precision which often assigned by the IP address owner in the registry. The original main idea of FGDefender is to verify whether the location of the user's IP address identical to the location which the user delivers to LBS application. Since the static mapping is not real time and precise, ICMP-based probing technology may satisfy the needs of FGDefender. Since mobile devices do not directly connect to Internet, they often do so by their MNO 3G/4G mobile networks or WiFi hotspots. This is the reason why the correlation between the edge router's IP and its location is focused in this study. Since the correctness of an edge router location is difficult to achieve, we use other claimed locations provided by nearby LBS users using this edge router. This design is based on an important assumption that most users are believed to be benign.

As a pure network solution, the main obstacle may be proxies on Internet. Both Virtual Private Network (VPN) and proxy cache service leave us a great number of unknown information behind the front server, and cause inaccuracy to our system. Hence, we verify and exclude known proxy service users from our system first. The reason is that users with the qualified network seldom to use LBS application through VPN service, since VPN cause obvious latency and inconvenience. Some LBS applications prevent users from using VPN or other network proxy service in their application agreements because of the service regional restriction. As a result, we consider such pre-condition of proxy service is acceptable.

The next step is to examine the edge router. As shown in Figure 2, it is unlikely that a device, using "Router A" (denote as R_A in following) as its edge router, has the location obviously far from the area where R_A is responsible for. Suppose that there are other normal users using the same LBS application, and we can get some of them who also use R_A as their edge router (shown in Figure 2 as green dots). By clustering location data of those other R_A users, the possible location of R_A can be inferred. We defined it as "responsible area" of R_A. Hence, among those devices with an edge router, the device who is obviously far away from others can be the possible LSA device we are looking for.

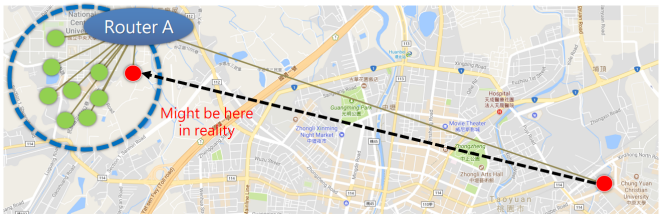


Figure 2. The responsible area of an edge router

When we focus on the result of one edge router, the possible result may be similar to Figure 3 (a). There is an obvious crowd and the attacker resides far away from others. The only way that the attacker can evade our mechanism is to purchase a lot more device or register lots of accounts for the LBS application, as shown in Figure 3 (b).

IV. System Implementation

FGDefender contains two major components. Each component runs a phase, as shown in Figure 4. The first component runs VPN detection phase, and the second component runs the database phase for maintaining edge router locations.

Phase 1: In the first phase, we verify if service user is connecting through a VPN or proxy server. Apart from black-listing some common VPN/proxy service, we use WITCH [17] or getIPIntel [18], both of them are open web services to help us detect and exclude the VPN/proxy users. Those web-based VPN/Proxy detection service basically check packets MTU with the help of a important principle: A normal MTU is 1,500 bytes, which can deliver 1460 bytes payload in one packet. VPN user was not able to transmit such a long data in one packet due to the extra header used by VPN delivering.

Phase 2: For each LBS application user, we perform reverse *traceroute* [19] to find out which edge router they are using. We built a database. The routers IP address is used as the key denoting which edge router we discover, and store the claimed location for each user. We record three closest routers from client to prevent the case that attackers control the network of the first router (edge router).

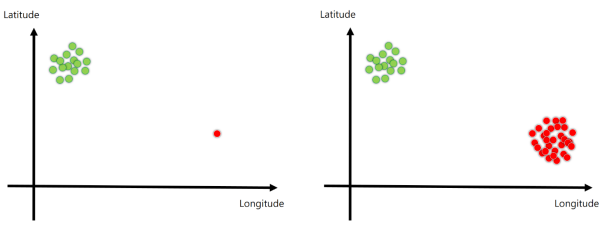


Figure 3. The way for attacker to defeat Fake GPS defender

 ISBN: 978-1-61208-658-3

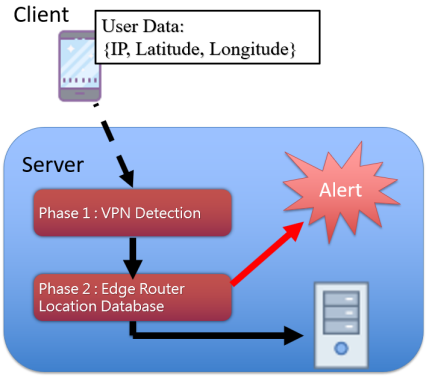


Figure 4. Fake GPS Defender system

For data analyzing, we try to find out outliers [20] from all users who have the same edge router. Using the mean value of longitude and latitude (gravity) as the reference point, we first calculate the distance between reference point and each users position. Second, after sorting all the distance, we can now obtain upper quartile (Q_3), lower quartile (Q_1) and interquartile range (IQR) of those collected data. Outliers (denote as x) in Tukey's range test [21] are often defined as:

$$x > Q_3 + 3IQR$$

$$x < Q_1 - 3IQR$$

Note that three times of IQR is usually called *major outliers*, which is a restriction in Statistics. As a result, we use this method to find out majority of outliers. For FGDefender, outliers are suspicious LSA attackers. Once FGDefender finds suspicious outliers, it removes the outliers data to prevent large amount of data coming from attackers.

V. Evaluation

In this section, we describe how we deploy and evaluate FGDefender. This experiment is to prove that users who change their location significantly, often result in changing to different edge router to connect to Internet.

A. Environment

In the experiment, we use two smartphones: Sony Xperia Arc S running on Android 4.0.2, and Sony Xperia X Performance running on Android 7.1.1 respectively, and a SIM which provides Internet accessibility from mobile network of Chunghwa Telecom, to gather GPS location and the corresponding network characteristics at a specific location. The detailed technical specifications of experiment are shown in Table I. We also build a server as the database in our laboratory, with one core CPU, 1GB RAM and 20GB hard disk divided from our VMware ESXi (Elastic Sky X integrated) workstation in our campus network.

B. Experiments

We developed and installed an app on both experimental smartphones. This app regularly switch the flight mode on and off, and then report genuine GPS values to our database server. We carried these smartphones to many different cities

TABLE I. EXPERIMENT MOBILE PHONES TECHNICAL SPECIFICATION

Model name	Xperia Arc S (LT15i)	Xperia X performace (F8132)
Android Version	Android 4.0.2	Android 7.1.1
Network tech.	HSPA+	LTE-A
Processor	Qualcomm MSM8255T Snapdragon S2	Qualcomm MSM8996 Snapdragon 820
RAM	512 MB RAM	3 GB RAM

in Taiwan. Once the server gets the reported data, it performed *traceroute* process to the smartphones IP addresses. We performed test on both computer networks and mobile networks, and examine the feasibility and accuracy of FGDefender.

C. Results

On the trip from Taipei to Taichung covering cities of Taipei, New Taipei, Taoyuan, Hsinchu, Miaoli, and Taichung, we collected 63 mobile network IP addresses under the 3G mobile network, which derives six edge routers. The result shows that we can obviously separate the northern Taiwan, which includes cities listed above, into two sub-areas, using these corresponding edge routers as shown in Table II

TABLE II. EDGE ROUTER RESULTS IN 3G MOBILE NETWORK

Place	Edge Router
Taipei-Hsinchu	210.65.126.161 210.65.126.209
Hsinchu-Taichung	210.65.126.193 210.65.126.185 220.128.24.237 220.128.25.169

In the 4G mobile network, we increase the GPS report frequency to making the experiment result more accurately because of the faster transmission data rate. As shown in Figure 5, there is obviously a strong correlation between the location of the users and edge routers they connect to. We collected 303 mobile network IP addresses, which also derives six edge routers. We then separate the northern Taiwan, including cities listed above, into three sub-areas, using corresponding edge routers as shown in Table III.

TABLE III. EDGE ROUTER RESULTS IN 4G MOBILE NETWORK

Place	Edge Router
Taipei	210.65.126.165 210.65.126.161
Taoyuan-Hsinchu	210.65.126.213 210.65.126.209
Hsinchu-Taichung	210.65.126.185 210.65.126.189

The reason causing the difference of reasonability area deployment between 3G and 4G mobile network is that there are fewer people using 3G mobile network than 4G mobile network. For the result of multiple reasonability areas, it must exist a boundary between two adjacent areas. As we performed in our experiments (Figure 7 and Figure 8), when the user

 ISBN: 978-1-61208-658-3

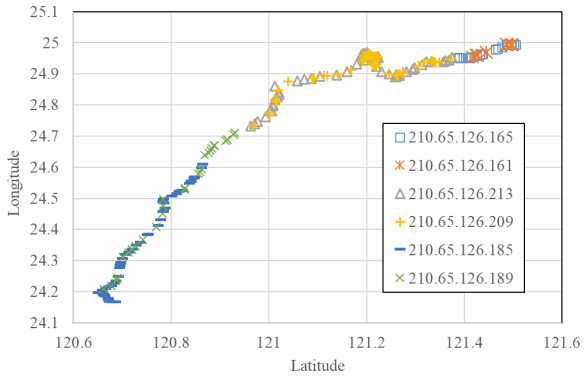


Figure 5. Edge router used in different positions in mobile network

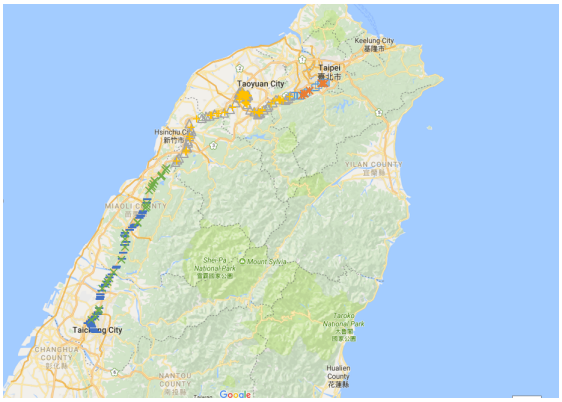


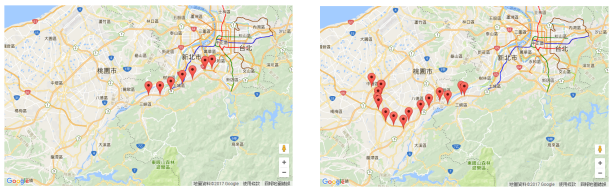
Figure 6. Mapping Figure 5 on Google Map

moves from Taipei to Taichung through the highway, the location that the edge router changes is in Sanxia Distinct. It denotes that the user who stays at the location marked in Figure 7 (a) are using 210.65.126.165 or 210.65.126.161 as the edge router to connect to Internet. The user stays at the location marked in Figure 7 (b) are using 210.65.126.213 or 210.65.126.209 instead. The changing point between Taoyuan-Hsinchu responsible areas and Hsinchu-Taichung responsible area fall in the Zhunan Township. It similarly means that users stay in Figure 8 (b) are using 210.65.126.185 or 210.65.126.189 as the edge router to connect to Internet. When devices handover to the base station (or eNB in LTE 4G networks), which belongs to another responsible area near the changing point, it results in an edge router switching. As a result, when mapping Figure 5 with a map as shown in Figure 6, we can finally figure out how MNO manage their IP address in the mobile network.

Due to the definition of outliers, the radius of the acceptable area we discover is $Q_3 + 3IQR$. It is often much larger than responsible area that the edge router actually manages. Therefore, for the case that client moving cross the boundary, he will locate in the overlapping region between two adjacent responsible areas, and cause little impact to our mechanism.

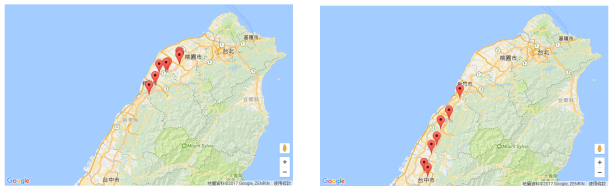
VI. Discussion

In this section, we focus on the accuracy, and potential enhancement if a larger responsible area is encountered. For



(a) (b)

Figure 7. Edge router switches from Taipei to Taoyuan



(a) (b)

Figure 8. Edge router switches from Taoyuan to Taichung

false negative cases, we specifically discuss this situation and the problems attackers need to deal with for this case. The accuracy of FGDefender depends hardly on the precision we can detect VPN or proxy users, the information of the users edge router we can discover, and the MNOs router and IP addresses management. If FGDefender fails to detect VPN/proxy users, those LSA users will not be detected when they connect through the proxy service. However, FGDefender forces the attacker not to spoof their location with a large distance from the position where the VPN/proxy server locates. Although the attacker successfully evades the VPN detection, they may still be detected once its spoofed location is out of the responsible area where its VPNs router should be. In other words, the attacker who wants to spoof their location must find an undetectable proxy server residing near that spoofed location first before performing fake GPS.

Another situation which might cause false negative case is that when we obtain a much larger responsible area of the mobile network router. Not all MNOs allocate different edge routers for different cities. If only very few sets of edge routers are used for the entire country, it means that there are very few responsible areas. Attackers can successfully spoof the location within cities in the same and large responsible area. Our later discussion of RTT Detection Enhancement is designed to deal with such issue. Attackers using mobile network should present a RTT value in a reasonable range according to other nearby normal users. Since the RTT value may dramatically change, the attacker who wants to evade the detection must emulate a reasonable RTT value for each spoofed location. Such enhancement can bring lots of effort and cost for attackers, and will be implemented in the future.

A. Comparison

The location proof, an approach mentioned in the section II, also provides location verification for LBS application. Compared to FGDefender, there are at least two major differences. The first one is that the location proof is a client-side approach, which means obtaining the trustable location data is done on the device. FGDefender does not use this

design because users can easily modify their device firmware, and making the location data untrusted today. The second difference is that the location proof required deployment to the infrastructure. FGDefender is a pure server-side approach, and no modification to infrastructure is required. FGDefender can be deployed with much lower cost than location proofs.

B. Limitation

ICMP-based approaches suffer from an obvious limitation that most network routers may discard ICMP packets due to security and performance reason. In our prototype system, only ICMP packet is used. More types of probing packets, such as UDP packets, can be used to mitigate this issue for future implementations.

C. Misc

Since the location of the user is considered as a part of privacy, there may be some solutions aiming at providing users functionalities of location privacy protection. In other words, users can decided whether to provide their location to others explicitly. As a server-side location verification solution, like FGDefender, it cannot verify the correctness of the location when users refuse to provide their actual location. But, such location privacy-preserving solutions are designed for stopping unauthorized data transmitted to unauthorized service providers, not for crafting fake locations to them. When users refuse to provide their location for a FGDefender-protected LBS, such identification performed by FGDefender will not be taking place. If the user provides the location, the identification should be conducted. The LBS administrator can receive a report explicitly listing no-location users and fake location users. The connection between the client and the server may suffer from Man-In-The-Middle (MITM) attack if the connection is not well-secured. This is the channel authentication issue, and is not the focus of this study. Most mobile devices support Assisted-GPS (AGPS) to enhance the location precision. According to our survey, there seems to be correlation between the IP addresses of edge routers and device location data from AGPS module. Such correlation is similar to the case of device GPS data, so that we do not distinguished GPS and AGPS data in this study. The device may sometimes obtain GPS location with errors, and causing problems for LBS application. However, most of such errors do not cause problems for FGDefender because FGDefender, which works based on the location of edge routers in the responsible area, can tolerate most margins of error generated by the imprecise GPS location.

VII. Conclusion and future work

In this paper, we proposed FGDefender, which is a server-side and network-based method to detect LSA users. We probe the IP address of user's edge router and its inferred location, which can be obtained from the application's server side, to authenticate whether the claimed location by the user is reasonable. As a pure network-based solution, the major contribution of FGDefender is that it has better detection precision and lower deployment cost. It also brings inevitable costs for LSA users. The experiment shows that the accuracy of FGDefender depends on the MNO's network configurations. More mobile network operators, as well as more locations, are planned to be included in the future.

References

[1] Pokmon GO Official Website, May 2018. [Online]. Available: http://www.pokemongo.com/

[2] LineageOS Android Distribution, May 2018. [Online]. Available: https://lineageos.org/

[3] A. Developers, "Location strategies," May 2018. [Online]. Available: https://developer.android.com/guide/topics/location/strategies

[4] BlueStacks, May 2018. [Online]. Available: https://www.bluestacks.com/tw/index.html

[5] Genymotion, May 2018. [Online]. Available: https://www.genymotion.com/desktop/

[6] Xposed framework, May 2018. [Online]. Available: http://repo.xposed.info/

[7] M. Ossmann, "Rapid radio reversing," May 2018. [Online]. Available: https://greatscottgadgets.com/tr/gsg-tr-2016-1.pdf

[8] greatscottgadgets.com, "Hackrf one," May 2018. [Online]. Available: https://greatscottgadgets.com/hackrf/

[9] S. Saroiu and A. Wolman, "Enabling new mobile applications with location proofs," in Proceedings of the 10th Workshop on Mobile Computing Systems and Applications, ser. HotMobile '09. New York, NY, USA: ACM, 2009, pp. 3:1–3:6. [Online]. Available: http://doi.acm.org/10.1145/1514411.1514414

[10] J. Magiera and R. Katulski, "Detection and mitigation of gps spoofing based on antenna array processing," Journal of Applied Research and Technology, vol. 13, no. 1, 2015, pp. 45 – 57. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1665642315300043

[11] M. R. Mosavi, A. R. Baziar, and M. Moazedi, "De-noising and spoofing extraction from position solution using wavelet transform on stationary single-frequency gps receiver in immediate detection condition," Journal of Applied Research and Technology, vol. 15, no. 4, 2017, pp. 402 – 411. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1665642317300652

[12] K. Jansen, N. O. Tippenhauer, and C. Pöpper, "Multi-receiver gps spoofing detection: Error models and realization," in Proceedings of the 32Nd Annual Conference on Computer Security Applications, ser. ACSAC '16. New York, NY, USA: ACM, 2016, pp. 237–250. [Online]. Available: http://doi.acm.org/10.1145/2991079.2991092

[13] GeoIPTool, May 2018. [Online]. Available: https://geoiptool.com/

[14] MaxMind, GeoIP2 Databases, May 2018. [Online]. Available: https://www.maxmind.com/en/geoip-demo

[15] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, "Inferring user routes and locations using zero-permission mobile sensors," in 2016 IEEE Symposium on Security and Privacy (SP), May 2016, pp. 397–413.

[16] F. Restuccia, A. Saracino, S. K. Das, and F. Martinelli, "Lvs: A wifi-based system to tackle location spoofing in location-based services," in 2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), June 2016, pp. 1–4.

[17] WITCH - detects OpenVPN via MSS values, May 2018. [Online]. Available: http://witch.valdikss.org.ru/

[18] getIPIntel, May 2018. [Online]. Available: https://getipintel.net/

[19] G. S. Malkin, "Traceroute Using an IP Option," RFC 1393, Jan. 1993. [Online]. Available: https://rfc-editor.org/rfc/rfc1393.txt

[20] G. J. Kerns, Introduction to Probability and Statistics Using R. GNU Free Documentation License, 2011, p. 44.

[21] H. Abdi and L. Williams, "Tukey's honestly signiflcant difierence (hsd) test," May 2018.

 ISBN: 978-1-61208-658-3