POSTER: Hide me if you can. Location Blurring on Firefox OS.

Marta Piekarska
Security in Telecommunications
Technische Universität Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
Email: marta@sec.t-labs.tu-berlin.de

***Abstract*—As smartphone become an indispensable parts of our life, the market for mobile applications also undergoes strong growth, with currently over two million applications in various online stores. More and more of those require access to the location of a user in order to enhance her experience. However, the information on where the device is might pose privacy threats. Geolocation may be used not only for targeted malware spreading, or phishing, but also to threat the user's physical security. We present a solution which allows the user to choose how precise her location is on per-application basis. We implement our solution as a part of an emerging web-based mobile OS, namely Firefox OS. The user can choose the granularity of the information given to the apps on several levels: precise, random, defined, as well as rounded to the city and country. Our idea also allows to define specific areas where the GPS should be turned off, or set to a specific position. The solution is flexible, and does not influence services which require full precision like the lost phone trackers, neither does it interfere with the laws of various countries.**

I. Introduction

Location-based services use the information about the geographical position of a mobile device to enhance the user's quality of experience (QoE). The integration of Global Positioning System (GPS) receivers into mobile devices made the task easy. Applications like FriendsZode (by AxisMobile), Enhanced 911 (FCC) or Thing Finder (Intel) allow for high-resolution determination of the current position. However, instead of improving the QoE, the location is often used for targeted advertising and malware spreading. Moreover, the collected data can be subject to interception and leakage - there are companies which gather and sell the information. Due to the possible threats, most mobile operating systems ask users to give their consent to reveal the location of their device to the app. On Android this decision has to be made during the installation phase, when the list of permissions required by the app is shown. The user can then make a binary choice of either accepting all or rejecting them, thus not installing the app. On iOS, on the other hand, whenever an application attempts to access the address book, calendar, location services and photo library, a pop-up appears on the screen asking for the consent of the user to share his location. There is a potential leakage problem: granting GPS access to the camera application that every photo will contain location data even when moved to e.g. social media. None of the existing mobile OSes allow to control precisely what information is shared with the apps.***Contributions.*** First we present a novel approach to the location blurring, which is user-friendly and does not interfere with local legislation. The algorithm we plan to use is resistant to most known attacks. Moreover, the control over the way the location sharing is handled is fine grained and flexible which allows for precise user control. We include not only blurring mechanism, but also user-defined position and, after identification of the threat to the GPS anonimization, namely the geoIP, we try to address the problem.

II. Background

Firefox OS Any app for Firefox Operating System (FxOS) is a web app, meaning it uses HTML5, JavaScript, CSS, and other Open Web technology. All system calls are done through the Web APIs. The applications are divided into three trust groups: certified, privileged, and web. The more trusted the app is, the bigger the group of APIs it can access. Every time an app calls an API, the system will consult the Manifest file to check if the permission was listed there, and if the type of the App (certified, privileged or web) is sufficient to grant the rights. Whenever the request is easy to understand, or impacts the privacy of a user, she is being asked to decide. She can allow the app to access the data once, always or not at all. If, however, the decision requires technical knowledge, it will be made implicitly during the review process in the Market Place.

Locating a device. Location of a device can be obtained in four ways. The most popular is usage of a GPS chip integrated in a mobile phone. The receiver listens to the signals containing the position of a satellite and the time when the message was sent. The transmission time of each message gives the distance of the satellite that sent the signal, and can be used in the navigation equations to get the precise location of the device. In cases where GPS signal is not available, or in order to increase the precision of the positioning, the location area and base station ID's can be used. Each mobile device keeps a list of unique cellIDs of the nearby base stations, and their signal strength, which allows to estimate the position of the phone. Third method, which has become quite precise throughout the last years is the usage of geolocation of the IP address. The IP address of a device it can be connected to the region it belongs to on a ZIP code level. Lastly the location could be reported by the user.

III. Privacy Concerns

As per the European Directive on Privacy and Electronic Communications, location-based services (LBS) must be permission-based. This means that the end user must opt-in to

the service in order to use it. In most cases, this means clicking the "accept" button when the application prompts for access to the location. However, the users do not get transparency what will happen with the obtained information. The same data might be used for enhanced QoE, or abused for potentially malicious or improper use, like advertisement, building user-profiles or even sharing with third parties. The notion of what is "private" differs quite a bit between people. Studies by Consolvo show that the people are most concern about *who* requests the information, *why* do they need it, and *what* level of detail would be most useful to the requesting unit. According to the data shared by Microsoft during Data Privacy Day 2011, 52 percent of the people express concerns about sharing their location details with others. Additionally, 49 percent stated that they will be more comfortable using LBS, if they could easily and clearly manage who sees their location information.

IV. Proposed solution

The solution we are proposing is a simple tweak to the original design of the FxOS webAPI architecture. Notably, it only changes what the apps can see, and not the way the GPS works. This is important in cases of emergency like locating of a stolen device or - even more important - locating of a lost or hidden person. As mentioned in Section II there are four ways to obtain user location. We only plan to interrupt the process on the GPS and geoIP levels. We do not influence the way the service provider records the position of the users during a phone call, which would be illegal in many countries. Normally on FxOS, the app calls the GeolocationAPI. This forwards the request for location to the GPS driver, which activates the GPS receiver. Once the positioning is finished the result is delivered to the webAPI, which then hands the information over to the app that requested it. Our solution allows the user to choose, on per-application basis, the granularity of the spacial-precision of the location given to the apps:

Turn Location Off allows the user to choose not to give any GPS data at all.

Give Precise Location leaves the system without any changes.

Choose a Position allows the user to fix his position to a set of coordinates. We provide a list of predefined values and a search that allows to find a City or Country (where the coordinates are set to the center of mass of the place). Additionally the user can enter his own GPS data.

Blur by X km here the user chooses the distance by which his position will be randomized. The choice is flexible and can vary from 1 to 500 km.

In the case of the first and third option the system does not even have to use the geolocation data. In case of the last choice the process is altered at a very late stage, after obtaining the precise geolocation data. In any case, once the GeolocationAPI is called, it checks the granularity of the information set for the calling app, adjusts the information accordingly and returns the altered result to the application. In FxOS there is no other way to obtain the position other than with the use of the GeolocationAPI, which means that the app cannot find out if the received information was changed in any way. Nor can it request more precise data. There are two ways that we consider to hide the location of the device - one uses a grid method, which means that the reported location will be always in the center of the grid's square. Second is usage of an algorithm we have designed based on the geohash algorithm. It will return a randomly chosen location within the chosen blurring granularity. The Geohash algorithm changes the longitude and latitude values into a binary and next to a base32 values. Depending on the required precision certain number of digits from the right are removed and the hash is decoded back to the geolocation values. We are also considering a third method, probably the best from security point of view, but most complicated from implementation point. It is a combination of grid and geohash algorithm, where user chooses the precision of the GPS values based on logical concepts: District, City, Country, Continent. The blurred possition is set to center of mass coordinates of chosen region.

V. Evaluation and Comparison

The problem of location blurring has been already considered by researchers, although none of the solutions have become part of one of the available OSes. The usual methods of tackling the problem are false dummies, landmark objects, spatial cloaking, usage of some middleware, or changing physical location to a logical one. All suffer from the same attack vectors: combining the traces of the user with the roadmap network, identification of the creator by observing the location they visit most frequently, replay attacks. Additionally the longer a user can be tracked the more distinctive the track becomes. In case of spatial cloaking the overlapping of regions is another problem. Issues that are rarely considered are the revealing of location with the use of geoIP. Even if the location is blurred on the API level, the apps can obtain the IP address of the device and use that data.

We plan to address the problems by finding an algorithm that will not be prone to the replay and roadmap attack. We will not have the problem of overlapping regions, as they will be fixed in our case. The overtime information gathering will be reduced by limiting the signals send to the apps when the device is staying in one place. We also consider allowing the user to define regions where the GPS would be turned off. All of the existing attacks do not work when user will choose option 3, i.e. when the actual GPS value is not taken into account.

VI. Conclusion

The proposed scheme is a simple solution to the privacy concerns of the users, yet it does not influence the usability. It is fully adjustable, which makes it very flexible and easy to use. Because the solution does not interact with the GPS sensors themselves it should not collide with any laws. Most importantly this will be the first location blurring service that will be part of a vanilla OS, which makes it not only an important addition to the end-users privacy, but also a good compromise between security and usability. Our main contribution to the field include (1) a novel solution of the problem of location blurring, (2) an algorithm that is resistant to the existing attacks, (3) advances in the field of usability by giving the user very precise and flexible control, (4) including the possibility of a user defined position, (5) identification of the threat of the geoIP locating, and addressing it.