An Analysis of Tracking Settings in Blackberry 10 and Windows Phone 8 Smartphones

Yogachandran Rahulamathavan[1], Veelasha Moonsamy[2], Lynn Batten[2], Su Shunliang[3], and Muttukrishnan Rajarajan[1]

[1] School of Engineering and Mathematical Sciences, City University London, London, U.K.
{yogachandran.rahulamathavan.1,r.muttukrishnan}@city.ac.uk
[2] School of Information Technology, Deakin University, Melbourne, Australia
{v.moonsamy@research.deakin.edu.au,lynn.batten@deakin.edu.au}
[3] Multimedia Information Technology, City University of Hong Kong, Kowloon, Hong Kong
{shunlsu2-c@my.cityu.edu.hk}

Abstract. The use of tracking settings in smartphones facilitates the provision of tailored services to users by allowing service providers access to unique identifiers stored on the smartphones. In this paper, we investigate the 'tracking off' settings on the `Blackberry 10` and `Windows Phone 8` platforms. To determine if they work as claimed, we set up a test bed suitable for both operating systems to capture traffic between the smartphone and external servers. We dynamically execute a set of similar `Blackberry 10` and `Windows Phone 8` applications, downloaded from their respective official markets. Our results indicate that even if users turn off tracking settings in their smartphones, some applications leak unique identifiers without their knowledge.

1 Introduction

Many service providers offer tailored services to their customers based on data gathered from the users' smartphones. Advertisements embedded within applications can be used to notify users of promotional offers in their nearby surroundings; however, several existing papers have shown that sensitive information such as the device's unique identifier and user's physical location are often leaked via advertising libraries without the device owner's consent [1]. In general, smartphone users are given the option to control the following two tracking services: location services and advertising. There are several papers in the literature analyzing these settings in `Android` and `iOS` based smartphones [2–6]. Some of this work indicates that information can be leaked through advertising.

In this paper, we ask the question: If tracking settings are turned off on `BlackBerry 10` and `Windows Phone 8` smartphones, is it possible that the devices are still tracked? On both of these platforms, to our knowledge, the default options that are provided by the smartphones have not been tested in any prior work. We test them in this paper by addressing the following specific questions:

W. Susilo and Y. Mu (Eds.): ACISP 2014, LNCS 8544, pp. 430–437, 2014.

- Can we verify if applications leak the location information of the smartphone when we turn off the *location services* and *advertising* tracking settings?
- Does any application access the smartphone's unique identifiers when we revoke the application's permission to access information?

The contributions of this work can be summarized as follows:

1. We implement a real-time traffic monitoring platform and demonstrate how to capture communication on Wi-Fi enabled smartphones.
2. We determine how well the tracking settings for *location services* and *advertising* work when they are turned off on a sample of `BlackBerry 10` and `Windows Phone 8` applications.
3. In the event where the settings fail to operate properly, we provide recommendations on how users can ensure that attackers do not compromise communications due to improper implementation[1] of Secure Sockets Layer (SSL) in applications.

The rest of the paper is organized as follows: Section 2 summarizes existing work and Section 3 provides background on tracking services. In Section 4 we explain our experimental work, followed by an analysis of our empirical results in Section 5. In Section 6, we conclude the paper and provide some recommendations.

2 Related Work

2.1 BlackBerry

The `BlackBerry 10` OS offers developers a list of permissions which can allow users to control the resources accessible to the applications once installed on the device. Unlike `Android` [5], many `Blackberry` permissions can be unchecked, prompted or revoked from application permissions settings on the `BlackBerry 10` devices. If an application tries to perform an action for which it does not have the required permission, the user is given a prompt. This often leads users to accept all permissions by default. `BlackBerry` reminds developers to have a highly visible privacy policy of their own, and ensure that they comply with local internet privacy legislation. This frees the `Blackberry` official applications site, BlackBerry App World, of any responsibility for privacy breaches by applications.

2.2 Windows Phone

The permission system on the `Windows Phone` platform bears many similarities to that of the `Android` OS. However, Microsoft's smartphone OS offers about 20 permissions for its application developers to which the users have to grant full access upon installation of the application.

Even though many publications have been written about the `Android` and `iOS` OS, to the best of our knowledge, there is little work in the literature analysing any versions of the `BlackBerry` and `Windows Phone` OS.

[1] `http://heartbleed.com/`

3 Tracking Services

We establish an experiment to determine if turning off tracking service settings actually prevents tracking. If we capture leaked information when the setting is 'off', it is not working as it should. Since no data capture when the setting is 'on' does not necessarily indicate a malfunction, we do not test the 'on' settings. We divide the tracking services into the following two categories: *location services* and *advertising*.

3.1 Tracking Services on BlackBerry 10

Location services can be accessed under the *Settings* option of a `BlackBerry` device. By default, the assumption is that whenever the setting for *location services* is turned off, applications should not have access to the user's location. To verify if this stands true, we monitor access to the following three pieces of information which are unique to the device and its user: (i) Media Access Control (MAC), a 12-character unique device identifier, (ii) Internet Protocol (IP) address and (iii) Global Positioning System (GPS) coordinates.

In order to test whether the tracking setting for *advertising* leaks any information when it is turned off, we monitor the use of two device identifiers that are unique to a `Blackberry` smartphone: (i) International Mobile Equipment Identity (IMEI), a 15-digit identifier and (ii) Hardware Personal Identification Number (PIN), an 8 alphanumeric identifier. These distinct codes are highly sought after by advertising companies for efficient user-profiling and to target advertisements.

3.2 Tracking Services on Windows Phone 8

We follow a similar rationale as described in Section 3.1. To test whether the `Windows Phone 8` device leaks location-related information when the *location services* setting is turned off, we observe the use of MAC address, IP address and GPS coordinates by installed applications. On the `Windows Phone` platform, users can switch off *Location* under the *Settings* option to deter access to location information.

As for advertising, users do not have any option on the actual device to regulate access to advertisements. Instead, they have to use their Microsoft account and visit the online 'opt out page'[2] to opt out of receiving personalized advertisements and prevent applications from sending unique device identifiers to 3rd parties. We survey the usage of the IMEI and Device Identifier (ID), an alphanumeric string, by installed applications after a user has opted out of communicating this information to external servers.

[2] `http://tinyurl.com/l2x8dyv`

4 Experiment

4.1 Dataset Collection

The experiment was conducted using applications downloaded from the official markets of `BlackBerry 10` and `Windows Phone 8` OS. To ensure that the experiment was consistent, we only considered applications which were developed by the same developer or the same company for both platforms. Since developer profiles cannot be publicly accessed on the application markets, we manually checked the developer's information for each application for both of the OS before including it in our dataset. Due to this constraint, we conducted our study with a small dataset of 40 `BlackBerry` and 40 `Windows Phone` applications.

4.2 Experimental Work

We set up a traffic monitoring test bed, as in [7], which is suitable for capturing information from any device using Wi-Fi connectivity as shown in Fig. 1. We used a BlackBerry Z10 smartphone (contributed by BlackBerry) running on the `BlackBerry 10` OS and a Nokia Lumia 520 smartphone running on the `Windows Phone 8` OS to test our dataset. The traffic sniffing tool, Mallory, was installed in a VM which was running on Ubuntu version 12.04. The Mallory tool was developed by the firm Intrepidus Group[3] and is capable of capturing traffic packets to and from the smartphones. We chose this tool as it facilitates the interception of SSL traffic and acts as a MiTM proxy to capture packets in real-time communication.

We faced the issue of unstable network connections at the beginning of the experiment, and, because traffic was being relayed through Mallory, we experienced lengthy delays which often resulted in IP addresses being reset. To counter this problem, we set up a dedicated mini Wi-Fi modem which allowed us to connect all our devices on the same network. The VM hosting the Mallory tool was allowed to connect to the Internet. In order to relay traffic between the smartphones and Mallory, with the `Blackberry` phone, we were able to set up a VPN, while at the time of the experiment, Microsoft had not yet implemented the VPN option on their smartphone OS. To bypass this issue, we used a Wi-Fi USB adapter which allowed us to carry out packet injections. This ensured that Mallory could piggyback on the traffic being sent to and from the Windows Phone 8.

As the smartphones and Mallory both share the same Internet connection, any Internet-based traffic on the smartphones can be captured by Mallory (see Fig. 1). The communication from smartphone to server is referred to as "c2s" and server to smartphone as "s2c"; the Mallory tool captures both. This information is then recorded in an SQL database which was later exported for further analysis. Since the aim of this experiment is to monitor information leaked by the smartphones when tracking services are turned off, we ignored the "s2c" communications and instead focused on the "c2s" ones as they are more likely to

[3] `http://intrepidusgroup.com/insight/mallory/`

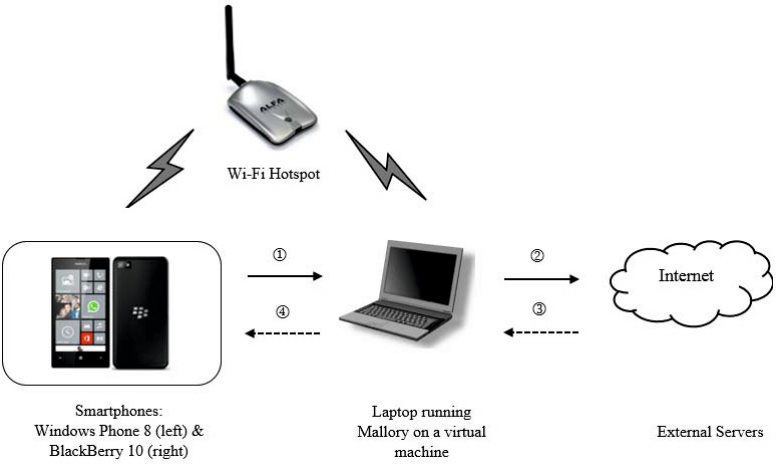

Fig. 1. Overview of Our Experimental Setup

reveal whether the applications on the smartphones are sending out information without the user's knowledge.

As explained in Section 3, we began by turning off all the tracking settings on our experimental devices and then analyzed our dataset by installing one application at a time. Each application was tested for two minutes. During the execution, we dynamically executed the application by checking all its features and clicking on advertisements. Once the execution time was over, we stopped Mallory from recording further traffic and uninstalled the application. We repeated these steps for each of the 40 `BlackBerry` and 40 `Windows Phone` applications. Once the experiment was concluded, we exported all traffic logs outside the VM and searched for the keywords mentioned in Section 3.

5 Analysis of Results

5.1 Presentation of Results

For each application in our dataset, we present our empirical results in two parts: *Location Services* and *Advertising*. The results for `BlackBerry` and `Windows Phone` are listed in Table 1 where the symbol ✗ means that one or more keyword items were leaked; alternatively, ✓is placed to demonstrate that the application did not leak any information when tracking settings were turned off.

Recall that in our choice of application, we specifically chose ones for each OS by the same developer and with the same name, therefore appearing to have identical functionality. However, Table 1 indicates that this is not the case. For instance, the 2nd and 23rd applications, namely **Tube Map** and **Poynt** from Table 1 are considered to be *consistent* as the results on both OS are identical. Conversely, the 1st and 3rd applications - **BBC iPlayer** and **Bible**

Table 1. `BlackBerry` and `Windows Phone` Applications with Tracking Setting Off (✓= Not leaked and ✕ = Leaked)

BlackBerry		**Application Name**	**Windows Phone**	
Location Services	*Advertising*		*Location Services*	*Advertising*
✓	✓	1.BBC iPlayer	✕	✓
✓	✓	2.Tube Map	✓	✓
✕	✓	3.Bible	✓	✕
✓	✕	4.Carrefour	✓	✓
✓	✓	5.To Do List	✓	✓
✓	✓	6.Combo Pic	✓	✕
✕	✓	7.Copter	✓	✕
✓	✓	8.Jetpack	✓	✓
✓	✓	9.Wizards Choice	✓	✕
✓	✓	10.Economic Times	✓	✓
✕	✓	11.Falldown	✓	✕
✓	✓	12.Real FootBall2013	✓	✓
✓	✓	13.Flashlight	✓	✓
✓	✓	14.Hangman	✓	✕
✓	✓	15.HDFC Bank	✓	✓
✓	✓	16.Kompass	✓	✓
✕	✓	17.Lega De Fulbol	✓	✓
✓	✓	18.Logo Quiz	✓	✕
✓	✓	19.Millionaire	✓	✕
✓	✓	20.Nu NL	✓	✓
✓	✓	21.OK Magazine	✓	✕
✕	✓	22.PGA Tour	✕	✓
✕	✓	23.Poynt	✕	✓
✓	✓	24.QR Code	✓	✓
✓	✓	25.Robotek	✕	✓
✓	✓	26.Skyscanner	✓	✓
✓	✓	27.Texas Holdem P.	✓	✓
✓	✓	28.Top Gear	✓	✓
✓	✕	29.Tuding	✓	✓
✓	✓	30.Tune in Radio	✓	✓
✕	✓	31.Wikipedia	✓	✓
✕	✓	32.WWE	✓	✓
✕	✓	33.XE Currency	✓	✓
✓	✓	34.Avianca	✓	✓
✓	✓	35.Chelsea FC NEWS	✓	✕
✓	✓	36.Daily Express UK	✓	✓
✓	✓	37.USA Today	✓	✓
✓	✓	38.Money Control	✓	✓
✓	✓	39.Toshl Finance	✓	✓
✓	✓	40.Park Mobile	✓	✓

respectively, are *inconsistent* as despite being the same applications executed on two different platforms, the results are different. In fact, there are 18 applications in our dataset that produced *inconsistent* results[4].

We believe there might be several reasons behind this discrepancy. During the experiment, each application was executed for a period of two minutes. Although this time constraint was applied to both `BlackBerry` and `Windows Phone` platforms, there is no guarantee that the execution patterns on each OS were identical. Moreover, despite the fact that all the applications in our dataset are available on both the `BlackBerry` and `Windows Phone` official application

[4] Application numbers 1,3,4,6,7,9,11,14,17-19,21,25,29,31-33,35 from Table 1.

markets and share application developers, we did not verify if the applications also make use of the same third party libraries. Different advertising libraries can appear in several posted versions of the same initial application because the business model of smartphones encourages developers to embed multiple third party libraries in order to increase revenue from advertising.

5.2 Information Leaks When Tracking Setting Is Off

Out of 40 applications executed on the `BlackBerry` platform, 10% (4) leaked either the GPS coordinates or IP address despite the setting for *location services* being off. These 4 applications are also part of the subset of 18 applications which produced *inconsistent* results, as described in Section 5.1. However, the results for *advertising* are far better as only 2 applications, **Carrefour** and **Tuding** leaked information about the Hardware PIN to advertisers when the tracking setting was turned off. It is also worth mentioning that none of the `Blackberry` applications leaked the IMEI or MAC address.

As for the `Windows Phone` platform, 35% (14) leaked information related to either *location services* or *advertising*. Whilst none of the 40 `Windows Phone` applications sent out GPS coordinates, the following four applications leaked either the MAC address or IP address: **BBC iPlayer**, **Robotek**, **PGA Tour** and **Poynt**. Unlike on the `BlackBerry` platform, the last 2 applications do not appear in the subset of 18 applications mentioned in Section 5.1. In terms of *advertising* related information leaked when the tracking setting was off, 10% of the `Windows Phone` applications sent out the Device ID information and the IMEI identifier was not divulged at all.

6 Conclusion

In this paper, we empirically analyzed 40 `Blackberry` and 40 `Windows Phone` applications. We tested whether tracking service settings for *location services* and *advertising* leak information when their tracking is turned off. We found that some applications still leak the user's location and device related information to third parties. Additionally, we observed that if an application does not leak any information on one particular smartphone OS, for instance `BlackBerry`, there is no guarantee that the same application will behave in a similar way on a different platform, (here for example, `Windows Phone`). Finally, we recommend some actions to overcome the issues we highlighted based on our empirical results.

6.1 Recommendations

Application developers earn revenue from in-application advertisements which is why many offer their applications free of charge. Advertising is very important for the smartphone application ecosystem as it is a major factor in the business model of the smartphone platform. Generally, the applications are required to send the smartphone's unique identifiers to advertizing agencies and in return,

the application developers earn a revenue for using that agency's advertizing library. Hence, there is a trade-off between convenience and user privacy. Therefore, we recommend the following:

1. Smartphone users should have easy access to adequate functionalities on their devices that will help protect their private information. As such, we recommend that Microsoft implements a setting on their smartphones to allow users to easily opt out of advertising, instead of doing so via the web.
2. Application developers should be obliged to list the names and owners of third party advertising libraries that are used in their applications. Users should be made aware upfront of the advertising companies that have access to their information.
3. Smartphone OS providers should ensure that when *location services* is turned off, no location-related information is revealed to third parties. This could either take the form of an additional check that is conducted when a new application is uploaded on the application market or after the fact, fines could be issued to deter application developers from unethically accessing such information.

References

1. Moonsamy, V., Alazab, M., Batten, L.: Towards an understanding of the impact of advertising on data leaks. International Journal of Security and Networks 7(3), 181–193 (2012)
2. Han, J., Owusu, E., Nguyen, L., Perrig, A., Zhang, J.: ACComplice: Location inference using accelerometers on smartphones. In: Proceedings of the 4th International Conference on Communication Systems and Networks (COMSNETS 2012), Bangalore, India, pp. 1–9 (January 2012)
3. Mann, C., Starostin, A.: A framework for static detection of privacy leaks in android applications. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC 2012), pp. 1457–1462 (March 2012)
4. Micinski, K., Phelps, P., Foster, J.S.: An Empirical Study of Location Truncation on Android. In: Proceedings of the 2013 Mobile Security Technologies Conference (MoST 2013), San Francisco, CA, pp. 1–10 (May 2013)
5. Shekhar, S., Dietz, M., Wallach, D.: Adsplit: Separating smartphone advertising from applications. In: Proceedings of the 20th USENIX Security Symposium (USENIX Security 2012), Bellevue, USA, pp. 1–15 (August 2012)
6. Zhao, Z., Osono, F.: TrustDroid: Preventing the use of Smartphones for information leaking in corporate networks through the use of static analysis taint tracking. In: Proceedings of the 7th International Conference on Malicious and Unwanted Software (MALWARE 2012), Puerto Rico, USA, pp. 135–143 (October 2012)
7. Moonsamy, V., Batten, L., Shore, M.: Can Smartphone Users Turn Off Tracking Service Settings? In: Proceedings of the 11th International Conference on Advances in Mobile Computing & Multimedia (MoMM 2013), Vienna, Austria, pp. 1–9 (December 2013)