Poster: Web Beacon Based Detection of Data Leakage for Android

GyeongRyun Bae
Dept. of Information Security
Seoul Women's University
Seoul, Republic of Korea
ryoon_9206@naver.com

Hae Young Lee
DuDu IT
Seoul, Republic of Korea
whichmeans@gmail.com

***Abstract*—This poster presents a web beacon based data leakage detection method, called B2-D2, in which data leakage carried out by spy apps can be detected by injecting web beacons into Android and tracing triggered beacons. Compared to the existing solutions, B2-D2 is very lightweight and does not require a system modification. Also, by injecting JavaScript tags instead of web beacons, B2-D2 would be able to exploit cross-site scripting vulnerabilities against attackers. Through preliminary experiments, we have confirmed that it works against many spy apps.**

Keywords—data leakage detection, mobile privacy, Android, web beacons

I. INTRODUCTION

Due to a dramatic increase of Android malware, especially spy apps, sensitive data theft from Android devices has become a major form of attack in recent years [1]. Although researchers have proposed a large number of solutions for detecting or preventing data leakage from Android devices, most of them are heavy, e.g., static and/or dynamic analysis based solutions [2,3], and/or need system modifications [4,5].

In this poster, we propose a web beacon based data leakage detection method (B2-D2) for Android. B2-D2 has focused on the detection of data leakage carried out by spy apps, e.g., listed in [6]. Since most spy apps provide websites for checking stolen data remotely, B2-D2 makes 'data' of web beacons, such as HTML image tags that are traceable. Upon checking stolen data through such a website, the beacons included in the stolen data may be triggered, so that the data leakage can be detected. B2-D2 is very lightweight; a user just need to install an app that injects web beacons on Android, and, if needed, may uninstall the app right after the injection. Also, B2-D2 does not require any system modification, such as kernel rebuilding. Furthermore, B2-D2 may be used to strike back against attackers, by exploiting cross-site scripting (XSS) vulnerabilities.

II. B2-D2

As shown in Fig. 1, B2-D2 uses (a) an Android app that injects web beacons and (b) a web server that tracks the injected beacons. The B2-D2 app first injects web beacons into common data such as contact information, calendar information, and media files on an Android device, through the content providers supplied by Android. Fig. 2 shows an example of a web beacon that was injected into a short text message. Note that each beacon would have a unique name in the final release.

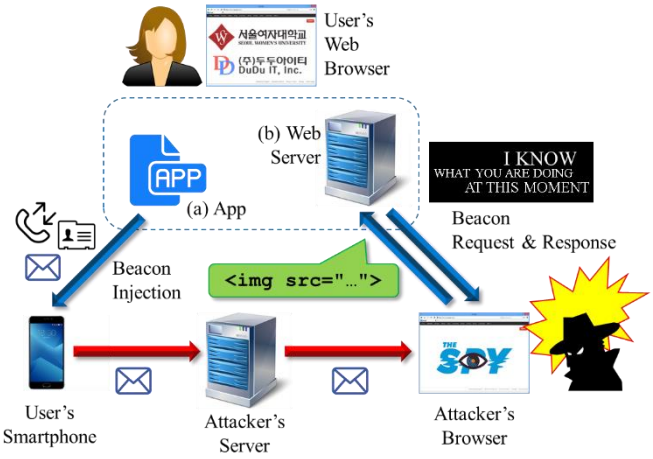

Fig. 1. Overview of B2-D2

Fig. 2. Example of web beacons

Fig. 3. Beacon awakens

Assume that the phone has been infected with a spy app, by an attacker. Then, some of the common data, together with some of the injected beacons, may be leaked (stolen) to the attacker's server(s) sometime, through the spy app. Most spy apps provide websites for checking stolen data remotely, so that the attacker would check the stolen data through a website, with a web browser. Due to the leaked beacons included in the stolen data, this may result in sending HTTP requests for the corresponding images on the B2-D2 web server (i.e., the beacons may be triggered). Fig. 3 shows an example of triggered beacons. In the final release, web beacons would be made invisible by using a transparent image, an inline frame (iframe), or a JavaScript file.

Thus, the B2-D2 user may detect the leakage of some data on the device, the IP address of the attacker's computer, the time the leaked data were checked by the attacker, and so on, through the B2-D2 web server. Also, the user may detect whether the spy app provider, if any, checked the leaked data. We have conducted preliminary experiments with some spy apps and confirmed that B2-D2 works against many spy apps (e.g., Figs. 3 and 4).

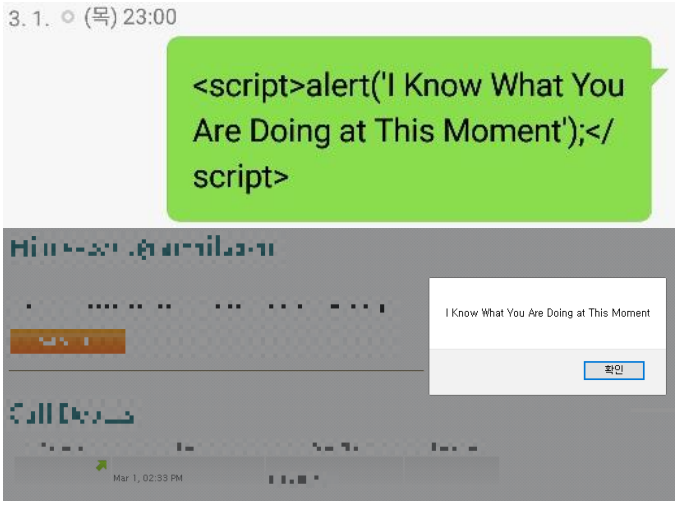

Fig. 4. Victim strikes back with a JavaScript tag

III. The Victim Strikes Back

If B2-D2 works, the user could have a new hope; JavaScript code may be injected and used to strike back against the attacker, by exploiting XSS vulnerabilities on the website provided by the spy app. For example, we may be able to: simply warn the attacker, manipulate some of the leaked data, forward to a phishing site, and so on. Through our preliminary experiment, we have also confirmed that many websites for spy apps have XSS vulnerabilities, which may be exploited by B2-D2, as shown in Fig. 4.

IV. Conclusion and Future Work

In this paper, we presented B2-D2, a lightweight method for detecting data leakage from Android. In B2-D2, data leakage carried out by spy apps can be detected by injecting and tracing web beacons. However, B2-D2 has several limitations: 1) Data leakage cannot be detected if attackers do not check stolen data or check them through other channels (e.g., consoles or text based browsers). 2) It is easy to be defeated (e.g., by secure coding). 3) Due to injection, the B2-D2 app may be blocked by Google, as in our previous work [7].

Nevertheless, thanks to its simplicity, B2-D2 could be an efficient and effective countermeasure against spy apps until it is defeated by all of them. Therefore, we will develop a mature version of B2-D2 and study a XSS based counterattack against each spy app. Another approach to overcoming the B2-D2's limitations will be also investigated.

References

[1] S. Wu, P. Wang, X. Li, and Y. Zhang, "Effective detection of android malware based on the usage of data flow APIs and machine learning," Information and Software Technology, vol. 75, 2016.

[2] H. Chen, H. Leung, B. Han, and J. Su, "Automatic privacy leakage detection for massive android apps via a novel hybrid approach," in IEEE ICC, 2017.

[3] L.H. Tuan, N.T. Cam, V.H. Pham, "Enhancing the accuracy of static analysis for detecting sensitive data leakage in Android by using dynamic analysis," Cluster Computing, vol. 2017, 2017.

[4] J. Bell and G. Kaiser, "Phosphor: Illuminating dynamic data flow in the jvms," in ACM OOPSLA, 2014.

[5] W. Enck, P. Gilbert, B.G. Chun, L.P. Cox, J. Jung, P. McDaniel, A.N. Sheth, "TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones," in USENIX OSDI, 2010.

[6] Best Cell Phone Spy Software Reviews 2017. http://www.bestphonespy.com/

[7] S.Y. Choi, J.A. Lee, W. Lee, H.Y. Lee, "COIN-VASE: Code Injection Vulnerability Scanning Environment for HTML5-Based Android Apps," in MobiCASE, 2016.