International Journal of Advanced Smart Convergence Vol.6 No.1 1-8 (2017)
https://doi.org/10.7236/IJASC.2017.6.1.1

IJASC 17-1-1

Simulated Dynamic C&C Server Based Activated Evidence Aggregation of Evasive Server-Side Polymorphic Mobile Malware on Android

Han Seong Lee[1], Hyung-Woo Lee[1†]

[1]*Division of Computer Engineering, Hanshin Univ., Rep. of Korea*
[†]*hwlee@hs.ac.kr*

Abstract

Diverse types of malicious code such as evasive Server-side Polymorphic are developed and distributed in third party open markets. The suspicious new type of polymorphic malware has the ability to actively change and morph its internal data dynamically. As a result, it is very hard to detect this type of suspicious transaction as an evidence of Server-side polymorphic mobile malware because its C&C server was shut downed or an IP address of remote controlling C&C server was changed irregularly. Therefore, we implemented Simulated C&C Server to aggregate activated events perfectly from various Server-side polymorphic mobile malware. Using proposed Simulated C&C Server, we can proof completely and classify veiled server-side polymorphic malicious code more clearly.

Key words: *Android, Mobile Malware, Dynamic Fake C&C Server, Server-Side Polymorphism, Transaction Proof, Evidence Aggregation*

1. Introduction

A newly emerging malicious mobile code called Server-Side Polymorphism (SSP) has been developed to detour detection system such as the Android Mobile Vaccine. In detail, Server-Side Polymorphism is a technique used by malware attacker as an attempt to evade detection by anti-virus software because a suspicious polymorphic (literally "many shapes") malwares change its internal data through obfuscation and encryption ensuring that no sample looks the same [1,2]. Therefore, polymorphic mobile codes also change constantly to evade detection steps by continuing its evolution after adding new propagation vectors, functionality and by using stealth techniques to hide its presence and evade the detection of antivirus software [3,4].

Recently, complicated polymorphic mobile malwares by adopting malicious server-side polymorphism have been developed to perfectly evade anti-virus vaccine detection procedure on Android through obfuscation and complex encryption by continuously modifying its internal code of mobile application [1,2]. A user is cheated to download android application from 3rd party app store as the installed mobile apps are updated after downloading malicious code from Command and Control (C&C) server activated by using attacker-oriented

Manuscript Received: Jan. 20, 2017 / Revised: Feb. 5, 2017 / Accepted: Feb. 28, 2017
Corresponding Author: hwlee@hs.ac.kr
Tel: +82-31-379-0642, Fax: +82-31-379-0238
Division of Computer Engineering, Hanshin University, Korea

server-side polymorphic malware. Because the mobile apps downloaded by each user who undergoes this process are installed with slightly different code, those mobile apps can't be detected though conventional mobile anti-virus S/W [3,4]. Therefore, it is necessary to develop an advanced detection system using simulated C&C server based real malware analysis mechanism against complicated server-side polymorphic android malwares. The C&C server generates a different type of malicious code and stores it in the app download server. Then, the user downloads a different type of malicious code from the pertinent server every time and installs it in the user's device. Because the mobile apps downloaded by each user who undergoes this process are generated with slightly different code, they have the characteristic of not being detected though conventional mobile detection methods. Generally, it was very hard to detect or classify suspicious mobile apps as a Server-Side Polymorphic code for lack of sufficient evidence [7,8,9].

As a result, it is necessary to detect suspicious server-side polymorphic attack by aggregating uncompleted and remained transaction using dynamic and simulated C&C Server. By implementing pseudo dynamic C&C Server, we can gather more detailed circumstantial evidences or proofs for classifying suspicious apps as a Server-side polymorphic mobile code.

2. Server-Side Polymorphic Mobile Malware

2.1 Detailed Mechanism of Server-Side Polymorphic Mobile Malware

New type of malicious code such as Server-Side Polymorphic malware is being developed and expended in third party open market. This new type of malware repeatedly mutates itself to evade from anti-virus detection by constantly adapting evading filtering or detection procedures. Recently, smart mobile attacker used polymorphism to generate several variants in which the malware would morph itself into diverse mobile application successfully to by-pass existing signature-based detection. Therefore, it is very hard to detect Server-Side polymorphic mobile malware because the malware analyst does not know a detailed mechanism[10].

The existing Server-side polymorphic mechanism looks like follow Figure 1. Once infected, the user's computer sends registering information to a C&C server. The C&C server then replies with a set of commands to execute on the victim's computer. A new piece of malware is generated by a "Polymorphic Generator", which re-packs or re-encrypts it with a randomly generated key. This technique ensures that the malware is unique giving it a significant advantage – it will never have been caught and analyzed by malware researchers. This vastly increases the likelihood that it will not be detected. The attacker can choose to scan the newly created copy with popular antimalware products to verify that no detection occurs. Once the copy is generated and verified as not being detected, it is stored on a "Download Server" and the link is sent to the victim [2].

When server-side polymorphic malware code is installed in each user's device, it operates such that it has mutually different signature values. Therefore, in the devices of different users, it is installed with values different from the known signatures of malicious codes, and consequently, it has the characteristic of avoiding the detection process of mobile detection methods. By applying an encryption for the string information value declared in the internal codes, it is hard to detect an obfuscation function from suspicious metamorphic mobile apps. Furthermore, they are devised to perform different types of execution processes every time by applying an advanced mechanism, such as changing the execution sequence for internal codes of the mobile app through the use of random numbers selected arbitrarily. Therefore, when this function is applied, they show a characteristic of detecting avoidance through mobile anti-virus vaccines [10,11,16,17].

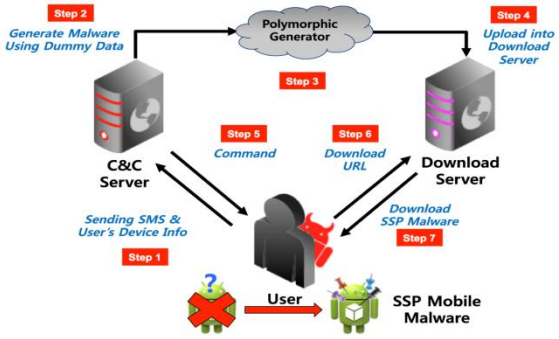

Figure 1. Mechanism of Server-Side Polymorphic Mobile Malware

3. Difficulty on Detecting SSP Mobile Malware

3.1 Complicated Transaction of Server-Side Polymorphic Mobile Malware

Generally, we can gather and analyze suspicious Server-Side Polymorphic mobile code using static and dynamic analysis mechanism. A user unknowingly goes through the registration process with a C&C server assigned by an attacker. Then the C&C server generates a different type of malicious code and sends it in the app download server. After this generation process, the user downloads a different type of malicious code from the server and installs it in his/her device. Because the mobile apps downloaded by each user are generated with slightly different code, it is very hard to classify or detect by through conventional mobile detection methods. The main reason is that these kinds of polymorphic malware have an additional component after performing a mutation process by using obfuscation techniques such as inserting junk code, reordering instructions and using mathematical contrapositives. Therefore, this type of malware still can't be still recognized by anti-virus software easily.

3.2 Difficulty on Transaction Aggregation from Suspicious SSP Mobile Malware

On performing both static and dynamic analysis process on suspicious mobile application, it is very hard to detect network connection or original transactions of Server-Side Polymorphic mobile malware perfectly because the original C&C server used by malware was shut downed or its IP address was changed to another one for evading detection. In both case of DevilsCreed and FlashPlayer malware, the connection to its own C&C server was not established because the C&C server was shut downed as follow Figure 2. Therefore, it is necessary to establish Fake C&C Server to aggregate suspicious server-side polymorphic transactions activated from mobile malware. If we implement fake C&C instead of real malicious C&C server, it is possible for us to aggregate complete evidences for proofing the overall transactions and for verifying its malicious activities completely[11,16].

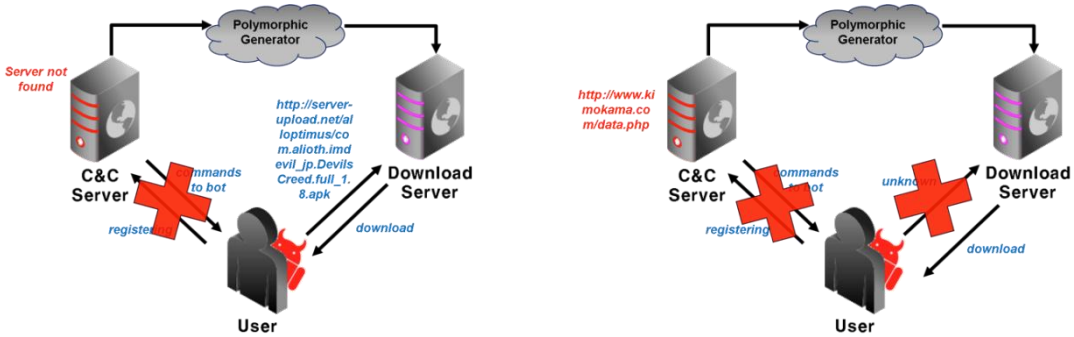

Figure 2. DevilsCreed and FlashPlayer Server-Side Polymorphic Transaction

4. Implementation of Simulated Dynamic C&C Server

4.1 Fake C&C Server for Detecting Server-Side Polymorphic Mobile Malware

Previously, we constructed a fake C&C server to aggregate real-time evidence activated from server-side polymorphic mobile apps. As a follow Figure 3, we can provide fake connection with suspicious malicious apps using our Fake C&C Server. The main idea of constructing Fake C&C Server will be start by modifying hosts file's contents stored on Android device. The 'hosts' file has been used to perform ill-intentioned Pharming attack. But, it also can be used to detour IP address of original C&C server to that of fake server. By modifying IP address on hosts file, the malicious mobile apps can be connected with Fake C&C Server. By using ADB pull command, we could get hosts file from '/system/etc/hosts' folder. Then, we modified hosts file by masquerading IP address of original C&C server into that of Fake C&C Server using common text editor. After that, we can update hosts file by using ADB push command.

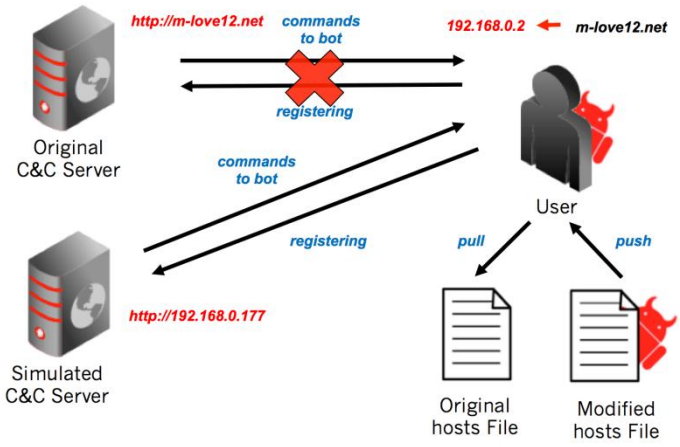

Figure 3. Previous Fake C&C Server based Evidence Aggregation Mechanism [11]

However, we can't gather really activated evidence from SSP malware because the original C&C server didn't operate. Therefore, we only can aggregate partial transaction activated from suspicious mobile malware by implementing previous fake C&C server. As a result, we can aggregate detailed evidence for dynamic analysis on suspicious mobile malware by additionally implementing simulated C&C server for gathering those uncompleted transactions correctly.

4.2 Implementation of Simulated Dynamic C&C Server

We implemented simulated dynamic C&C Server instead of previous fake one as follow Figure 4. Suggested dynamic C&C server will provide both the registration and responding function by providing integrity functions of download server. To doing these transactions, this simulated C&C server has to provide both GET and POST script for performing as an original C&C server. Additionally, simulated C&C server performs a function of existing download server instead by including the function of fake download server. If this simulated server receives App mutation request, then this server sends SSP download script generated from suspected mobile malware. As s response, masquerading script on simulated C&C server will send updated application to suspicious mobile malware instead of original C&C server.

In the index.php in the simulated C&C server, we set to collect both the client (emulator) IP address and user agent information. The log file stores the IMEI values passed by IP, User Agent and Malware. The malicious mobile code is configured to receive an updated application's apk file from the fake download server after receiving a JavaScript-style command. We upload another malware(FlashView.apk) to download to the Fake

Download Server. And then, complicated code-reversing step was required to implement simulated dynamic C&C Server. We already acquired java source core such as Boot.java, ConFigurejava, CRCTimer.java, MainService.java, R.java, Site.java and SMSReceiver.java from Dex file of FlashPlayer.apk. In the de-compiled source code, we could find GET message string, which are used to send command and phone number with its version information from MainService.java. In case of this GET message, we could identify real URL address of original C&C server. And we could extract message sending format based on GET message string, which are used to send SMS message and device information from SMSReceiver.java. And then we can additionally manuscript dynamic web server code for downloading mutated application from this simulated C&C server as Figure 4. As a corresponding response, we add return script (echo message) that will be sent from simulated C&C server to mobile apps. The meaning of return script is that simulated C&C server will give the command to send download link on each user's mobile phone. To verifying overall transaction correctly, we recorded transaction log data into 'log.txt' file.

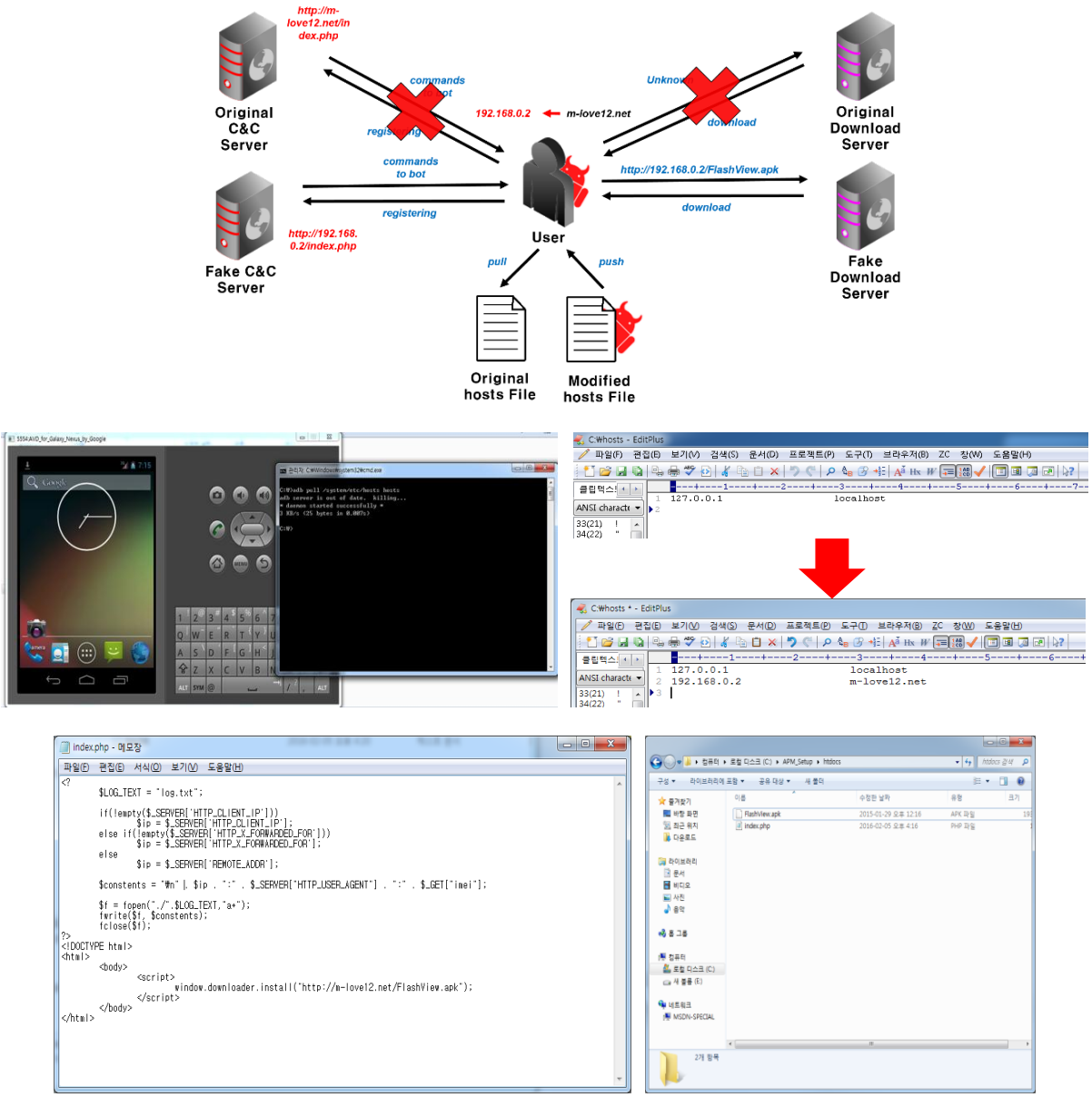

Figure 4. Masquerading Scripts to Perform as a Simulated C&C Server

5. Evidence Aggregation for Detect SSP Malware with Simulated C&C Server

5.1 Dynamic SSP Malware Reproduce with Simulated C&C Server

Based on those masquerading scripts, we can monitor real-time transactions activated from suspicious SSP mobile apps. Just after executing suspicious mobile application (Russian SSP Mobile Application) on emulator, we can find first log data sent from simulated C&C server as a result of requesting registration. Additionally, we can find that application version information like "1454656801585.apk" was finally downloaded from the simulated C&C server after install malicious "Adobe Flash Player 11" application simultaneously as follow Figure 5.

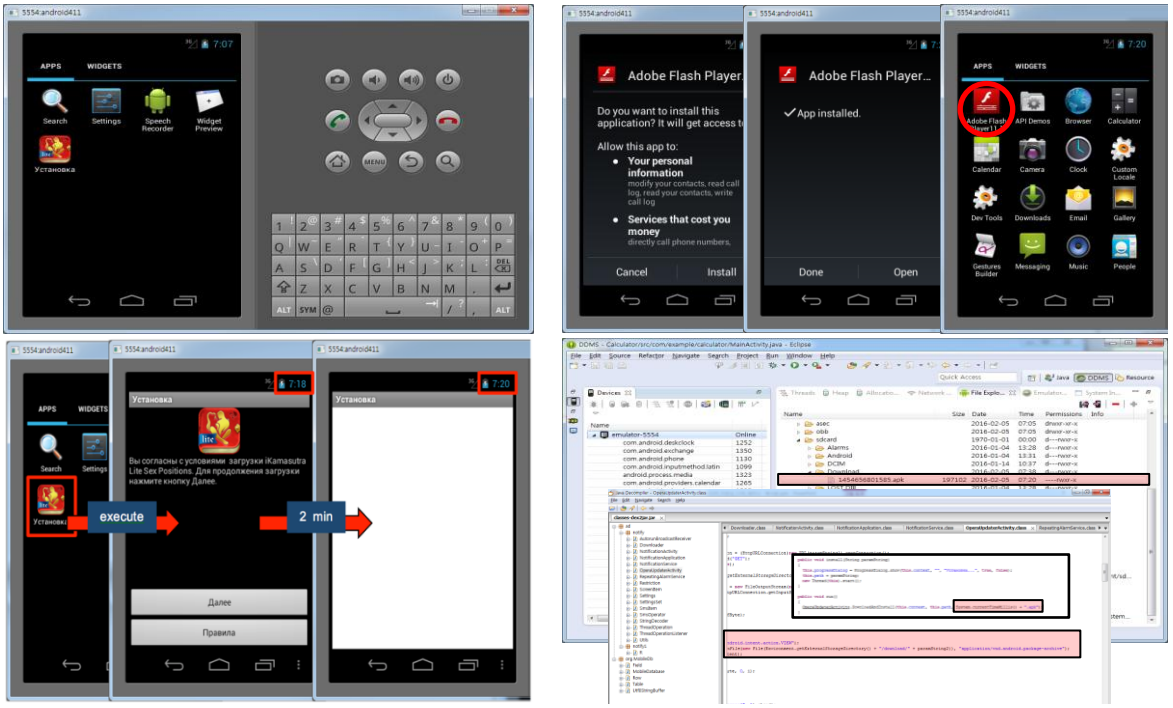

Figure 5. Activity of SSP Mobile Malware with Simulated Dynamic C&C Server

In case of using Fake C&C server, we can just find a few events on executing FlashPlayer(com.adobe.flashplayer.apk) malware, on which it didn't operate with original C&C server. However, we were able to collect dynamic events activated with simulated C&C server, with which we can proof and classify the pattern of server-side polymorphic mobile malware more in detail. Now you can see that the previous "FlashView.apk" file is downloaded to the user's phone as a "1454656801585.apk" file through the download server. In other words, it was confirmed that the file of the same size as that of FlashView.apk was changed to another file name "1454656801585.apk" at the time when the alarm occurred.

5.2 Evidence Aggregation on Dynamic SSP Malware with Simulated C&C Server

By carrying out network packet analysis on the suspicious mobile apps[17], it was possible to obtain the external transaction of SSP malware. Based on it, we can perfectly proof the external activity of hidden evasive server-side polymorphic mobile malware more in detail after capturing packet as follow Figure 6. In addition to analyzing malicious app information generated by simulated C&C server and network-based packet transmission / reception information, we could actually analyze the operating structure of SSP malicious apps. As shown in the figure below, the FlashPlayer.apk malicious app was able to confirm that it sent a command message to set the connection to the IP address 52.28.249.128, and it was also generated as a log file.

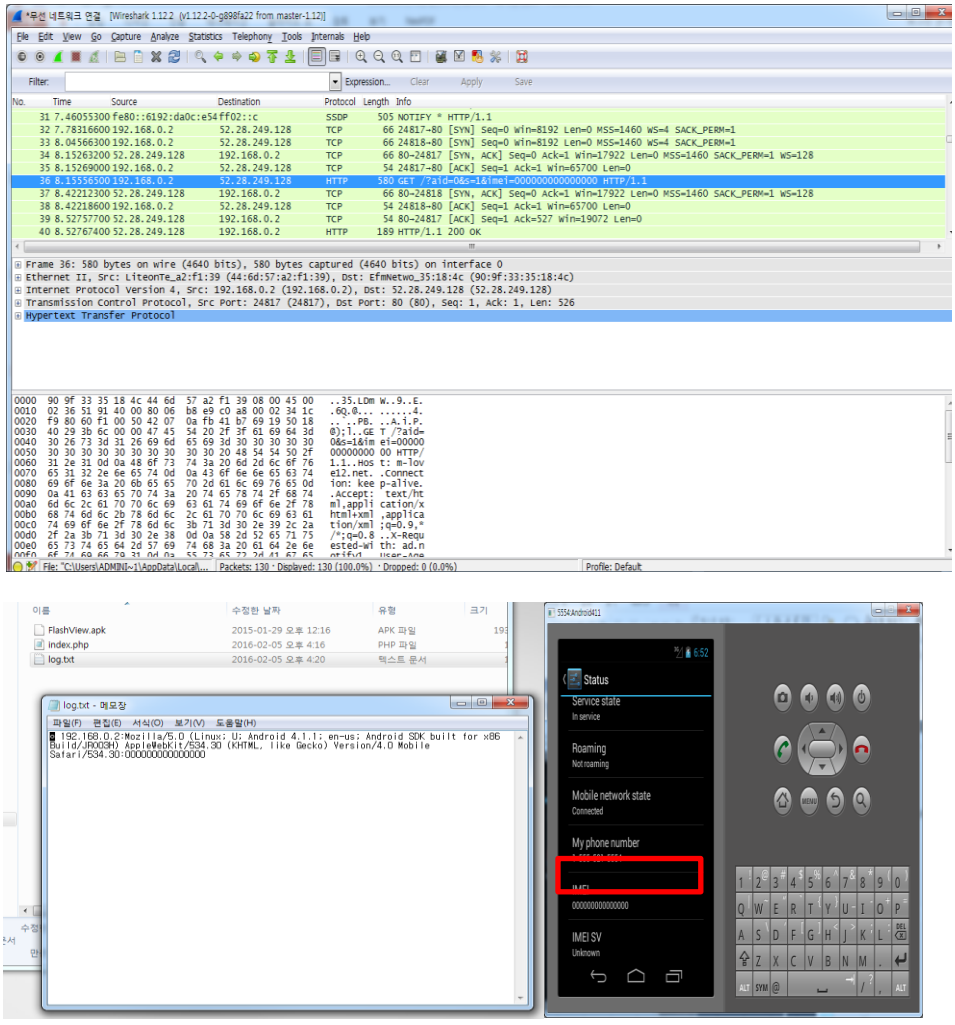

Figure 6. Evidence Aggregation of SSP Malware with Simulated C&C Server

6. Conclusions

The newly emerging type of server-side polymorphic malware has the ability to update and reshape its internal code by interaction with C&C server dynamically. As a result, it is very hard to detect suspicious transaction or real evidence of Server-side polymorphic malware code because its original C&C server was shut downed or its IP address was changed rapidly. Therefore, it was very difficult to detect or classify suspicious mobile apps because we can't aggregate real events as a proof of Server-Side Polymorphic code. Therefore, by building a pseudo dynamic C&C server, you can dynamically acquire complete evidence of the entire SSP malicious app's behavior. Therefore, by building a simulated C & C server, you can dynamically acquire complete evidence of the entire SSP malicious app's behavior. This evidence can be used to determine how SSP malicious apps work and to determine whether a particular app is malicious. By implementing simulated C&C Server, we can perfectly gather detailed evidences and detailed proofs to determine suspicious apps as a Server-side Polymorphic mobile code. Therefore, we implemented a pseudo dynamic form of simulated C&C server to aggregate suspicious Server-side polymorphic transactions perfectly. Using proposed simulated C&C server, we can aggregate evidence completely and classify newly emerging server-side polymorphic malicious code more concisely.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2014R1A1A2057430). And This article is a revised and expanded version of a paper [18] entitled "Simulated Dynamic C&C Server Based Activated Evidence Aggregation of Evasive Server-Side Polymorphic Mobile Malware on Android", presented at Advanced and Applied Convergence, 3rd International Joint Conference (IJCC2017) held on February 7-11, 2017, Hongkong.

References

[1] Shaun Nichols, Polymorphic malware on the rise, says Sophos, December 2012, http://www.v3.co.uk/v3-uk/news/2229214/polymorphic-malware-on-the-rise-says-sophos/.
[2] LAVASOFT, "Detecting Polymorphic Malware," http://www.lavasoft.com/mylavasoft/securitycenter/whitepapers/detecting-polymorphic-malware/.
[3] Ryan Sherstobitoff, "Server-Side Polymorphism: Crime-Ware as a Service Model (CaaS)," ISSA Journal, 2008.
[4] Vaibhav Rasgtogi, Yan Chen and Xuxian Jiang, "Catch Me if You Can: Evaluating Android Anti-malware against Transformation Attacks," IEEE Transactions on Information Forensics and Security, Vol.9, No.1 (2014), 99-108.
[5] Mohd Zaki Mas'ud, Shahrin Sahib, Mohd Faizal Abdollah, Siti Rahayu Selamat and Robiah Yusof, "Android Malware Detection System Classification," Research Journal of Information Technology, Vol.6 No.4 (2014) pp.325-341.
[6] Y. Zhou, and X. Jiang, "Dissecting android malware: Characterization and evolution," Proc. of the IEEE Symposium on Security and Privacy, San Francisco, California (2012), 95-109.
[7] Michael Spreitzenbarth, Felix Freiling, "Android Malware on the Rise," University of Erlangen, Dept. of Computer Science, Technical Reports, CS-2012-04, April 2012.
[8] Shaerpour, K., A. Dehghantanha and R. Mahmod, "Trends in android malware detection," Journal of Digital Forensics, Security and Law, Vol.8, No.3 (2013), 21-40.
[9] Xuxian Jiang, Yajin Zhou, Android Malware, Springer, NY, USA (2013).
[10] Han Seong Lee, Hyung-Woo Lee, "Implementation of Polymorphic Malware DB based Dynamic Analysis System for Android Mobile Applications," IJCC 2015, AACL 04 (2015), 170-173.
[11] Han Seong Lee, Hyung-Woo Lee, "Fake C&C Server based Server-Side Polymorphic Malicious Mobile Code Detection and Evidence Aggregation on Android Platform," Information, Vol.18, No.8, (2015) 3723-3737.
[12] Symantec Security Response, "Server-side Polymorphic Android Applications," https://www.symantec.com/connect/blogs/server-side-polymorphic-android-applications.
[13] "Droidbox," https://code.google.com/p/droidbox/.
[14] "Androguard," https://code.google.com/p/androguard/.
[15] Cool tools for admins: Check out our latest top ten lost in this free digital edition!, http://www.linux-magazine.com/Issues/2013/155/Code-Analysis/.
[16] Han Seong Lee, Hyung-Woo Lee, "Dynamic Analysis System for Detecting Remote Server-Side Polymorphic Malicious Mobile Apps on Android based Smartphone," International Journal of u- and e- Service, Science and Technology, Vol.8, No.11, (2015) 295-302.
[17] A. Shabtai, L. Tenenboim-Chekina, D. Mimran, L. Rokach, B. Shapira, Y. Elovici, "Mobile malware detection through analysis of deviations in application network behavior," Computers & Security, Vol.43, June 2014, 1-18.
[18] Han Seong Lee, Hyung-Woo Lee, "Simulated Dynamic C&C Server Based Activated Evidence Aggregation of Evasive Server-Side Polymorphic Mobile Malware on Android", Advanced and Applied Convergence, 3rd International Joint Conference (IJCC2017), pp.118-119, 2017.