



ICONI



KOREAN SOCIETY FOR INTERNET INFORMATION

The 11th International Conference on Internet

(ICONI 2019)

Dec. 15-18, 2019, Lotte Hotels & Resorts Hanoi, Vietnam

<http://www.iconi.org>

Proceedings of ICONI 2019

| Organized by |

Korean Society for Internet Information (KSII)

| Sponsored by |

Korea Electronics Technology Institute (KETI)

Contents



7-7	Determinants of Continuance Intention in Mobile Payment Services Viyada Itthiphone, Dong Hyuk Jo, Chul Hwan Kwon (Soongsil Univ., ROK)	207-215
7-8	Cross-media Sentiment Analysis Based on Dual Channels Xibin Jia, Xun An (Beijing Univ. of Technology, China)	216-220
8-1	Improved Static Analysis Application for Ransomware Detection on Android So-Hee Park, Jin Kwak (Ajou Univ., ROK)	221-225
8-2	Energy Consumption Minimization of Moving UAVs with Delay Constraints for UAV-based FSO Vertical Networks Wonsuk Yoo, Jong-Moon Chung (Yonsei Univ., ROK)	226-228
8-3	High Availability Multi-Agent Key Recovery System Using Signcryption Scheme TaeHoon KIM, Hak-Yong Kim, Im-Yeong Lee (Soonchunhyang Univ., ROK)	229-231
8-4	Invasion of privacy through faces exposed to the media; YouTube Semi Park, Junhyoung Oh, Hyunsik Yoon, Chaeun Lee, Chaewoon Kim, Kyungho Lee (Korea Univ., ROK)	232-234
8-5	Analysis Method Study on Android malicious app equipped with anti-analysis technology Seokhun Lee, Jae-Cheol Ryou (Chungnam National Univ., ROK)	235-237
8-6	A Study on the Search Technique of Similar Posts Based on Simhash JiHee Ha, Chanung Hwang, Taejin Lee (Hoseo Univ., ROK)	238-240
8-7	Optimal UAV trajectory planning for urban multiple Wi-Fi structure coverage Youngho Kwon, Woongjae Lee, Yoojin Jang, June Hwang (Seoul Women's Univ., ROK)	241-245
9-1	A New Approach of Picture Fuzzy Sets integrated with TOPSIS model for Group Decision Making in Evaluation under Uncertainty Hai Pham Van, Long Kim Cu (Hanoi Univ. of Science and Technology, Vietnam)	246-252

Improved Static Analysis Application for Ransomware Detection on Android

So-Hee Park¹ and Jin Kwak²

¹ISAA Lab, Department of Computer Engineering, Ajou University, Suwon, South Korea

²Department of Cyber Security, Ajou University, Suwon, South Korea

[e-mail: shpark.isaa@gmail.com¹, security@ajou.ac.kr²]

²Corresponding author: Jin Kwak

Abstract

Android takes up big share of the mobile operating system market, and a variety of applications are emerging. This makes it difficult to distinguish between ransomware applications and normal function applications. Moreover, as it is difficult to recover from damage caused by ransomware, ransomware must be detected and blocked before being installed. Dynamic and static analyses are used to detect ransomware, but for detection before ransomware installation, a static analysis-based system is required. Therefore, in this study, we investigate and extract the functions necessary for ransomware static analysis and detection systems. In addition, we analyze the problems that may occur in existing detection systems and propose an improved ransomware static analysis and detection system.

Keywords: Android, Ransomware, Static Analysis, Detection

1. Introduction

Ransomware is a malicious program that disables a system by locking or encrypting its data and then demands ransom, thus holding the system hostage [1]. Android accounts have the largest share of the mobile operating system market, and there are several different types of applications being released [2]. As the type of applications become more diversified and the number of applications increases, distinguishing between ransomware and normal applications, and recovering from the damage caused by ransomware become difficult. In this regard, ransomware should be detected in advance and then blocked [1]. A system should be able to detect ransomware through static analysis before the ransomware is executed.

In this paper, we analyze the android ransomware detection system presented by

previous studies and select the functions that the ransomware detection system should have. Subsequently, we propose a solution to the problem of RansomwareElite by applying the selected functions.

This paper is organized as follows. In the Related Works section in chapter 2, we analyze RansomwareElite, SafeDroid, RansomwareProber, SUPOR, and HelDroid. In Chapter 3, we analyze the problems of existing ransomware detection systems. Chapter 4 proposes improvement measures, and Chapter 5 presents the conclusions.

2. Related Works

2.1 RansomwareElite

RansomwareElite is a system that detects the presence of ransomware on Android devices using an approach based on static analysis. It

consists of four modules, namely, threatening text detector, lock detector, offline permission verification module, and ransomware database. The system determines if the application is ransomware when it is in APK state before installation. It performs actions such as detecting threatening texts in images, finding functions or classes that lock the system or perform encryption, and creating a database for sharing ransomware application information [3].

2.2 SafeDroid

SafeDroid is a system that detects the presence of ransomware on android devices and presents an approach based on static analysis. The system detects threatening keywords in images found in ransomware applications and marks an extension of the static text analysis approach. It provides the capabilities to extract text from images, regain control of the device, remove suspicious applications, and upload important files to an online server [4].

2.3 RansomwareProber

RansomwareProber is a system that detects the presence of ransomware on Android devices and presents an approach based on static and dynamic analyses. Through the system, the encryption operation of ransomware can be detected in real time. The system presents the characteristics analysis and real-time detection of Android ransomware with three main steps: encryption analysis, foreground analysis, and layout analysis. This system detects the file encryption operation, determines whether the encryption operation is abnormal or not, and automatically records the layout information [5].

2.4 SUPOR

SUPOR is a system that detects the presence of ransomware on Android devices and presents an approach based on static analysis. It consists of three modules: layout analysis, ui sensitiveness analysis, and variable binding. It identifies user input that contains important user data. This system records layout information, automatically inspects the UI, and extracts resource files and bytecodes [6].

2.5 HelDroid

HelDroid is a system that detects the presence of ransomware on Android devices and presents an approach based on static and dynamic analysis methods. It consists of three modules: threatening text detector, encryption detector, and locking detector. The system detects abnormal encryption operation without user consent. Among them, the UI is analyzed by recognizing a ransomware execution screen using the static analysis method. It detects threatening phrases in strings, checks for unnecessary file encryption operation, and also checks whether the application being analyzed can lock the device [7].

3. Problem Analysis

In this section, we analyze the modules of a ransomware detection system and classify the functional items that a static analysis ransomware detection system should have. Additionally, we analyze the problems that may occur when the ransomware detection system is not equipped with the functional classification items.

3.1 Ransomware detection system analysis

In this section, by comparing and analyzing ransomware detection systems, we derived four items as follows: UI analysis, permission analysis, application removal, and ransomware information collection. Encryption detection is a function that detects a class, function, or object name suspected of ransomware in a manifest file, before an APK is installed. UI analysis is a function of detecting a ransomware execution screen fixed at the top or movements unintended by user. Permission analysis is a function that detects an unnecessary high permission for the application requesting for the permission or a permission used for ransomware. Ransomware information collection allows ransomware information to be stored on a separate online server and used for detection. Table. 1 details the functions of the ransomware detection system.

Table. 1. Static function of ransomware detection system

Ransomware detection system	Encryption detection before APK installation	UI analysis	Permission analysis	Ransomware information collection
RansomwareElite	0	0	0	0
SafeDroid		0		
RansomwareProber		0		
SUPOR		0		
HelDroid		0		0

3.2 Need for RansomwareElite improvements

It is necessary to improve the configuration of the system and the overall operation for performance expansion in order to analyze the problems of existing ransomware detection systems. Therefore, in this paper, we present the problems and improvements of RansomwareElite, the system with the highest number of required conditions.

First, in the threatening text detector, threatening image detector, and offline permission verification modules, ransomware and normal encryption applications should not only be distinguished by string comparison analysis using suspicious lists. Additionally, we need an additional, dedicated function to extract and analyze static strings of the list.

Second, the suspicious list, which determines if

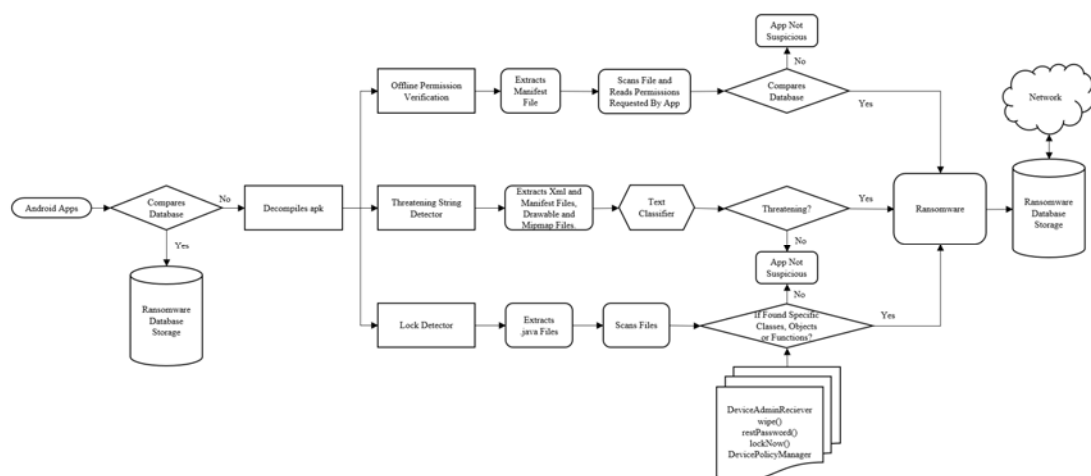
the requested permission is threatening, does not include some threatening permissions. After setting the exact criteria for classifying the list, it is necessary to further analyze the permissions in the list considering the latest trends.

Third, to prevent the spread of ransomware damage, a function of sharing or disclosing information about ransomware to other users is necessary.

4. Proposed Ransomware System

As for the encryption detection function to detect ransomware's behavior such as, device lock without the user's consent or abnormal encryption process such as data encryption, RansomwareElite is the only system with static analysis method, therefore, it has been excluded from the presentation of problems and improvements.

Improvements to the UI analysis item in [Table 1](#).

**Fig. 1.** Proposed Procedure of Ransomware Detection

are suggested in 4.1. Improvements to the permission analysis item are proposed in 4.2 and improvements to ransomware information collection item are presented in 4.3.

Fig 1. shows the structure of the improved ransomware detection system.

4.1 Threatening String Detector

In this section, we present a function to extract and analyze static strings against the suspicious list of an existing system. A good example of such a feature is HelDroid's threatening text detector. This detector is divided into three stages: text extraction, text classification and localization. It analyzes strings in code and resource files to extract static strings. Subsequently, the NLP classifier is used to detect threatening strings and sentences in the string. NAT's NLP classifier supports localization and it can detect unknown ransomware samples [7].

4.2 Ransomware Database

We propose a function to collect information about ransomware and share the information with other users. This can prevent the spread of ransomware damage. In the process of detecting ransomware, the database collects the application information determined to be ransomware, APK name, file structure, etc., and continuously synchronizes the database information with a separate online server. As a greater number of users proceed with the update, the damage spread can be effectively prevented.

4.3 Permission Verification Module

We propose a process of comparing and analyzing suspicious permissions for the permissions requested by the application. **Table 2.** shows a list of the permissions used by existing ransomware [9, 10, 11, 12]. This can help determine whether the application is ransomware by organizing and summarizing the permissions requested by the ransomware application with high frequency. This module periodically synchronizes the permissions requested by the latest ransomware and facilitates ransomware detection.

5. Conclusions

Damages caused by ransomware plague our lives on a daily basis [1]. In particular, the Android operating system is an environment vulnerable to threats as it provides open source services and also requests device permission when installing applications. Additionally, damages caused by ransomware are difficult to be recovered from. Therefore, it is necessary to take preventive measures against ransomware attacks by detecting them in advance and subsequently blocking them.

In this study, we analyzed the features and functions of the existing ransomware detection systems and proposed ways to improve these existing systems. Our findings are expected to help improve the efficiency and security of various applications.

Table 2. List of suspicious permissions

Type	Suspicious	Non-Suspicious
SYSTEM	RECEIVE_BOOT_COMPLETED, READ_PHONE_STATE, READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORE, WAKE_LOCK, GET_ACCOUNTS, BIND_DEVICE_ADMIN, DISABLE_KEYGUARD, INSTALL_SHORTCUT	GET_TASK, WRITE_SETTING, SYSTEM_ALERT_WINDOW, CAMERA
SMS	READ_SMS, SEND_SMS, RECEIVE_SMS, RECEIVE_WAP_PUSH, RECEIVE_MMS	
CONTACT	WRITE_CONTACTS, GET_ACCOUNTS, READ_CALL_LOG, WRITE_CALL_LOG	READ_CONTACTS, CALL_PHONE
NETWORK	INTERNET, ACCESS_NETWORK_STATE, READ_HISTORY_BOOKMARKS, ACCESS_WIFI_STATE	
LOCATION	ACCESS_COARSE_LOCATION, ACCESS_FINE_LOCATION	
STORAGE	READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORAGE	
SENSORS	BODY_SENSORS	

References

- [1] Korea Internet & Security Agency, "The definition and path of infection of ransomware", <https://www.krcert.or.kr/ransomware/information.do>
- [2] statcounter, "Mobile Operating System Market Share Worldwide", <https://gs.statcounter.com/os-market-share/mobile/worldwide>, 2019.
- [3] Shivanji, Gautam Sharma, Anubhav Johri, Akshita, Anurag Goel, Anuradha Gupta, "Enhancing RansomwareElite Application for Detection of Ransomware in Android Applications", 2018 Eleventh International Conference on Contemporary Computing (IC3), 2018.
- [4] Meet Kanwal, Sanjeev Thakur, "An application based on static analysis for android ransomware", 2017 International Conference on Computing, Communication and Automation (ICCCA), 2017
- [5] Jing Chen, Chiheng Wang, Ziming Zhao, Kai Chen, Ruiying Du, Gail-Joon Ahn, "Uncovering the Face of Android Ransomware: Characterization and Real-Time Detection", , IEEE Transactions on Information Forensics and Security, 2018
- [6] Jianjun Huang, Zhichun Li, Xusheng Xiao, Zhenyu Wu, Kangjie Lu, Xiangyu Zhang, Guofei Jiang, "SUPOR: Precise and Scalable Sensitive User Input Detection for Android Applications", 24th USENIX Security Symposium , 2015.
- [7] Nicolás Andronio, Stefano Zanero, Federico Maggi, "HelDroid: Dissecting and Detecting Mobile Ransomware", International Symposium on Recent Advances in Intrusion Detection, 2015.
- [8] Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Voddén, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, Patrick McDaniel, "FlowDroid: precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android applications", ACM SIGPLAN conference on Programming Language Design and Implementation (PLDI 2014), 2014.
- [9] Asma Khatoon, Peter Corcoran, "Android permission system and user privacy A review of concept and approaches", 2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin), 2017.
- [10] Karthick Sowndarajan, Sumitra Binu, "Android Security Issues and Solutions", 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2017.
- [11] Daniel E. Krutz, Nuthan Munaiah, Anthony Peruma, Mohamed Wiem Mkaouer, "An Analysis of Developer Permission Changes in Open Source Android Applications", 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft), 2017.
- [12] Ryan Stevens, Jonathan Ganz, Vladimir Filkov, Premkumar Devanbu, Hao Chen, "Asking for (and about) permissions used by Android applications", 2013 10th IEEE Working Conference on Mining Software Repositories (MSR 2013), 2013.