Indian Journal of Science and Technology, *Vol 9(46), DOI: 10.17485/ijst/2016/v9i46/107185, December 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Enhanced Intrusion Tolerant System for Mobile Payment

Daeyoo Kim and Eugene Rhee*

Department of Computer System Engineering, Sangmyung University, Korea;
kimxxg@naver.com, eugenerhee@smu.ac.kr

Abstract

Objectives: To prevent malicious software and hacking in near field communication-based mobile payments, this paper suggests a new intrusion tolerant system. **Methods/Statistical analysis:** As Cuckoo Sandbox restricts the access to the system resources of the applet transmitted through the network, this new intrusion tolerant system adopted Cuckoo Sandbox for the intrusion detection and the safe normal service of the operation system from various malicious attacks. To enhance the intrusion tolerant level, the scalable intrusion-tolerant architecture for distributed service is adopted to analyze the status of the system and the risk in the current environment through the intrusion detection, which optimizes the security level of a system. **Findings:** The intrusion tolerant system suggested in this paper may cover various malicious applications such as network attacks to the weak service, data driven attack to applications, privilege escalation or intruder login, main file access from intruders, and host-based attacks. This system prevents the access to the server cluster and blocks direct attacks by forming a barrier layer with a firewall and the proxy. It can detect various attacks through checking the file integrity by the operation of challenge/response protocol. This system also finds the infected server by checking the responses from each server. In addition, this system has much more active corresponding level than the existing scalable intrusion-tolerant architecture. In this intrusion detection system, both the data analysis and detection proceed simultaneously through the migration of the system, which maintains the operation of the system as it was. This virtualization technique is good for the implementation of the intrusion tolerance system and the migration of the mobile payment system in active state to another system. Especially, the live migration method is very effective to minimize the loss of time and data. **Application/Improvements:** Especially, this intrusion tolerant system makes the operating system safe from mobile payments intrusion and malicious software.

Keywords: Cuckoo Sandbox, Infection Analysis, Intrusion, Mobile Payment, Tolerant System

1. Introduction

As computers have become central for people and companies to conduct various businesses and to exchange very sensitive data using computers, computer security need to cover a wide area of computing and has become extremely important over the years. Consequently, there are adversaries whose ultimate motive is to seize these data for evil purposes. The number of malicious applications on the internet increases rapidly every year and locally occurred infections caused by malicious software has reached massive numbers[1–3]. Adversaries often disguise their malicious applications as benign and publish them on the internet in order to trick casual computer users into downloading and executing these applications locally, hence giving adversaries an opportunity to gain unauthorized access to system resources and seize or manipulate the sensitive data stored on users' computers.

*Author for correspondence

With recent wide use of smart phones, the main pattern of the mobile service has changed from past voice calls to various ubiquitous services, and many studies have been done for this trend[4–6]. In this trend, the near field communication based mobile payment services have emerged as a buzzword in the mobile business. The near field communication is excellent in security than traditional radio frequency identification as users should touch their smart phone directly to the communication partner for exchanging data. As the near field communication can exchange data as a non-contact near field communication, users can replace transportation cards, credit cards and various coupons with their near field communication built-in smart phone. The near field communication based mobile payment service is a service to exchange user group's data with partner mobile terminal (payment equipment) by touching and can be divided into application services and mobile payment services. The near field communication uses a built-in host processor with radio frequency and baseband. The protocol communication between the host and the processor is generally made in the firmware level stage, and the virtual interface human-computer interaction is for the compatibility of hosts. The near field communication front-end can communicate directly with a variety of device (mobile phone, personal computer), and the near field communication human-computer interaction includes reader/writer mode, P2P mode, and card emulation mode[7–9].

2. Obfuscation Technique

The obfuscation technique is one of the techniques that cover the contents of the original program. The contents can be covered by a program with encoding the original program to be run as a virtual machine. With changing the original program to the virtual machine byte code that indicates the same result, and if the executable file includes the program encoded with byte codes and a virtual machine which can execute this, it can perform the same result as the original program. However, using more than two virtual private clouds, virtual machines, or program storages to the obfuscation technique through a virtualization makes it difficult to drive the analysis as it can cause a variety of variants. Although the sophisticated obfuscation technique applied to the obfuscated program, the program should indicate the same result of the original program. In other words, even if the program is highly obfuscated, it should perform the same result as the original program[10–12].

Among various obfuscation techniques, this paper mainly adopts Proguard which is built as a common tool and used in Android. As the identifier transformation technique is applied to Proguard as open source, Proguard does not need additional code or member variables and supports the optimization for improved performances. In addition, this paper partially adopts commercially available software such as Stringer, Allatori, and Zelix Klassmaster. The techniques applied to this system are as shown in Table 1.

Table 1. Applied techniques per Android obfuscation

	Proguard	Stringer	Allatori	Zelix Klassmaster
Interchanging Virtual Path Identifier	O		O	O
Debugging Information Removal	O		O	O
Control Stream Change			O	O
String Encryption		O	O	O
Encoding	O			
Obfuscation	O			

3. Malicious Software

Malicious software, also referred to as malware or malicious program, is a term coined by malicious and software. Usually, malicious software's main target is Windows. However, it can infect all computing devices including smart phones and tablets. Recently, malicious software has become a big tendency in the infection of malicious software for mobile devices and spreads more and more widely to various areas[13–15]. The main symptoms may include system performance degradation, network traffic, file deletion, personal information leakage, automatic e-mail sending, and remote control. Malicious code can be classified as viruses, worm virus, and Trojan horse. The virus program is a combination of executable instructions that

copy themselves or any part of its variations on the magnetic data. The worm, which is also called as a network aware virus, is a combination of instructions that copy or modify itself without infection of other programs. The worm exists as a code form or executable file in a memory location, and, once run, it copies files or codes to another system. The Trojans is a combination of instruction and does not copy itself on purpose. With this reason, the Trojans is different from the virus or a worm that copies itself[9–11].

Recently, these malicious codes are expanding into mobile areas with worldwide increases in the prevalence of smart phones and mobile payments. The mobile malicious code infection process is as shown in Figure 1. These mobile malwares are usually repacked and have various types of attack; 1) personal information export including short message service and address book, 2) exploit to attack the vulnerability of user's mobile devices, 3) botnet to annoy control user's mobile devices, 4) Spyware to annoy users by launching push alarms.

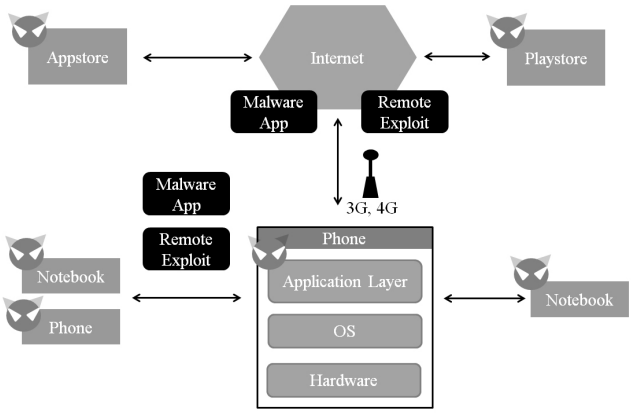

Figure 1. Mobile malicious code infection process.

The feature of the mobile malicious code is as follows. The mobile malware is installed in the terminal to perform an action, such as a personal information export, a smart phone remote control, and causing user billing. And these behaviors of the mobile malware provide a variety of benefits, including the money, to the hacker. For example, a hacker may try a phone call with an unspecified phone number using the smart phone infected by malware. In this case, a heavy load can be given to the base station and a call failure can occur. Zombie smart phones can transmit a large amount of voice call or short message to generate a load to a base station, which results in the failure to the base station and cause a denial of service. In addition, generating a large network traffic from the external Internet network can cause a bottleneck between the internet network and 3G network. Moreover, distributed denials of service can occur due to the excess of the maximum capacity of the available line, and mobile devices cannot use the external internet network through the 3G network. Zombie smart phones can perform not only distributed denial of service attacks but also mass distributions of malicious spam message.

As the method of distributing mobile malware is evolving fast, one of the well-known methods is the application repackaging. The distributed denial of service occurs as shown in Figure 2. The application repackaging makes copy applications or camouflage applications by modifying the existing source code or inserting some other application modules. And then, they are redistributed with the redistribution function of transmitting the input data from the existing application to attacker's server. Through data modulations such as malicious code insertion, replication of applications, and illegally collecting personal information, hackers will be able to implement the wanted function arbitrarily and freely. Especially, the attack of hackers is particularly active to the application that includes In-app billing function because hackers are naturally bound together where there is financial gains. Hackers can randomly modulate the prices of digital contents or items and have a direct financial impact on developers by intercepting their billing servers. With these reasons, games, banking, and e-commerce are mainly targeted by cyberattacks.

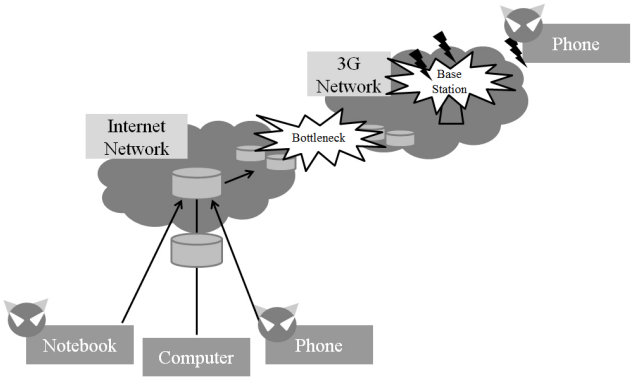

Figure 2. Distributed denial of service.

4. Cuckoo Sandbox

As the sandbox uses the resources only in the given area, it does not have much effects on the critical areas like as operating systems. Additionally, this sandbox can protect or cover the internal resources from outside access. Thanks to this characteristic of the sandbox, the environment analysis of the sandbox and the cloud service has been adopted in most vaccine programs from the past. With this reason, this paper adopted the Cuckoo Sandbox which is based on Python and can run all modules. Sandbox makes the program entered from outside operate in the protected area to prevent it from being operated falsely and restricts the access of applets transmitted over the network to the system resources of the sandbox[6]. Once, the applet is allowed to access to the sandbox, it can be operated normally. However, in other cases, the operation will prevent the damage of the system in a manner that can not be changed or read the local files. A specific area on the hard disk can be set with sandbox, and this accessible area makes resources available.

5. Intrusion Tolerant Systems

Currently, many researches related to the intrusion tolerant systems includes hardware-based intrusion tolerant systems and middleware-based intrusion tolerance system. First, a hardware-based intrusion tolerant system has a feature that tolerates a certain type of attack to maintain the required services of the system through the application of the hardware structure. In the beginning, a hierachical adaptive control of quality of service for intrusion tolerance architecture is studied. The hierachical adaptive control of quality of service for intrusion tolerance architecture has a simple duplex configuration of the primary server and the backup server as shown in Figure 3 and makes the backup server have the same condition as the primary server through data synchronizations between servers. When the primary server is under attack, it activates the backup server and maintains the service. In addition, by adding steps for accessing to the system, the hierachical adaptive control of quality of service for intrusion tolerance architecture made it harder to attack than existing other typical systems.

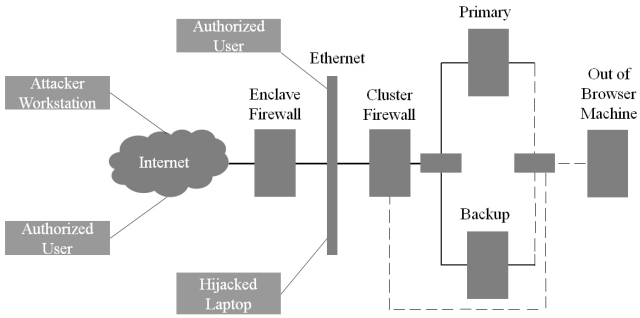

Figure 3. Hierachical adaptive control of quality of service for intrusion tolerance.

However, the hierarchical adaptive control of quality of service for intrusion tolerance architecture can tolerate the limited types of attacks due to its simple structure and has difficulty in responding when the frequency of the attack increases because it does not ensure a sufficient redundancy. To overcome these problems, this paper adopts the designing protection and adaptation into survivability architecture. The structure of the designing protection and adaptation into survivability architecture adopts the concept of demilitarized zones to protect critical parts of the system[7]. To prevent direct access from outside and to ensure the continuity of services, the designing protection and adaptation into survivability architecture has four replication servers. Each quad is consisting of executive zone, operation zone, and crumple zone as shown in Figure 4, and the role of each zone is also shown in Table 2.

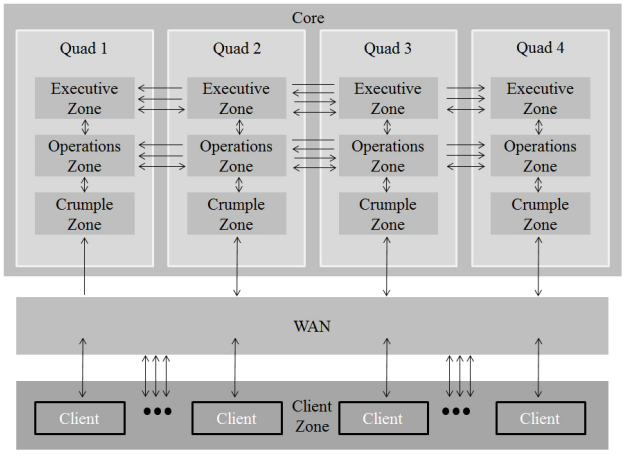

Figure 4. Designing protection and adaptation into a survivability architecture.

Table 2. Role of each zone

Zone	Role
Executive Zone	-System monitoring and management -Automatic control
Operation Zone	-Network intrusion detection -PSQ server running -Firewall running
Crumple Zone	-Management of traffic to core -Removal of packets that violate the security policy -Encrypting traffic to core

The hierachical adaptive control of quality of service for intrusion tolerance architecture can tolerate a wide range of attacks with a sufficient diversity and redundancy. Moreover, it prevents the system from falling out of the state service by protecting the central part of the system. However, the hierarchical adaptive control of quality of service for intrusion tolerance architecture cannot respond effectively to the attacks without a skilled administrator, the enormous cost is needed to implement because it uses a variety of hardware and software. In addition, it is vulnerable to attacks due to the long exposure time of the server. When the core server is damaged, it is very difficult to recover[8].

The middleware of middleware-based intrusion tolerant systems is a subsystem that exists between the external clients and the server cluster and performs a variety of roles such as detection, prevention of intrusion, and filtering. The features and characteristics of the structure and components in the scalable intrusion-tolerant architecture for distributed services which is the product of OASIS are as shown in Figure 5 and Table 3.

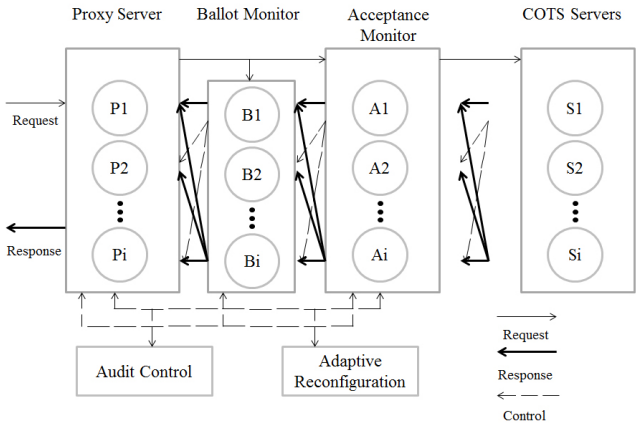

Figure 5. Scalable intrusion-tolerant architecture for distributed services.

Table 3. Role of each component

Component	Role
Proxy Server	-Forwarding requests and responses
Ballot Monitor	-Performing Byzantine consent procedures -Determine the final response
Acceptance Monitor	-Specific availability tests -Intrusion detection and trigger generation
Audit Control	-Monitoring and recording of the other element -Additional invasion diagnostic
Adaptive Reconfiguration	-Invasion information reception and security assessment -System reconfiguration for the environment

Using the middleware, the scalable intrusion-tolerant architecture for distributed services realized the intrusion tolerant system. To increase the intrusion tolerant level, it analyzes the status of the system and the risk in the current environment through the intrusion detection, which can optimize the security level of a system for its environment. However, unfortunately, this system has some disadvantages also. Due to its simple intrusion detection technique, it can detect a few types of intrusion and its response is slow. To overcome these problems, this paper proposes a specific structure that can be applied to the web server, and it is as shown in Figure 6.

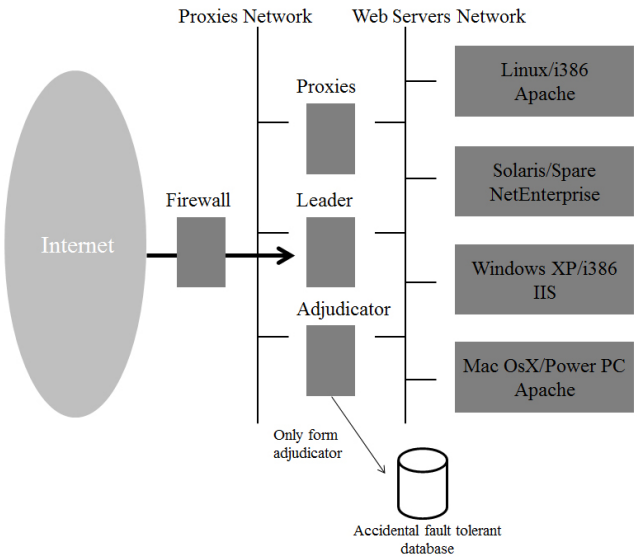

Figure 6. Intrusion tolerant system applied to web server.

With forming a barrier layer with a firewall and the proxy, this system makes it difficult to access to the server cluster, which means it can block direct attacks. And this system prevents the entire system things from falling to the out of order by using a different operating system, hardware, and software for each server. In addition, this system finds an infected server by comparing the responses from each server via the agreement protocol. It can do the fast detection for various attacks by checking the file integrity though the operation of challenge/response protocol. Moreover, this system shows much more active corresponding level to the threat than the existing scalable intrusion-tolerant architecture for distributed services.

As middleware-based intrusion tolerant systems have complicated structures and require a lot of communication, the performance of the system may drop significantly. And although they detect intrusions with various intrusion detection techniques, there still is a way to avoid them. If it does not detect an intrusion quickly, the entire system may suffer massive damages. And when it is attacked, as the only way to recover the system is reinstallation or restart, the recovery time may be long.

6. Virtualization System and the Implementation of the Security System for Mobile Payments

The main concept of the virtualization system is to abstract the computer system resources from your computer. In other words, the interaction among the application program, the other system, users, and the resource should be hidden or removed. This virtualization system generates single physical resources such as servers and operating systems which perform multiple logical resources function. Or a number of physical resources such as storage devices or servers can be generated also.

In order to prevent damages in case of mobile payments and to maintain the operation of the system, a variety of techniques are required. The suggested virtualization system in this paper is as shown in Figure 7. First of all, intrusion detection systems are needed to cover data attacks to the service, intruder login, and malicious software. In this paper, Cuckoo Sandbox is adopted for the intrusion detection system. As Cuckoo Sandbox is based on Python, Linux and Ubuntu programs, which provide Python and are designed to be easily used on a personal computer or servers, are adopted for the installation environment. And the VMware is used to run these programs.

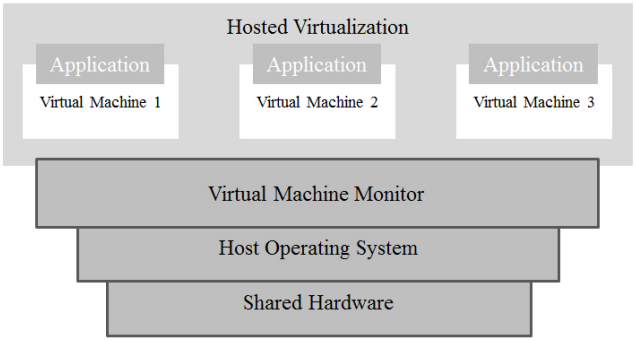

Figure 7. Virtualization System.

Using an intrusion detection system, the analysis and detection of data should be proceeded at the same time with the migration of the system. This is to maintain the operation of the system as it was, and this is the main concept of the intrusion tolerance system. To implement the intrusion tolerance system, a virtualization technique should be used to migrate the mobile payment system in active state to another system. To do this, the live migration method is recommended to minimize the loss of time and data. After the migration of the system is completed with a virtualization technique, the damaged system by malicious attacks should interrupt services.

7. Conclusions

As the mobile payment market has been increasingly growing, many systems for the security of mobile electronic payment are under developing. As one of the ways for the security aspects, this paper proposed a security system using the Cuckoo Sandbox model. By using the Cuckoo sandbox, this system may cover network attacks to the weak service, data driven attack to applications, privilege escalation or intruder login, main file access from intruders, and host-based attacks such as malicious software. As a future study, researches for the implementation of intrusion tolerant systems with virtualized systems in near field communication-based mobile payments environments are required.

8. Acknowledgement

This research was supported by Research Grant from Sangmyung University. We would like to address special thanks to Minjin Huh for her great encouragements and moral supports.

9. References

1. Sung YT, Tat EH. A mobile phone malicious software detection model with behavior checker. Proceedings of 3rd HSI, Japan; 2005. p. 57–65.
2. Jeong LK, Randy T, Cheol PG, Tae KY. A study on architecture of malicious code blocking scheme with white list in smartphone environment. Proceedings of FGCN, Korea; 2010. p.155–63.
3. Derrick SA, Frank P, Florian L, Sahin A. Monitoring smartphones for anomaly detection. Proceedings of the 1st International Conference on Mobile Wireless Middleware, Operating Systems, and Applications, Austria; 2008. p. 92–106.
4. Kapil S, Samrit S, Nehil J, Patrick T, Wenke L. Evaluating bluetooth as a medium for botnet command and control. Proceedings of 7th DIMVA, Germany; 2010. p. 61–80.
5. William S. Cryptography and network security. 6thedn. Pearson: England; 2014.
6. Wei Y, Heidemann J, Estrin D. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. IEEE/ACM Transactions on Networking. 2004 June; 12(3):493–506.
7. Verissimo P, Neves N, Cachin C, Poritz J, Powell D, Deswarte Y, Stroud R, Welch I. Intrusion-tolerant middleware: The road to automatic security. IEEE Security and Privacy. 2006 Jul; 4(4):54–62.
8. Miguel C, Barbara L. Practical byzantine fault tolerance and proactive recovery. ACM Transactions on Computer Systems. 2002 Nov; 20(4):398–461.
9. Paulo S, Neves BA, Miguel C, Ferreira NN, Paulo V. Highly available intrusion-tolerant services with proactive-reactive recovery. IEEE Transactions on Parallel and Distributed Systems. 2010 Apr; 21(4):452–65.
10. Taesoo K, Nickolai Z. Practical and effective sandboxing for non-root users. Proceedings of the 2013 USENIX Conference on Annual Technical Conference, San Jose; 2013. p. 139–44.
11. Robert W. Exploiting concurrency vulnerabilities in system call wrappers. Proceedings of the 1st USENIX Workshop on Offensive Technologies, Boston; 2007. p. 1–8.
12. Chris W, Crispin C, James M, Stephen S, Greg K. Linux security modules: General security support for the linux kernel. Proceedings of the 11th USENIX Security Symposium, San Francisco; 2002. p. 17–31.
13. Babazadeh SA, Salar R. A model for increasing usability of mobile banking apps on smart phones. Indian Journal of Science and Technology. 2015 Nov; 8(30):1–9.
14. Kumar DG, Rajasekaran S, Prabu R. PB verification and authentication for server using multi communication. Indian Journal of Science and Technology. 2016 Feb; 9(5):1–6.
15. Rehiman KR, Veni S. A secure authentication infrastructure for IoT enabled smart mobile devices. Indian Journal of Science and Technology. 2016 May; 9(9);1–6.