IJIBC 20-4-22

A Study on the Mobile Application Security Threats and Vulnerability Analysis Cases

Kim Hee Wan

Professor, Division of Computer Science & Engineering, Sahmyook University, Korea
hwkim@syu.ac.kr

Abstract

Security threats are increasing with interest due to the mass spread of smart devices, and vulnerabilities in developed applications are being exposed while mobile malicious codes are spreading. The government and companies provide various applications for the public, and for reliability and security of applications, security checks are required during application development. In this paper, among the security threats that can occur in the mobile service environment, we set up the vulnerability analysis items to respond to security threats when developing Android-based applications. Based on the set analysis items, vulnerability analysis was performed by examining three applications of public institutions and private companies currently operating as mobile applications. As a result of application security checks used by three public institutions and companies, authority management and open module stability management were well managed. However, it was confirmed that many security vulnerabilities were found in input value verification, outside transmit data management, and data management. It is believed that it will contribute to improving the safety of mobile applications through the case of vulnerability analysis for Android application security.

Keywords: *Security threats, vulnerabilities, mobile application, vulnerability analysis, reliability*

1. Introduction

The change to the smart era is providing various conveniences to individuals and companies, and it is also changing the way of life and work. Various information is provided through customized applications using a smartphone, and companies and governments develop and provide applications to provide convenience and information. There is a trend that companies adopt smart work steadily for reasons such as mobility without performance and space constraints, and cost benefits from improving office operation and productivity.

As a mobile service, security threats that exist in IT infrastructure and security threats to new mobiles exist at the same time. Mobile security-related accidents such as mobile malicious codes are already increasing rapidly, and in order to minimize security threats to the mobile environment, management services such as MDM (Mobile Device Management) and services such as mobile vaccines are receiving great attention. However, it is a reality that these solutions currently do not protect applications developed for mobile services from the threat of vulnerabilities of mobile applications [1][2].

Looking at the study on the leakage of smartphone information, a third party conducted a study in which

Manuscript Received: September. 18, 2020 / Revised: September. 22, 2020 / Accepted: November. 2, 2020
Corresponding Author: hwkim@syu.ac.kr
Tel: +82-2-3399-1783, Fax: +82-2-3399-1791
Professor, Division of Computer Science & Engineering, Sahmyook University, Korea

information registered on a smartphone was leaked through a man-in-the-middle attack (MITM) while installing and using a mobile app [3]. As a result of analyzing 140 free and popular apps on Android and iOS through the MobileAppScrutinator platform, leakage of personal identification information (Wifi, MAC address, AndoridID, IMEI) was confirmed [4]. In addition, a study on the vulnerability of mobile apps through static and dynamic analysis by extracting APK [5][6] and a study on the detection of personal information leakage of mobile applications [7] were published.

In this paper, in order to prevent security incidents that may occur in a mobile service environment, we specifically study Android application security based on Android application security review items and define the vulnerability analysis items that threaten security. Based on the defined analysis items, we want to analyze the vulnerability by checking the applications used by organizations or companies. It is expected that it will contribute to improving the safety of mobile applications through these examples of vulnerability analysis.

2. Mobile application security

2.1 Mobile service components

For safe mobile service, companies, institutions, and a various research divide the components of mobile office into content area, terminal area, network area, and server area as shown in Figure 1.

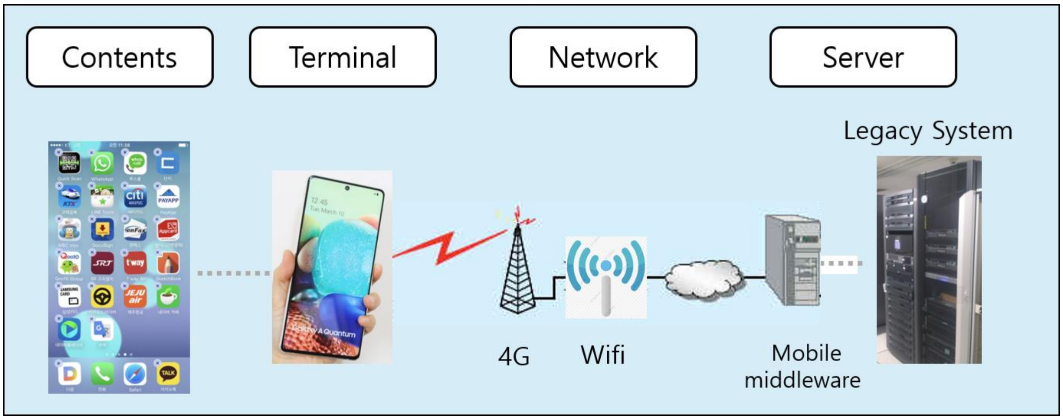

Figure 1. Mobile service component

The content area refers to all information transmitted to the terminal, such as installable mobile applications, emails, and advertisements that users most closely contact. The terminal area refers to a smartphone and a tablet PC possessed by users including a mobile OS. The network area refers to a network such as 4G and Wifi that connects the user's terminal and the server that provides the desired service. Finally, the server area refers to a server group that includes services to be provided to users, that is, a legacy system of a service provider. A secure mobile service environment will be constructed when the security for the mentioned components and the security for the section among the elements are provided [8].

2.2 Mobile application security threats

Mobile application security threats included in the content area mainly occur in the form of distributing malicious code that exploits operating system and application vulnerabilities through social engineering methods such as spam mail and SMS messages, thereby leaking personal information or gaining financial gain.

Table 1. Security threats in mobile application

Security threats	Explanation
Malware	Threats installed on the terminal for malicious behavior

Spam	Threats used to distribute advertisements and malware that can be sent to an unspecified number of people
Application vulnerability	Threats that perform malicious actions such as elevation of privileges by using the vulnerability of the developed application
Personal information extrusion	Threat of personal information leakage due to user carelessness when developing installed applications
Authentication bypass	Threats that randomly bypass or steal authentication for applications that require authentication
DoS	Threats that make the service provided by the application unusable

Security threats in these content areas can lead to vulnerability in most applications. Vulnerabilities in the application can bypass authentication and view unauthorized information.

It is possible to take malicious actions such as downloading and installing certain other applications with system level authority by forcibly rooting and jailbreaking the terminal by exploiting an already announced vulnerability. In addition, re-packaging, one of the techniques used by malicious codes, modifies normal applications to add malicious functions, or removes security functions and redistributes them so that many users install applications with malicious codes without suspicion.

2.3 Android application security research case

As interest in smartphones increases and power users who change the operating system increase, modifications that do not consider security, such as Jailbreak and Rooting, are easily delivered and used to general users. In the process, security technology is also gradually developing, centering on antivirus companies. In addition to mobile-specific antivirus, terminal management tools, mobile-specific IDS (Intrusion Detection System), and mobile security control services are emerging.

Lookout, an overseas mobile anti-virus company, has analyzed and converted more than 500,000 Android and iOS applications into a database on how applications access personal and sensitive information through the “App Genome Project,” which has been held since 2010. Through this, high-quality general statistics and statistics related to security are presented, and security threats can be predicted [9].

3. Application security method in mobile service

3.1 Response to mobile application security threats

Security threats to Android applications are increasing day by day, and the boundaries between normal and malicious applications are becoming blurred. A malicious user can inject malicious code into a normal application, and even a normal application plays the role of a malicious application due to a vulnerability for an instant.

Unintended use of functions by the developer, discovery of malicious code using application vulnerabilities, or redistribution that ignores copyright can result in loss of image, reliability, and profit loss to the organization. For example, in the case of a hacking case related to the leakage of personal information of SK Comms, the conclusion of the investigation that the hacking attempt was attempted using the server of Eastsoft as a stopover resulted in a decrease in the reliability and image of a company that is also in the security business.

It is more important to do defensive coding from the application development stage and distribute relatively safe applications through vulnerability checks after development.

3.2 Vulnerability check guide

The use of the vulnerability check guide can be roughly divided into two parts. First, it can be divided into a security review for the source code created through secure coding from a security review that must be done in the application implementation stage, and an action analysis that can be performed in the test stage after application development.

Table 2 categorizes the Android part of Veracode's top 10 mobile application threats. Functional parts that may be exploited are classified as Malicious Functionality, and parts that can be exploited as existing vulnerabilities are classified as Vulnerabilities. Also, in the reference of Malicious Functionality, the malicious code and concept that exploited the actual corresponding item were shown [10].

Table 2. Top 10 Veracode Mobile Application Threats [10]

Classification	Vulnerability name	Reference
Malicious Functionality	Activity monitoring and data retrieval	Secret SMS Replicator
	Unauthorized dialing, SMS, and payments	Premium rate SMS
	Unauthorized network connectivity	Exfiltration and C&C
	UI Impersonation	Android banking app Fraud
	System modification	Rootkit, APN proxy coning
	Logic or Time bomb	CWE-511
Vulnerabilities	Sensitive data leakage	CWE-200
	Unsafe sensitive data storage	CWE-312
	Unsafe sensitive data transmission	CWE-319
	Hardcoded password/keys	CWE-798

The Korea Internet & Security Agency has distributed a guide to verifying the security vulnerability of mobile apps as part of the mobile e-government service app security verification system in order to provide a service that performs source code security vulnerability checks for mobile applications developed by the state and public institutions. The guide describes verification procedures, among which the verification criteria are summarized in Table 3.

Table 3. Mobile App Security Vulnerability Verification Guide [11]

Security threats	Explanation
The presence of unspecified features	The contents of the function description such as the submitted security statement and UI name must match the actual app function.
Grant of least privilege	Grants only the minimum privileges required for function operation.
External input validation	When a function is operated based on external input information, the validity of the input information is verified to check whether the specified length is exceeded and the malicious code is included.
Safe management of sensitive information	Confirmation of encryption when storing and transmitting important information (personal information, personal location information, business information, etc.).
Confirmation of violation of mobile platform security model	Confirmation of the existence of platform modulation functions such as rooting and jailbreak.
Identification of source code security vulnerabilities	Confirmation of the existence of related vulnerabilities according to the classification of source code vulnerabilities such as input data verification and expression, API abuse, and security characteristics.
Safety assurance for commercial and public modules.	Verification of the suitability of the purpose and functions of commercial or public modules.
Verification of the use of common infrastructure security functions	Confirmation of the use of functions provided by the security infrastructure established according to the common infrastructure construction project for mobile e-government services
Confirmation of the existence of	Verification of the existence of known security vulnerabilities for the

known vulnerabilities.	platform and development language through the thesis website.

4. Mobile application security

4.1 Derivation of mobile application vulnerability assessment items

In security guides with different characteristics, we intend to derive security check items by deriving diagnostic items to check security after application implementation. We have summarized the diagnosis items of trusted organizations such as Ministry of Public Administration and Security and KISA for vulnerability diagnosis of Android-based applications, Google, which developed Android, and Fortify and Veracode, which are famous for source code diagnosis programs.

Mobile application vulnerability diagnosis items include permission management given to the application, verification of suitability for values entered into the application, encryption of DB and important files, management of data transmitted to the outside, and components used for convenience. A total of 9 items are used to check the possibility of exploitation of the function, vulnerability and programmatic safety.

Table 4. Security threats in mobile application

Checking No.	Diagnosis item	Explanation
1	Permission management	Management of data sharing through access rights and rights of other applications
2	Input value verification	Verification of suitability for values directly input through the user
3	Important file management	Ensure that important files are kept safe
4	Outside transmit data management	Encrypt important data when communicating with the outside world
5	Component management	Checking the possibility of abuse of used components
6	Security programming verification	Data exposure and safety check that may occur when coding programs
7	Data management	Notice of use of personal information, violation of mobile platform security model and configuration changes, and user authentication check
8	Open module safety management	Check the safety of the public modules
9	Database management	Confirmation of safety maintenance of database data

4.2 Case study of mobile application vulnerability diagnosis

In order to diagnose vulnerabilities against security threats in nine mobile applications derived from Table 4, security checks were conducted on applications used by three public institutions and companies. In order to inspect the nine areas in Table 4, we applied S application operated by public institutions that provide blood donation-related information including blood donation reservation management. And we also applied B application and social commerce of public institutions that provide user disease management and emergency medical information, and C application of a company that manages information.

4.2.1 S application

The S application version 1.5 is used by the H headquarters of a public institution, and has the function of managing blood donation reservations, searching, and providing blood donation-related information. The important information handled by the S application is account information. It is judged that MD5 was used as an encryption tool when developing an application, which can be said to be vulnerable due to the short number of bits of MD5. In addition, it was found that the S application does not manage sessions, so that other terminals can access with specific session information.

Table 5. Results of S application vulnerability diagnosis

Checking No.	Vulnerability Analysis Contents
2	In a form that receives user input, the input value is not restricted, which may lead to unintended results.
3	Insecure MD5 is saved as a hash value
4	Threat of sniffing always exists when using Wifi.
5	There is a threat that information is incorrectly transmitted by the user's choice when inserting a browser function by transmitting a URL that can be connected to a session through an implicit intent.
6	The key value used for application authentication is hard-coded and used.
7	There is always a threat of exposing stored DB information because it is not checked for platform alteration such as rooting.

4.2.2 B application

The B application version 1.5 is operated by Department B of public institutions, and has functions of finding medical institutions, managing user diseases, and providing emergency medical information, and important information is personal and medical/disease information. Disease information and personal information entered by the user are stored in an internal database. By using automatic login, addresses, phone numbers, and emails belonging to personal information are unnecessarily stored inside the terminal and reused for other functions, and such information is stored in plain text.

Table 6. Results of B application vulnerability diagnosis

Checking No.	Vulnerability Analysis Contents
2	In a form that receives user input, the input value is not restricted, which may lead to unintended results.
3	Not using secure encryption algorithms.
7	There is always a threat of exposing stored DB information because it is not checked for platform alteration such as rooting.
9	During automatic login, excessive information other than the account is stored on the terminal and used for other functions.
9	Without encrypting and storing important information, addresses, emails, and telephone numbers other than passwords are stored in plain text.

4.2.3 C application

The C application version 1.93 is operated by a private company L, which provides and manages social commerce information, and important information is account and product purchase information. When logging in, the account information is transmitted to the server for authentication. At this time, communication encryption is not used at the time of transmission, and only the password is hashed to MD5 and transmitted. If not combined with other key values, passwords may be exposed.

Table 7. Results of C application vulnerability diagnosis

Checking No.	Vulnerability Analysis Contents
2	Unintended results may occur because input values are not restricted in the user input form.
4	No encryption algorithm is used, only password is hashed using MD5 and transmitted.
4	There is always the threat of sniffing when using Wifi.
7	There is always a threat of exposing stored DB information because it is not checked for platform alteration such as rooting.

4.3 Analysis of three mobile application vulnerability diagnosis

Table 8. Analysis of three mobile application for security threats in mobile application

Checking No.	Diagnosis item	S application	B application	C application	Total
1	Permission management	0	0	0	0
2	Input value verification	1	1	1	3
3	Important file management	1	1	0	2
4	Outside transmit data management	1	0	2	3
5	Component management	1	0	0	1
6	Security programming verification	1	0	0	1
7	Data management	1	1	1	3
8	Open module safety management	0	0	0	0
9	Database management	0	2	0	2

As a result of application security checks used by three public institutions and companies, vulnerabilities of checking no. 1 and 8 were not found. This means that authority management and open module stability management are well managed in all institutions. However, 3 security vulnerabilities were found in the checking number 2, 4, and 7 items: Input value verification, Outside transmit data management, and Data management. In other words, since we did not limit the input value in the form that receives user input, unintended results could occur, and there was always the threat of sniffing using Wifi. In addition, it was confirmed that there was always a threat of exposing stored DB information because it did not check whether platform tampering such as rooting was performed. We found two security vulnerabilities in checking number 3 and 9, Important file management and Database management. During automatic login, excessive information other than the account was stored in the terminal and used for other functions, or important information was not encrypted.

5. Conclusion

With the mass spread of smart devices, security threats are increasing along with interest, and vulnerabilities in developed applications are also being announced in the midst of mobile malware. The government and local governments provide various applications for the public, and security checks are required when developing applications for reliability and security of applications. In this paper, we defined vulnerability analysis items to respond to security threats when developing Android-based applications among security threats that can occur in mobile service environments. Based on the set analysis items, the vulnerability analysis was performed by checking the applications of public institutions and private companies currently operating as mobile applications. It will contribute to improving the safety of mobile applications through the case of vulnerability analysis for Android application security.

As a future task, it is meaningful that items that can confirm application security vulnerabilities have been presented in detail in the development business for Android applications, which are the mains in mobile services, but there are areas that need to be improved in the future. There is a need for continuous research on functions and vulnerabilities caused by updating the operating system.

Based on this paper, we hope that research and development on ways to understand the vulnerabilities of mobile applications and minimize security threats will continue.

References

[1] Korea Information Society Agency, Yearbook information society statistics. 2019, Seoul : Ministry of Science and Technology Information and Communication

[2] J. H. Park, Mobile computing permeates mobile offices, expanding application fields, IT Daily, 2019.4.30, http://www.itdaily.kr/news/articleView.html?idxno=94456

[3] Timothy A. Chadza, Francisco J. Aparicio-Navarro, Konstantinos G. Kyriakopoulos, Jonathon A. Chambers, "A Look Into the Information Your Smartphone Leaks," 2017 International Symposium on Networks, Computers and Communications(ISNCC), pp. 1-6, May. 2017. DOI: https://doi.org/10.1109/isncc.2017.8072022

[4] Jagdish Prasad Achara, Vincent Roca, Claude Castelluccia, and Aurélien Francillon, "Mobileappscrutinator: A simple yet efficient dynamic analysis approach for detecting privacy leaks across mobile OSs," The 32nd Annual Computer Security Applications Conference (ACSAC), 2016.

[5] S. W. Ko, and S. G. Joung, "Implementation example of mobile application analysis and verification solution", Korea Institute of Information Security and Cryptology, Vol. 23, No. 2, pp. 21-28, April. 2013.

[6] S. H. Park, H. J. Kim, and T. K. Kwon, "OnSecurity of Android Smartphone Apps Employing Cryptography", Journal of The Korea Institute of Information Security & Cryptology, Vol. 23, No. 6, Dec. 2013. DOI: https://doi.org/10.13089/jkiisc.2013.23.6.1049

[7] S. J. Kim, and J. B. Hur, "Mobile Application Privacy Leak Detection and Security Enhancement Research," Journal of The Korea Institute of Information Security & Cryptology, VOL.29, NO.1, Feb. 2019.
DOI: https://doi.org/10.13089/JKIISC.2019.29.1.195

[8] J. Y. Shin, D.S. Kim, K. J. Han, and H. W. Kim, "A Study on the Security Checklist Improvements to improve the Security in the Mobile Applications Development," Journal of Digital Convergence, Vol. 12, No. 8, pp. 113-127, 2014. DOI: https://doi.org/10.14400/jdc.2014.12.8.113

[9] Lookout, "Introducing the App Genome Project", Jul. 2010

[10] Veracode, "Mobile App Top 10 List", Dec 2010, *https://www.veracode.com/blog/2010/12/mobile-app-top-10-list*

[11] Korea Internet & Security Agency, Mobile public service security vulnerability check guide, Ministry of Public Administration and Security, 2016.