http://dx.doi.org/10.14400/JDC.2014.12.8.113

모바일 어플리케이션 개발에서의 보안성 향상을 위한 보안 점검항목 개선에 관한 연구

신준엽[*] 김동수[**] 한기준[***] 김희완[****]

경기지방경찰청 수사과 사이버팀[*], 건국대학교 정보통신대학원[**], 건국대학교 컴퓨터공학부[***], 삼육대학교 컴퓨터학부[****]

A Study on the Security Checklist Improvements to improve the Security in the Mobile Applications Development

Jun-Yuop Shin[*], Dong-Soo Kim[**], Ki-Jun Han[***], Hee-Wan Kim[****]

Cyber Team of Criminal Investigation Section, Kyonggi Regional Police Agency[*]
Graduate School of Information and Telecommunications, Konkuk University[**]
Dept. of Computer Engineering, Konkuk University[***]
Division of Computer Engineering, Shamyook University[****]

요 약 모바일 기기의 사용은 개인 및 기업에게 다양한 편의를 제공해 주고 있다. 반면에 모바일 서비스를 위한 환경 구축으로 IT인프라에 존재하는 보안 위협과 새로운 모바일에 대한 보안 위협이 동시에 존재하고 있다. 모바일 환경에 대한 보안 위협을 최소화하기 위해 MDM(Mobile Device Management) 등의 관리 서비스와 모바일 백신 등의 서비스가 큰 관심을 받고 있다. 이러한 솔루션은 모바일 어플리케이션 자체 취약성의 위협으로부터 모바일 서비스를 위해 개발된 어플리케이션을 보호해 주지 못하는 것이 현실이다.
이로 인해 본 논문에서는 모바일 서비스 환경에서 발생할 수 있는 보안 사고를 예방하기 위해 어플리케이션 보안성 검토 항목을 기반으로 모바일 어플리케이션 보안 점검항목을 제시하였다. 이를 통하여 모바일 어플리케이션 개발에 대한 보안성을 향상시키고자 한다. 제시한 점검항목의 실효성 검증을 위하여 실제 안드로이드 기반 어플리케이션을 수집 및 분석하고 어플리케이션에 대한 전수검사를 실시하였고, 점검항목에 대해 전문가의 설문 조사를 통해 적합성을 검증하였다.

주제어 : 모바일 서비스, 보안위협, 어플리케이션 보안, 점검항목

Abstract The use of mobile devices offers a variety of services to the individuals and companies. On the other hand, security threats and new mobile security threats that exist in IT infrastructure to build the environment for mobile services are present at the same time. Services such as mobile and vaccine management services, such as MDM (Mobile Device Management) has attracted a great deal of interest in order to minimize the threat of security in mobile environment. These solutions can not protect an application that was developed for the mobile service from the threat of vulnerability of mobile application itself. Under these circumstances, in this paper, we proposed mobile application security checklists based on application security review items in order to prevent security accidents that can occur in a mobile service environment. We collected and analyzed Android applications, we performed a total inspection of the applications for verification of the effectiveness of the check items. And we checked that the check items through a survey of experts suitability was verified.

Key Words : Mobile Services, Security Threats, Application Security, Check Lists

Received 3 June 2014, Revised 17 July 2014
Accepted 20 August 2014
Corresponding Author: Hee Wan Kim(Shamyook University)
Email: hwkim@syu.ac.kr

ISSN: 1738-1916

1. 서론

스마트 시대로의 변화는 개인 및 기업에게 다양한 편의를 제공해 주고 있으며 생활과 업무의 방식까지 변화시키고 있다. 스마트폰을 이용하여 여러 정보를 맞춤형 어플리케이션을 통해 제공받고 기업 및 관공서는 편의와 정보를 제공하기 위해 어플리케이션을 개발하여 제공하고 있다. 또한 PC 못지않은 성능과 공간적 제약이 없는 이동성, 사무실의 운영과 생산성 향상에서 오는 비용적 이익 등의 이유로 기업들의 스마트워크의 도입이 꾸준히 증가하는 추세이다. 정보화진흥원의 '10년 통계에 의하면, 원격 스마트워크를 도입한 사업체 중 모바일을 이용한 근무 형태인 모바일 오피스를 도입한 업체가 70%에 이른다. 최근 '12년 2월 서울시가 모바일 오피스를 도입한 것과 같이 국가 기관 및 기업들은 직원들에게 스마트폰 및 테블릿 PC를 제공하고 업무에 활용하게 끔 하며 시대의 흐름에 발맞춰 모바일 오피스를 구축하고 있는 것이다. KT경제연구소에 의하면 '14년 5조 9000억 규모로 '09년에 비해 2배 이상 빠르게 성장할 것으로 보고 있으며, 삼성경제연구소에서는 모바일 오피스 구축 기업 및 공공기관은 2010년 15%에서 2012년 72%로 확대될 것으로 전망하고 있다. 최근 보안강화를 목적으로 금융권과 공공기관이 모바일 오피스에 가상화 기술을 적용하고 있다. 업무용 PC뿐 아니라 태블릿 PC, 스마트폰 등 모바일 기기로 안전하게 데이터베이스(DB) 정보 접근에 환경을 마련해 보안과 업무 효율성 향상 등 두마리 토끼를 한 번에 잡는다는 전략이다. 그러나, 모바일 오피스도 관리가 필요한 어플리케이션을 기반으로 이루어지며, 이에 대한 보안 또한 필요한 시점이다[1][2][3].

모바일 서비스를 위한 환경의 구축은 기존의 PC기반의 정보 제공 방식을 모바일 기반에서 사용하기 위한 환경을 구축함을 뜻하며 이는 기존의 PC환경, 즉 IT인프라에 존재하는 보안 위협과 새로운 모바일에 대한 보안 위협이 동시에 존재할 수 있음을 보여준다. 이는 '10년 삼성경제연구소의 조사에서 기업 CEO의 약 50%가 정보보안을 모바일 오피스 도입에서의 가장 큰 걸림돌로 생각하는 것으로 나타난 조사에서 잘 드러난다. 이미 모바일 악성코드 등 모바일 보안 관련 사고들이 가파르게 증가하며 발생하고 있으며 모바일 환경에 대한 보안 위협을 최소화 하기 위해 MDM(Mobile Device Management) 등의 관리 서비스와 모바일 백신 등의 서비스가 큰 관심을 받고 있다. 하지만 현재 이러한 솔루션은 모바일 어플리케이션 자체 취약성의 위협으로부터 모바일 서비스를 위해 개발된 어플리케이션을 보호해 주지 못하는 것이 현실이며 이를 위한 보안 점검항목은 많지 않은 실정이다[1][2].

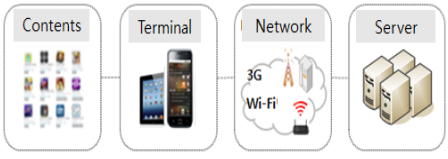
본 논문은 향후 모바일 서비스 환경에서 발생할 수 있는 보안 사고를 예방하기 위해 어플리케이션 보안성 검토 항목을 기반한 모바일 어플리케이션 보안 점검항목을 제시하여 모바일 오피스 구축 및 대국민 대상 모바일 어플리케이션, 내부 업무를 위한 모바일 어플리케이션 등 모바일 서비스에서 필요한 모바일 어플리케이션에 대한 보안성을 향상시키고자 한다.

2. 관련 연구

2.1 모바일 서비스 환경 구성 및 보안 위협

2.1.1 모바일 서비스 구성요소

안전한 모바일 서비스를 위해 기업 및 기관, 각종 연구에서 모바일 서비스의 구성요소를 황해수는 정보보안영역, 장치보안영역, Player 보안영역, 서버보안영역, 네트워크 보안영역의 5개영역[4], 방송통신위원회에서는 응용프로그램 및 플랫폼 부문, 단말 부문, 네트워크 및 서버 부문의 3개영역[5], ㈜SK Telecom에서는 장치, 서버, 네트워크의 3개영역[6], ㈜AIRCUVE에서는 어플리케이션, 스마트 모바일 단말, 서버, 네트워크 인프라의 4개영역[7]으로 적게는 3개, 많게는 5개의 영역으로 분류하고 있다. 각각 다른 용어를 사용하고 있지만 모바일 단말 영역과 네트워크 영역, 서비스 및 서버의 3개 영역은 공통적으로 언급된다. 본 논문에서는 공통된 3개의 영역, 즉 단말영역, 네트워크 영역, 서버 영역에 데이터 보안의 중요성을 감안한 컨텐츠 영역을 추가한 4개영역으로 구분하였다.



Source : [7] Smart Mobile Solution Configuration V4

[Fig. 1] Configuration of Mobile Services[7]

각 모바일 서비스 구성 영역이 포함하는 범위는 다음과 같다. 컨텐츠 영역은 사용자들이 가장 밀접하게 접하는 설치 가능한 모바일 어플리케이션이나 이메일, 광고 등 단말로 전송되는 모든 정보들을 말한다. 단말 영역은 모바일 운영체제를 포함하여 사용자들이 소지하는 스마트폰 및 테블릿 PC를 말한다. 네트워크 영역은 사용자의 단말과 사용하고자 하는 서비스를 제공하는 서버를 연결해 주는 3G와 WiFi 같은 네트워크망을 말한다. 마지막으로 서버 영역은 사용자들에게 제공할 서비스를 포함하는 서버군 즉 서비스 제공자의 레거시 시스템 및 모바일 관련 미들웨어를 말한다[5][6][7]. 언급된 구성요소들에 대한 보안과 요소간 구간에 대한 보안이 제공되었을 때 안전한 모바일 서비스 환경이 구성될 것이다.

2.1.2 모바일 서비스 영역별 보안 위협

<Table 1>은 모바일 영역을 구성하는 요소와 그에 대한 보안 위협을 나타낸다.

〈Table 1〉 Mobile domain-specific security threats[4]

Domain	Components	Security Threats
Contents	Mobile application contents Mobile advertisement Application market	Malware and Spam Application vulnerabilities Authentication bypass and leakage of personal information DoS
Terminal	Mobile Terminal	Lost and stolen Firmware / mobile OS vulnerabilities Authentication bypass and leakage of personal information DoS
Network	WiFi network VPN 3G network	Abnormal traffic Rouge AP Installation (Including tethering) BS Core network attack MITM attack
Server	Service System (Web, Mail, BBS) Mobile middleware	Server OS vulnerabilities Server application vulnerabilities Content distribution of malware infections

Source : [4] H. S. Hwang's Security Threats Summary

모바일 단말의 보안 위협은 크게 두 가지로 나뉠 수 있다. 분실 및 도난과 모바일 OS 취약점에서 오는 보안 위협이다. 소형인 모바일 단말의 크기로 인해 분실 및 도난이 쉬우며, 오는 5월부터 시행될 예정인 블랙리스트 제도에 의해 분실 및 도난 된 단말도 신고 이전까지 다른 USIM 칩으로도 사용 가능하게 되어 개인정보 유출의 위험이 더욱 커지고 있다. 아래 <Table 2> 는 취약점 데이터 제공 기관인 CVE 기반의 안드로이드와 iOS의 최근 주요 취약점 현황이다[8].

〈Table 2〉 Status of major vulnerability for Android and iOS operating systems [9]

	Vulnerability	Contents	CVE
Android	Buffer overflow	Rooting Libsysutils possible vulnerability	2011-3874
	Cross Application Scripting	Any domain of any executable Java script code	2011-2357
	Integer value range check	Rooting void volume manager and code execution vulnerability	2011-1823
	System space access bypass	An elevation of privilege by bypass the sandbox and Rooting	2011-1149
iOS	Improper system calls handling	Sandbox kernel vulnerability to execute arbitrary code	2012-0643
	Integer underflow	The catalog file HFS, DoS and arbitrary code execution	2012-0642
	Overflow	DoS and arbitrary code execution vulnerability in WebKit	2012-0635
	Cross site scripting	Web script injection vulnerability in WebKit	2012-0590

Source : [9] MITRE's Major Vulnerability Summary

위 공격들은 모두 기존에 존재해왔던 취약점을 이용한 공격들로써 PC 기반의 운영체제에도 발생할 수 있는 취약점들이다. 이는 모바일 환경에도 PC와 비슷한 수준의 위협이 존재하며 그에 상응하는 보안이 필요하다는 것을 의미한다.

모바일 네트워크에서의 위협은 PC환경에서의 위협과 WCDMA, WiBro, WiFi 등 모바일 통신 환경 고유의 위협을 포함하고 있어 개인 정보 유출의 위험이 더욱 확대되었다. 네트워크 구간에서의 Rouge Ap의 설치 및 WiFi 망에서의 패킷 스니핑을 통한 MITM 공격은 뉴스에도

종종 소개될 정도로 가장 쉽게 접할 수 있는 네트워크 공격 중 하나이다. 일반적인 커피전문점이나 공원과 같은 공공장소에서는 AP에 대해 암호를 사용하고 WPA, WEP 등 암호화 기법을 사용하지만 공개된 AP에 대한 암호는 널리 알려져 있으며, WEP나 WPA에 대한 복호화도 가능한 실정이다. 아직 모바일 서비스 시장이 초기 단계이며, 모바일 단말의 성능상의 이슈로 통신 암호화를 하지 않아 개인 계정 정보 등 MITM 공격을 통해 쉽게 불특정 다수의 정보를 수집할 수 있다. 또한 최근 3G 네트워크 망에 대한 해킹 위험을 알리는 실험들이 많이 등장하고 있다. 2010년에 GSM 및 3G KASUMI 네트워크에 대한 프로토콜 암호화가 복호화 될 수 있음이 실험을 통해 알려졌으며, 블랙햇 컨퍼런스에서는 오픈소스 소프트웨어인 'OpenBTS'와 저비용 장비를 이용해 가짜 기지국을 만들고 근방의 아이폰 사용자를 찾아내고 메시지를 보낼 수 있는 이동통신 네트워크상의 결함을 발표한바 있다[10].

컨텐츠 영역에서의 보안 위협은 주로 스팸메일 및 SMS 메시지 등의 사회공학적 방법을 통해 운영체제 및 어플리케이션 취약점을 악용하는 악성코드를 유포하여 개인정보를 유출하거나 금전적 이득을 취하는 형태로 발생한다. 어플리케이션의 취약점으로 인증을 우회하여 허가되지 않은 정보를 열람할 수 있으며, 실제로 아이폰, 안드로이드의 어플리케이션에서 Directory Traversal 등의 취약점이 발견되었다. 이미 발표된 취약점을 악용하여 단말을 강제로 Rooting 및 Jailbreak 하여 시스템 레벨의 권한으로 특정 다른 어플리케이션을 몰래 다운로드하고 설치하는 등의 악의적인 행위를 취할 수 있다. 개인정보 유출 또한 어플리케이션에서의 데이터베이스 관리 소홀 및 잘못된 설계로 인해 발생할 수 있다. 보안 전문가가 아닌 악의적인 해커에 의해 해당 취약점이 발견되어 충분히 악용될 수 있는 부분이다[11].

모바일 서비스의 보안 위협은 모바일 악성코드의 성장으로 더욱 커지고 있으며, PC 악성코드의 형태나 기능을 답습하는 양상을 보이고 있다. 이는 모바일 어플리케이션에 대한 직접적인 위협이며 어플리케이션 개발 단계부터 보안 위협으로부터 안전하게 개발 되어야 할 필요가 있다.

2.2 정보시스템 보안 감리

정보시스템 감리는 "정보시스템의 효율적 도입 및 운영 등에 관한 법률" 제2조 제3호에 따르면, 감리발주자 및 피감리인의 이해관계로부터 독립된 자가 정보시스템의 효율성을 향상시키고 안전성을 확보하기 위하여 제3자적 관점에서 정보시스템의 구축에 관한 사항을 종합적으로 점검하고 문제점을 개선하도록 하는 것을 말한다[12].

정보시스템 보안감리는 정보시스템의 구축과 운영 과정에서 정보시스템의 기밀성(Confidence), 무결성(Integrity), 가용성(Available) 을 보장하기 위하여 정보보안의 문제점을 식별하고 개선사항을 도출하여 시정토록 하는 것이다. 정보시스템 보안감리는 대상에 따라 사업보안감리와 운영보안감리로 나눌 수 있다. 정보시스템 사업보안감리는 정보시스템 기획단계에서부터 진행 단계별로 정보보안에 관련된 문제점을 분석, 평가하고 개선시켜 정보시스템 도입 이후의 발생할 보안문제를 최소화하고 보안사고로 발생할 수 있는 비용과 손실을 최소화하기 위해 실시하는 정보시스템 감리활동이다. 정보시스템보안감리지침에서는 개발공정을 분석, 설계, 구현, 시험, 전개의 다섯 가지 공정으로 구분하고 이에 대한 영역별 감리 평가항목을 제시한다. 이는 사업의 전 단계를 포함하고 있으므로 신규 정보화 사업에 대한 정보보안감리 역시 이 공정단계에 맞추어 적용할 수 있다[13].

본 논문은 모바일 서비스의 어플리케이션 개발단계에서 발생하는 보안 요소에 대한 연구로써 그 적용 가능한 범위는 정보시스템 사업보안감리에 속한다고 할 수 있다.

3. 모바일 어플리케이션 개발 보안 점검항목

3.1 모바일 어플리케이션 보안 기법 도출

3.1.1 모바일 어플리케이션 보안 가이드라인 검토

모바일 어플리케이션의 보안 감리 기법을 도출하기 위해 신뢰성 있는 기관 및 기업에서 이미 SANS 및 OWASP, CWE 등을 참조하여 연구된 모바일 어플리케이션에 대한 보안 가이드를 수집하였다. 수집된 보안 가이드에는 모바일 어플리케이션 보안 취약점 진단 시 사용되거나 시큐어 코딩을 위한 가이드가 포함되어 있으며 이를 구분한 것이 <Table 3>과 같다.

〈Table 3〉 Mobile application vulnerabilities of institutions and enterprises[14][15][16][17][18]

	A	B	C	D	E
	MOSPA	Google	Fortify	KISA	Veracode
1	Input data validation	Data storage	Insecure Storage	Undefined function	Activity monitoring and data retrieval
2	API abuse	IPC use	Query String Injection	Minimum grant option	Unauthorized dialing, SMS, and payments
3	Security attributes	Permission	Access Control	External input data validation	Unauthorized network connectivity
4	Time and status	Network	Privilege Management	Safe management of sensitive data	UI Impersonation
5	Error Handling	Dynamic loading code	Bad Practices	Mobile platform security model validation	System modification
6	Code Quality	Input value validation		Language-based security vulnerability check	Logic or Time bomb
7	Encapsulation	User data operation		Safety assurance for the module	Sensitive data leakage
8		Encryption		Common infrastructure security check	Unsafe sensitive data storage
9				Known vulnerabilities	Unsafe sensitive data transmission
10					Hardcoded password/keys

Source : [14]-[18] Application vulnerabilities Summary
(MOSPA: Ministry of Security and Public Administration)

여러 기관과 기업에서 보안 가이드를 발표하였지만 비교적 비슷한 항목이 많으며 서로 부족한 부분을 교차 확인 할 수 있는 진단 항목이 필요하다. 따라서 이러한 항목을 하나의 항목으로 정리하고 정리된 보안 취약점 진단 항목에서 진단 항목을 도출하고자 한다.

3.1.2 모바일 어플리케이션 취약점 진단항목 도출

서로 성격이 다른 보안 가이드에서 어플리케이션 구현 후 보안성을 점검하기 위한 진단항목을 도출하여 보안 점검항목을 도출하기 위해, <Table 3>에서 수집한 보안 취약점들을 통하여 <Table 4>와 같이 모바일 어플리케이션 취약점 진단항목을 도출하였다.

〈Table 4〉 Mobile application vulnerabilities diagnostic items

	Diagnostic items	Explanation	Reference
1	Permission management	Access permissions and privileges of other applications such as data sharing and management	B3, C4 D1, D2 E1, E2
2	Input data validation	Validation of input data through the users	A1, A2 B5, B6 C2, D3
3	Important file management	Safety check of important files check	A3, B1 C1, D4 E8
4	External data transfer management	Important data encryption communicating with external data	A1, B4 B8, D8 E9
5	Component management	Abuse check of the used components	A3, B2 C5
6	Security program check	Data exposure and safety check in the program code	A4, A5 A6, A7 D6, E6 E7, E10
7	Data use policy management	Use of personal information notices and violation of the mobile platform, the security model, and user authentication	B7, C3 D5, D8 E5
8	Safety management for the open module	The public availability of the safety check for the open module	D7
9	DB data management	Maintaining the safety of the database data verification	A3, B1

3.1.3 기존 정보시스템 개발 점검항목 검토

기존의 정보시스템 감리지침이 모바일 어플리케이션 개발 시의 보안 취약점 존재 여부를 점검할 수 있는지에 대해 확인해 볼 필요성이 있다. 따라서 정보시스템 감리지침 객체지향·컴포넌트기반 개발 모델은 따로 보안에 대해 영역을 구분하지 않고 있어 <Table 5>와 같이 각 영역에서의 보안 부분의 검토항목 및 세부검토항목을 정리하여 도출하였다. 시스템 아카텍처 영역에는 시스템 도입계획 및 설계에 따라 도입/설치되었는지, 시스템의 구성요소에 대한 검증이 이루어졌는지를 검토한다. 응용프로그램 영역에는 사용자 접근 통제 및 보안사항이 적

정하게 구현되었는지를 확인하고, 데이터베이스 영역에서는 데이터에 대한 접근권한 및 통제가 설계에 맞게 구현되었는지, 데이터 중요도 및 데이터 암호화에 대한 설계 관점이 구현되었는지 점검한다.

〈Table 5〉 Security check items of object-oriented, component-based development model [19]

Audit Domain		Check items
System architecture	1.1	Did you install the system according to the plan and design of the system ?
		Verification activities of the security requirements of security solutions (security level, features, etc.). The installation of security solutions, security and consistency of the security design.
	1.2	Did you verify the system components ?
		Verification activities of the performance, reliability , availability, and security of system software architecture.
Application program	2.1	Did it implement the user access control and security issues ?
		Access control of application systems, control and audit functions. Security issues related to the process of the application system and user requirements.
Database	3.1	Did it implement the data access privileges and control ?
		Data access privileges and control according to the design. Implementation of the important data and data encryption

Source : [19] Information System Audit Checks V3.0 Summary

<Table 6>에서는 모바일 서비스 사업 중 어플리케이션을 개발할 때 <Table 5>에 정의된 기존의 감리지침 및 방법이 모바일 어플리케이션에 대한 보안성과 위협을 점검할 수 있는지에 대한 여부와 본 논문에서 도출한 보안 취약점 진단 항목이 어플리케이션에 존재하는 위협에 대해 대응할 수 있는지 확인해 보았다.

부분가능의 경우 상세 위협에 중 일부분만 점검이 가능한 경우를 뜻한다. <Table 6>에서 어플리케이션 취약점을 이용한 권한공격의 보안위협 감리지침은 <Table 4>의 어플리케이션 취약점 진단항목 9개 중에서 4개의 항목인 입력값 검증, 보안 프로그램 확인, 권한관리, 컴포넌트관리가 감리점검 가능하여 부분 가능한 것으로 표시하였다. 또한 사용자 부주의로 인한 중요 정보 유출항목은 9개의 진단항목 중 중요파일 관리, DB 데이터 관리, 보안 프로그래밍 확인, 권한 관리 및 컴포넌트 관리와 데이터 사용정책 관리 항목의 6개의 항목이 감리점검 가능하여 부분적으로 점검 가능한 부분이다.

〈Table 6〉 Audit availability for security threats

Applications security threats	Threat security vulnerability diagnosis item	Audit availability	Note
Malicious modification and redistribution of the normal application	Encryption of the source code External data transfer management	Impossible	-
Attacks exploiting application permissions	Input data validation Security program check Permission management Component management	Partially available	2.1
Important user information leakage due to carelessness	Important file management DB data management Security program check Permission management Component management Data use policy management	Partially available	2.1 3.1
authentication bypass for applications that require authentication	Input data validation Component management Data use policy management	Impossible	-
DoS attacks against application	Security program check Safety management for the open module	Impossible	-
Internal / external transfer data interception	External data transfer management Component management	Impossible	-

3.2 모바일 어플리케이션 보안 점검항목 도출

3.2.1 모바일 어플리케이션 개발 보안 점검항목 제안

안드로이드 기반의 모바일 어플리케이션이 가지는 특징인 권한과 컴포넌트 점검 및 비밀번호 혹은 세션, 이미

지 파일 등을 저장하기 위한 SQLite와 같은 자체 데이터 저장소에 대한 점검 항목과 JAVA 언어 기반인 어플리케이션이 가질 수 있는 취약점에 대한 점검 항목을 도출하였으며, 감리 시 확인해야 할 산출물에 대해 정의 하였다. 또한 본 점검 항목은 어플리케이션 구현에 있어서의 보안성 여부를 확인하기 위한 것으로, 주된 감리영역은 응용시스템 영역이며 어플리케이션이 가지는 데이터베이스에 대한 영역으로 이루어진다[12].

점검 항목 중 '권한(permission)관리를 하였는가?'는 안드로이드의 고유 Permission 보안에 대한 항목이지만 기타 운영체제에서도 '어플리케이션에서의 시스템 권한을 사용하는 명령의 사용 자제' 등으로 항목화 할 수 있다. 'Activity, broadcast Receiver, ContentProvider, Sercvice와 같은 컴포넌트에 대해 관리 하였는가?'를 제외한 항목들은 타 운영체제에서도 해당 운영체제의 어플리케이션 개발 언어로 변경하여 사용 가능한 항목들이다. 또한 데이터베이스 영역의 점검 항목은 단말 내부에서 SQLite를 사용하는 운영체제인 경우 점검 가능하다.

한국정보화진흥원의 정보시스템 감리기준[12]에 의하여 모바일 어플리케이션 보안 점검항목을 구분하면 <Table 7>과 같이 도출할 수 있다. 권한관리, 외부 데이터관리, 중요 데이터 관리, 외부 전송 데이터 관리, 컴퓨넌트 관리, 보안 프로그래밍 여부, 데이터 사용정책 관리, 공개 모듈의 안정성, 데이터베이스의 데이터 안전성 관리와 같이 9개의 항목으로 나눌 수 있다. 각 항목 당 상세 점검 항목은 총 28개로 각 점검항목에 대한 구체적인 보안 점검 기능으로 구성되어 있다.

〈Table 7〉 Mobile application security audit check items

	Domain	OS	Check items Summary	Count
1	Application	Partial Common	Did you manage the permission?	3
2	Application	Common	Did you check the external data?	3
3	Application	Common	Did you manage the important data?	4
4	Application	Common	Did you manage the external transfer data?	3
5	Application	Android	Did you manage the components such as activity, ContentProvider, Broadcast receiver, Sercvice ?	3
6	Application	Common	Did you check the security program?	4
7	Application	Common	Did you manage the data management policies?	3
8	Application	Common	Did you check the safety for public inspection module ?	2
9	Database	Common	Did you secure the data in the database?	3
Total Check Items				28

산출물로는 기능 정의서, 보안 코딩 정의서, 컴포넌트 정의서, 응용프로그램 구조도, 데이터베이스 구조도, 보안 점검 결과서가 있다. 기능 정의서는 개발되는 어플리케이션의 기능별 요청 권한과 기능에 대한 설명을 작성하여 관리하며, 보안 코딩 정의서는 응용시스템에서 발생할 수 있는 보안 취약점을 사전에 방지하기 위해 위협에 대한 방어 코드를 정의해 관리해야 한다. 컴포넌트 정의서는 사용된 컴포넌트간의 기능별 구조를 정의하며 응용프로그램과 데이터베이스 구조도는 응용프로그램의 동작 흐름과 데이터베이스의 속성 등 구성에 대해 구조화하여 관리한다. 보안 점검 결과서는 취약점 진단을 통해 얻어진 결과로 이행점검을 수행할 수 있도록 상세화하여 작성한다.

3.2.2 모바일 어플리케이션 보안 상세점검 방법

9가지의 점검 내용은 28가지의 상세 점검 항목으로 세분화 된다. 세분화된 내용은 실제 보안성 구현 여부를 확인하며 항목의 필요성과 확인 방법, 점검 방법은 아래와 같다.

1) 권한관리 점검항목

〈Table 8〉 Permission management check items

Classification		Check Items
1		Did it manage the permission management?
Details	1-1	Does it match the function description feature and the actual app function?
	1-2	Did it grant the only minimum required function?
	1-3	Did it not ask for permission to spare for future versions of the function?

가) 필요성

안드로이드에서 제공하는 대부분의 기능들은 운영체

제로부터 권한을 얻고 사용해야 하는 보안절차를 가진다. 따라서 고의적이든 실수로든 추가적인 기능이 부여될 시 악의적인 사용자는 사용자 모르게 해당 기능을 악용할 위험이 있다.

나) 확인 및 조치 방법

AndroidManifest.xml 파일과 기능 정의서에 정의된 기능과 권한을 확인하고 기능에 필요한 권한만이 부여됐는지 확인한다. 향후 업데이트를 위해 권한을 미리 부여하는 방법은 지양하며, 최소한의 권한만 사용하도록 권고한다[15][16][17][18].

2) 외부 입력데이터 점검항목

〈Table 9〉 External data check items

Classification		Check Items
2		Did you do the verification of external data?
Details	2-1	Do you prepare the SQL Injection of important information?
	2-2	Did you remove inputs, including routine check for Null values and special characters?
	2-3	Didn't it create the file name directly about external inputs?

가) 필요성

사용자가 입력하는 값을 아무런 검증 없이 사용 시 개발자가 의도치 않은 보안 위협에 노출된다. 안드로이드 또한 쿼리 문을 통한 삽입공격과 Directory Traversal 등을 통한 파일 접근 등과 비슷한 공격에 취약할 수 있다. 또한 외부에서 입력되는 값이 직접적으로 파일이름을 지정할 시 만약 단말이 Rooting되어 있다면 '../../../' 와 같은 디렉터리 이동을 통해 알려진 경로의 파일들에 접근할 수 있으며 코드에 따라 개발자가 의도하지 않은 파일이 삭제, 노출될 수 있다. 또한 입력 값의 제한이 없을 경우 버퍼오버플로우 등에 위험할 수 있다[14].

나) 확인 및 조치 방법

'Query'문을 사용시 사용자 입력 값을 포함하는 동적 SQLite Query문을 바로 사용할 시 취약하다고 볼 수 있다. 따라서 스트링 연산('+')을 통해 입력 값을 바로 받아 사용하는 것을 지양하고, "selectionArgs"를 통해 입력 값을 파라미터 치환문자('?')로 받아오는 매개변수 형식의 Query 문을 사용한다[16].

또한 입력 값을 받는 부분에 사용되는 문자열에 대해 특수문자를 제거하거나 입력 길이 제한 등의 유효성 검사하는 부분 없이 입력 값을 바로 사용하는 경우 취약하며 유효성 검사를 실시하는 프로세스를 구현한다[14].

3) 중요 데이터 관리 점검항목

〈Table 10〉 Important data management check items

Classification		Check Items
3		Did it manage the important data? (Personal ID or account information, Location information, Finance information, Authentication Cache information)
Details	3-1	Did you use the secure encryption algorithm about important information?
	3-2	Did you use unintentional factors other applications can not access the value during creating a file?
	3-3	Did you store inside the application directory instead of external storage device?
	3-4	Did you generate the secure random number generator to generate the key value like SecureRandom?

가) 필요성

기본적으로 어플리케이션이 사용하는 파일에 대한 접근은 해당 어플리케이션만 가능하다. 제3자의 접근이 가능한 외부저장장치에 중요 정보 저장시 프로그램 삭제 후에도 남아있을 수 있어 정보의 노출이 있을 수 있다. 하지만 내부 디렉터리 또한 Rooting이나 에뮬레이터를 통해서 접근 가능하다. 따라서 중요정보를 저장한 데이터에 대한 안전한 암호화가 필요하며 암호화를 위한 안전한 키 생성이 필요하다[16].

나) 확인 및 조치 방법

외부저장장치에 저장되는 정보와 중요도를 확인한다. 내부 디렉터리에 데이터를 저장할 때에도 다른 어플리케이션이 접근할 수 있도록 할 수 있으며, 파일의 사용권한에 대한 설정이 'MODE_WORLD_READABLE', 'MODE_WORLD_WRITABLE' 일 경우 가능하게 된다[14].

또한 상대적으로 안전하지 않은 MD5, DES 등의 암호화 알고리즘의 사용과 예상 가능한 난수 생성 함수인 'Math.random()'의 사용은 지양하며 안전한 암호화 알고

리즘의 사용과 'SecureRandom()'의 사용을 권장한다 [14][17][18].

4) 외부 전송 데이터 점검항목

〈Table 11〉 External transfer data check items

Classification		Check Items
4		Did you manage the external transfer data ?
Details	4-1	Did you use the secure encryption algorithm communicating with the outside via sockets ?
	4-2	Did you recommend the 3G network instead of wifi network sending important information ?
	4-3	Did you minimize the unnecessary information transmission ?

가) 필요성

사용자의 민감한 데이터를 평문 형태로 Wifi를 사용하여 외부 서버로 전송하는 경우 악의적인 사용자가 정보를 가로챌 수 있다. 또한 암호화 시 보안수준이 낮은 알고리즘 사용 시 쉽게 복호화 될 위험이 있다.

나) 확인 및 조치 방법

소켓 통신시 아무런 암호화 로직이 없으면 취약하다고 볼 수 있다. 필요이상의 정보를 외부로 전송하는 것을 지양하고 필요한 경우 'SSLSocketFactory'를 사용하여 SSL을 통한 통신 암호화와 3G 네트워크의 사용을 권장한다. 하지만 퍼포먼스 문제와 같은 특별한 상황 시 단대단 암호화 솔루션 도입 등의 대체 수단을 강구해야 한다 [14]17].

5) 컴포넌트 관리 점검항목

〈Table 12〉 Component management check items

Classification		Check Items
5		Did you manage the components such as activity, ContentProvider, Broadcast receiver, Sercvice ?
Details	5-1	Did you set random components can not be activated from outside ?
	5-2	Did you avoid the sharing through sensitive data component ?
	5-3	Did you use the Intent to secure components?

가) 필요성

안드로이드 어플리케이션은 인텐트 필터와 인텐트 호출을 통해 어플리케이션간 컴포넌트를 호출할 수 있다. 컴포넌트에 대해 아무런 접근통제를 하지 않고 인텐트 필터 사용시 개발자의 의도와 다르게 악의적인 어플리케이션에 의해 특정 UI, 서비스 등이 실행될 수 있다. 중요한 정보를 담고 있는 UI 라면 정보가 노출될 수 있다.

악의적인 사용자는 어플리케이션의 구성요소 중 Manifest파일에 PackageManager를 통해 접근 하여 어플리케이션의 인텐트를 확인할 수 있으며, 동일한 인텐트를 사용하여 정보를 수집할 수 있다. 동일한 인텐트, 즉 인텐트 필터의 Action/Data/Type 생성 시 어떤 어플리케이션을 사용할 지 Resolver Activity에 의해 사용자 선택을 하게 되며, 선택된 어플리케이션은 시스템 레벨의 ID를 가지고 보안절차를 우회하게 되어 악의적인 사용자의 어플리케이션을 실행하게 끔 할 수 있다. 만약 해당 컴포넌트가 서비스라면 단말에 먼저 설치된 어플리케이션의 서비스로 정보가 전달된다[14][20][21].

나) 확인 및 조치 방법

아무런 권한이나 옵션 적용 없이 인텐트 필터를 사용 시 사용된 컴포넌트 명을 알아내어 해당 액티비티를 악의적인 어플리케이션을 통해 실행시킬 수 있다.

의도하지 않은 어플리케이션의 컴포넌트 사용을 방지하기 위해, Manifest 파일에서 인텐트 필터를 사용하는 컴포넌트에 대해 권한을 적용한다. 특정 액티비티를 사용할 어플리케이션은 액티비티 사용에 대한 <permission androidname="액티비티명">와 같은 권한을 필요로 하며, <activity android:permission="액티비티명">과 같이 사용한다. 해당 권한을 요청하는 어플리케이션 설치 시 사용자는 권한의 요청을 확인할 수 있다 [14][20][21].

사용되는 인텐트는 프로그래밍적으로 어떻게든 노출될 수 있는 부분이다. 따라서 인텐트를 통해 중요 정보를 전달하는 것을 지양하고 필요 시에는 정보를 여러 조각으로 나누어 전송한다. 또한 암시적 인텐트가 필요한 네이티브 어플리케이션 외의 내부 메시지를 위한 어플리케이션에 대한 인텐트의 전송은 명시적 인텐트의 사용을 권장한다.

6) 보안 프로그래밍 점검항목

〈Table 13〉 Security program check items

Classification		Check Items
6		Did you check the exposure and safety of the data during program coding?
Details	6-1	Did you check the error handling and important information not included in the error message?
	6-2	Did you process the data with the Public and Private array type?
	6-3	Did you avoid the exposure of important information for debugging code ?
	6-4	Didn't you code the hard-coding the important information, such as account information in the source code?

가) 필요성

코드 작성이나 디버깅의 편의를 위해 오류 메시지에 과도한 정보를 나타내거나 소스코드 내에 주석으로 과도한 정보를 나타내는 경우 역공학이나 강제 오류메시지 출력을 통해 해당 정보들이 노출될 수 있다. 또한 중요 데이터에 대해 캡슐화 오류를 범하는 경우 허용되지 않는 사용자들 간의 데이터 수정이 발생할 수 있다.

나) 확인 및 조치 방법

오류처리 시 'printStackTrace();' 와 'getMessage()'를 통한 상세 오류 메시지의 출력을 확인하고 중요정보가 임의로 사용되었는지 확인한다. 또한 중요 정보가 쓰이는 소스코드에 해당 정보가 하드코딩 되어있는지 확인한다. 개발자의 실수로 private로 선언한 배열이 public으로 선언되었는지 확인한다[14][18].

private 배열에 대해 public에서 사용시 private 배열에 대한 복제본을 사용하게 하여 의도하지 않은 직접적인 수정을 방지하게 한다[14][18].

7) 데이터 사용 정책 관리 점검항목

〈Table 14〉 Data management policy check items

Classification		Check Items
7		Did you manage the data management policies?
Deta Ils	7-1	Did you notice the application to the user for the personal information?
	7-2	Did you check the violation of the mobile platform security model to deal with the important information from the application ?
	7-3	Did you check the user authentication for access to the application data through Query?

가) 필요성

개인정보보호법에 의해 개인정보를 사용함에 있어서 사용자에게 적절 히 알려야 한다. 비용이 발생하거나 금융 이용 정보 등이 사용되는 중요한 정보가 오가는 어플리케이션의 경우 Rooting 등 모바일 플랫폼 변경 여부를 확인하여야 보안 모델의 보호를 받을 수 있으며, 정보에 대해 적절한 접근제어가 없을 시 악의적인 사용자는 허용된 것 이외의 정보를 열람할 수 있다.

나) 확인 및 조치 방법

어플리케이션의 설치 전 혹은 실행 전에 개인정보 사용에 대해 팝업 등을 사용하여 적절하게 알린다.

개발된 어플리케이션이 Rooting으로 인해 받을 수 있는 보안 위협을 판단한 후 Rooting된 단말에 대한 접근 허용 여부를 결정한다. Rooting된 단말에서 발견될 수 있는 '/system/bin/su', '/system/xbin/su', '/system/app/superuser.apk', '/data/data/com.noshufou.android.su'의 존재여부를 확인하거나 'Runtime.getRuntime().exec("su");' 를 통해 실제 루트 권한을 요청해 보는 방법을 사용하여 Rooting여부를 판별한다.

사용자가 제어하는 기본키를 포함한 SQLite Query문을 실행 시 인증 정보가 없는 경우 악의적인 사용자는 허가 받지 않은 레코드를 볼 수 있으므로 정보열람 시에는 인증 절차를 필수로 추가하도록 한다. 예를 들어 학번만으로 성적조회가 가능할 시 학번을 유추하여 불특정 다수의 정보를 획득할 수 있으므로, 사용자 ID와 같은 다른 인증 정보를 추가하여 이를 방지한다[16][17][18].

8) 공개 모듈 관리 점검항목

〈Table 15〉 Public module safety check items

Classification	Check Items
8	Did you check the safety for public inspection module ?

Details	8-1	Did you document the use of open modules?
	8-2	Did you conform the safety in use of the public module?

가) 필요성

프로그래밍 시 모듈의 재사용 등 공개된 모듈의 사용은 불가피하다. 이러한 모듈의 사용을 관리하고 안전성 여부를 확인하지 않으면 공개된 부분이기 때문에 더욱 많은 대상에게 위험으로 노출될 수 있다[15].

나) 확인 및 조치 방법

공개된 모듈의 사용을 확인하고 이에 대해 정리된 문서의 존재여부를 파악한다. 이렇게 사용된 모듈에 대한 보안 취약점이 관리되고 조치되고 있는지 확인한다.

9) 모바일 안드로이드 데이터베이스 데이터 관리 점검 항목

〈Table 16〉 Data security check items

Classification		Check Items
9		Did you secure the data in the database?
Details	9-1	Are there the permission to access the database files properly?
	9-2	Are there the unnecessary information, such as personal and financial information?
	9-3	Did you perform encryption for the important information stored in the terminal?

가) 필요성

편의성을 이유로 어플리케이션이 생성하는 데이터베이스 파일에 과도하게 많은 중요정보를 저장하고 사용하는 경우가 많다. 이에 대한 보안 요소를 적용하지 않거나 파일에 대한 접근 권한이 잘못 적용되었을 시 민감한 데이터의 노출이 있을 수 있다.

나) 확인 및 조치 방법

안드로이드는 UNIX의 보안요소를 일부 사용하며 파일에 대한 접근권한을 소유자ID/그룹ID/기타로 나누어 관리한다. 기본적으로 생성되는 데이터베이스 파일에 대하여 소유자 및 그룹에 대해 접근 가능하게 되어 있으나 종종 소유자 권한이 아닐 시에도 접근이 가능한 경우가 있어 확인이 필요하다.

중요 정보 저장 시 암호화를 하여 데이터베이스에 저장하는 것이 비교적 안전하다. 하지만 키관리의 문제 등으로 100% 안전한 것은 아니므로 중요 정보에 대한 저장을 최소화 하는 것이 중요하다[14][17].

Ⅳ. 제안의 검증

본 논문에서 제안한 점검 항목의 실효성을 검증하기 위해 정부에서 국민을 대상으로 실제 서비스 중인 어플리케이션에 대한 보안 점검과 개발 및 보안, 감리 등 현업 종사자를 대상으로 점검 항목에 대한 실효성 여부를 설문조사를 통해 판단하고자 한다.

4.1 모바일 어플리케이션 개발 보안 점검항목 적용 검증

모바일 어플리케이션 개발 보안 점검항목을 적용하여 실제 어플리케이션에 대한 실효성을 검증하기 위해 공공기관과 행정기관과 민간의 안드로이드 기반 어플리케이션을 수집 및 분석하였다. 하지만 개발단계 산출물을 확인 할 수 없고 동적 분석과 소스코드 분석만을 통해 기능을 이해하였음에 점검할 수 없는 항목이 있는 한계가 있다. B어플리케이션의 경우 11개의 점검항목 중에서 3개의 항목을 확인할 수 없었고, 전체적으로는 7-3인 쿼리를 통한 사용자 인증을 확인할 수 없었다. 전체적으로는 44개의 점검항목 중 5개가 확인이 되지 않았으며, 11%의 확인이 되지 않은 한계점이 있다. 확인이 안 된 3개의 점검항목 <7-3>은 쿼리를 통한 사용자 인증이므로 실사용자는 사용하지 않는 기능이므로 전체 모바일 어플리케이션 보안 점검에는 영향이 미미하다고 할 수 있다.

<Table 17>은 대한적십자사 혈액관리본부 '스마트헌혈', 보건복지부의 '응급의료1339', 우정사업본부의 '우체국택배&EMS', Life Market '쿠차'의 4개의 어플리케이션에 대한 전체 소스코드 및 동적 분석을 실시한 전체 결과를 나타내며, 취약점이 발견된 항목에 대하여 정리하였다. 4개의 어플리케이션 'A', 'B', 'C', D'는 무작위로 4개의 어플리케이션을 배정하였다. 또한 'o'는 안전, 'x'는 취약'을 나타내며 '-'는 기능을 사용하지 않아 확인하

지 않은 항목이거나 소스코드만으로는 확인할 수 없음을 나타낸다. 안전도의 산출 방법은 어플리케이션 별 진단 항목 수에 따른 탐지 항목 수를 백분율로 나타내었다.

〈Table 17〉 Results of the application development security audit

Classification	Check Items	Application			
		A	B	C	D
2-2	Did you remove inputs, including routine check for Null values and special characters?	x	x	o	x
3-1	Did you use the secure encryption algorithm about important information?	x	x	o	o
3-3	Did you store inside the application directory instead of external storage device?	o	o	x	o
4-1	Did you use the secure encryption algorithm communicating with the outside via sockets?	o	-	o	x
4-2	Did you recommend the 3G network instead of wifi network sending important information ?	x	-	x	x
5-3	Did you use the Intent to secure components?	x	o	o	o
6-4	Didn't you code the hard-coding the important information, such as account information in the source code?	x	o	o	o
7-2	Did you check the violation of the mobile platform security model to deal with the important information from the application ?	x	x	x	x
7-3	Did you check the user authentication for access to the application data through Query?	-	-	x	-
9-2	Are there the unnecessary information, such as personal and financial information?	o	x	o	o
9-3	Did you perform encryption for the important information stored in the terminal?	o	x	o	o
통 계		68 %	71 %	82 %	79 %

비교적 중요한 정보를 다루는 C어플리케이션은 입력 값 검증 및 인증서를 사용하는 등 약 82% 이상의 안전성을 나타내고 D어플리케이션은 데이터 저장 시 비교적 안전한 암호화 사용을 하며 약 79%의 만족할 만한 안전성을 보이고 있으나 A, B 어플리케이션은 약 68~71%의 안전성을 나타내고 있음을 확인할 수 있다. 이렇게 어플리케이션에 위협이 존재함을 확인하고 정량적으로 위험 정도를 확인하는데 본 논문이 제안한 점검항목이 적합함을 알 수 있다.

4.2 설문을 통한 검증

본 조사에서 설문 대상자로 선정한 모집단은 모바일 보안에 대한 이슈가 최근 활성화 됐기 때문에 모바일 보안 업무 수행자의 수요가 적어 표본에 대한 적절한 수집이 어려워 모바일 업무 경험을 가진 보안실무자, 감리원, 개발자 등 IT 각 분야의 실무자를 설문의 대상으로 하였으며 설문 응답자 중 모바일과 관련된 업무 수행 경험이 있는 응답자는 27명이었으며, 27명에 대한 설문 대상자의 분포는 보안실무자 34%, 개발자 26%, 감리원 40%로 구성되어 있다. 설문은 응답자의 특성을 확인한 후 각 점검항목별 세부점검 항목에 대해 리커트 5점 척도법을 이용하였고 기타 의견을 수렴하는 방법을 택하였다.

모바일 어플리케이션 개발 감리 시 보안 요소의 확인이 필요성에 대해서 설문 대상자 100% 모두 필요하다고 응답하였다. 보안 실무자와 개발자, 감리원의 전체 평균이 각각 4.4점, 4.3점, 4.4점으로 나타나 4점 이상으로 '필요' 척도에 부합한 것으로 나타났다. 또한 '매우필요하다'와 '필요하다'를 적합으로, '보통이다'를 보통, '필요없다'와 '전혀 필요없다'를 부적합으로 보았을 때 '적합'이 전체 92% 이상으로 나타나 모바일 개발 보안 감리 점검 항목이 감리활동에 있어서 사용하기 적합한 것으로 판단된다.

〈Table 18〉 Results of detailed check item verification

Classification		Suitability			Security Executives	Developers	Auditors	Average	SD
		Appropriate	middle	Inappropriate					
Permission Management	1-1	26	1	0	4.7	4.6	4.1	4.4	0.26
	1-2	27	0	0	4.7	4.6	4.2	4.4	0.22
	1-3	22	5	0	4.2	3.7	3.9	4.0	0.21
Suitability		92.6 %	7.4 %	0.0%	Appropriate				
Input Validation	2-1	23	4	0	4.6	4.1	4.7	4.5	0.26
	2-2	24	3	0	4.3	4.6	4.6	4.5	0.14
	2-3	24	1	2	3.8	4.1	4.7	4.3	0.37
Suitability		87.7 %	9.9 %	2.5%	Appropriate				

Critical Data Management	3-1	26	1	0	4.8	5.0	4.2	4.6	0.34
	3-2	25	2	0	4.6	4.3	4.2	4.3	0.17
	3-3	23	3	1	4.0	4.0	3.9	4.0	0.05
	3-4	26	1	0	4.3	4.4	4.2	4.3	0.08
Suitability		92.6 %	6.5 %	0.9%	Appropriate				
Data Transfer Management	4-1	27	0	0	4.7	4.6	4.8	4.7	0.08
	4-2	21	6	0	4.0	3.7	4.8	4.3	0.46
	4-3	26	1	0	4.3	4.1	4.9	4.5	0.34
Suitability		91.4 %	8.6 %	0.0%	Appropriate				
Components Management	5-1	26	1	0	4.2	4.3	4.1	4.2	0.08
	5-2	25	2	0	4.1	4.1	4.1	4.1	0.00
	5-3	26	1	0	4.3	4.0	4.0	4.1	0.14
Suitability		95.1 %	4.9 %	0.0%	Appropriate				
Secure Coding	6-1	27	0	0	4.9	4.9	5.0	4.9	0.05
	6-2	25	2	0	4.4	4.1	4.9	4.6	0.33
	6-3	26	1	0	4.7	4.7	5.0	4.8	0.14
	6-4	27	0	0	4.8	4.7	5.0	4.9	0.12
Suitability		97.2 %	2.8 %	0.0%	Appropriate				
Data Use Policy Management	7-1	24	3	0	4.4	4.6	4.2	4.4	0.16
	7-2	19	8	0	4.0	4.0	4.1	4.0	0.05
	7-3	22	4	1	4.2	4.3	4.2	4.2	0.05
Suitability		80.2 %	18.5 %	1.2%	Appropriate				
Public Module Management	8-1	26	1	0	4.2	4.0	4.2	4.1	0.09
	8-2	26	1	0	4.3	4.3	4.4	4.3	0.05
Suitability		96.3 %	3.7 %	0.0%	Appropriate				
Secure DB Management	9-1	27	0	0	5.0	4.9	4.6	4.8	0.17
	9-2	27	0	0	4.9	4.6	4.6	4.7	0.14
	9-3	27	0	0	5.0	4.7	4.7	4.8	0.14
Suitability		100.0 %	0.0 %	0.0%	Appropriate				
Total		92.6 %	6.9 %	0.5%	4.4	4.3	4.4	4.4	0.16

V. 결론 및 향후 연구 과제

기존의 모바일과 관련된 정보시스템 개발 사업에서의 점검항목은 모바일에서 존재할 수 있는 보안 위협에 대한 특수성을 만족하지 못하며 구체적인 항목이 부재하여 일관적이고 객관적인 감리 수행의 어려움을 알 수 있다. 이에 본 논문에서는 모바일 서비스 환경에서 발생할 수 있는 보안 위협 중 어플리케이션의 개발 시 보안 위협에 대응하기 위한 방안에 대해 분석하였으며 이를 모바일 어플리케이션 개발 보안 점검항목으로 구체화하였다. 점검항목에 대한 점검 방법 및 조치 방안은 안드로이드를 기준으로 하였으며, 각 운영체제별 어플리케이션 개발 시 참조할 수 있다. 구체화된 점검항목은 실효성 검증을 통해 그 필요성을 확인하였으며 이를 통해 어플리케이션의 안전성 향상에 기여하였다. 하지만 운영체제의 업데이트 및 기술의 발전으로 인한 기능과 취약점에 대한 꾸준한 연구와 점검항목의 업데이트가 요구된다. 또한 본 점검 항목은 구현 단계에 초점을 맞춰 도출되었으며 요구사항 분석 및 분석, 설계 단계를 포함하는 것에 대한 연구와 모바일 생태계 전반을 아우르는 보안에 대한 모델의 개발이 필요하겠다.

본 논문을 기반으로 모바일 어플리케이션의 취약점을 이해하고 보안 위협을 최소화하는 보안 점검항목의 연구와 개발이 지속되길 기대해 본다.

REFERENCES

[1] Korea Information Agency, Information Statistical Compilations 2011, Seoul: Korea Information Agency, 2011.

[2] D. K. Seo, KT, mobile office business was resilient, Electronic newspaper, Oct 25, 2010.

[3] D. J. Kwon, Separating logical network, the area expansion with the virtual mobile office, Etnews, May 7, 2014.

[4] H. S. Hwang, K. H. Lee, A study on the mobile security model for secure smartwork, Review of Korea Institute of Information Security & Cryptology, Vol. 21, No 3, pp.22-34, 2011.

[5] Korea Information Agency, Smart Work Guidebook for a Enterprises, Seoul: The Korea Communications Commission, 2011.
[6] SK Telecom, Smartphone security threat trends and countermeasures, Seoul: SK Telecom, 2011.
[7] Aircube, Smart Mobile Solutions Configuration V4, Seoul: Aircuve, 2011.
[8] Symatec, Internet Security Threat Report, California:IBM, 2011.
[9] MITRE: http://cwe.mitre.org/
[10] ZDNet Korea, IPhone Hacks fake GSM base station?, Meganews, Jan 20, 2011.
[10] ZDNet Korea:http://blog.naver.com/PostView.nhn?blogId=rikajunsu&logNo=20121087032
[11] Android Police, "Exclusive: Vulnerability In Skype For Android Is Exposing Your Name, Phone Number, Chat Logs, And A Lot More", Apr. 2011
[12] National Information Society Agency, Information Systems Audit Standards Commentary, Seoul: National Information Society Agency, 2009.
[13] J. Y. Lee, D. S. Kim, H. W. Kim, A design of the information security auditing framework of the information system audit, Korea Society of Digital Industry and Information Management, Vol 6, No 2, pp.233-245, 2010.
[14] Korea Internet & Security Agency, Android-JAVA Secure Coding Guide, Seoul: Ministry of Security and Public Administration, 2011.
[15] Korea Internet & Security Agency, Mobile App Security Vulnerability Verification Guide, Seoul: Ministry of Security and Public Administration, 2011.
[16] Fortify, A Taxonomy of Coding Errors that Affect Security, Fortify Lab, 2011.
[17] Google, "Designing for Security", Aug. 2010
[18]VERACODE:http://www.veracode.com/directory/mobileapp-top-10,
[19] National Information Society Agency, Information System Audit Checks Cookbook V3.0, Seoul: Korea Information Agency, 2009.
[20] Dwivedi, Himanshu, Mobile Application Security, New York:McGraw-Hill, 2010.
[21] Rilly Hassell, Malicious Intent-Exploiting Android Activities to Escalate Privilege, Privateer Labs Research, May, 2011.

신 준 엽(Shin, Jun Youp)

- 2002년 2월 : 성결대학교 (학사)
- 2010년 2월 : 건국대학교 정보통신대학원(공학석사)
- 2008년 2월 ~ 2012년 3월 : ㈜씨아이솔루션 대리
- 2012년 12월 ~ 현재 : 경기지방경찰청 안산상록경찰서 경장
- 관심분야 : 정보시스템 보안, 정보시스템 감리, 프로젝트 관리, 소프트웨어 공학
- E-Mail : jyssbc@gmail.com

김 동 수(Kim, Dong Soo)

- 1981년 2월 : 광운대학교 전자계산학과(이학사)
- 2001년 2월 : 서울산업대학교 전자계산학과(공학석사)
- 2005년 2월 : 국민대학교 경영정보학과(경영학박사)
- 1991년 12월 : 전자계산조직응용기술사 취득
- 1995년 8월 : 정보통신기술사 취득
- 1998년 2월 ~ 현재 : (주)키삭 대표컨설턴트
- 2008년 3월 ~ 현재 : 건국대학교 정보통신대학원 겸임교수
- 관심분야 : u_city 감리, 프로젝트 관리, 정보시스템 감리, 소프트웨어 공학
- E-Mail : dskim@kisac.co.kr

한 기 준(Han, Ki Joon)

- 1979년 2월 : 서울대학교 수학교육학과(이학사)
- 1981년 2월 : KAIST 전산학과(공학석사)
- 1985년 2월 : KAIST 전산학과(공학박사)
- 1990년 1월 ~ 1991년 1월 : Stanford 대학 전산학과 Visiting Scholor
- 1985년 3월 ~ 현재 : 건국대학교 컴퓨터공학부 교수
- 관심분야 : 데이터베이스, GIS, LBS, 텔레매틱스, 정보시스템 감리 등
- E-Mail : kjhan@db.konkuk.ac.kr

김 희 완(Kim, Hee Wan)

- 1995년 8월 : 성균관대학교 정보공학과(공학석사)
- 2002년 2월 : 성균관대학교 컴퓨터공학과(공학박사)
- 1996년 5월 : 정보관리기술사 취득
- 2007년 1월 : 정보시스템 수석감리원 자격 취득
- 2001년 3월 ~ 현재 : 삼육대학교 컴퓨터학부 교수
- 관심분야 : 정보시스템 감리, 프로젝트 관리, 데이터베이스, 소프트웨어 공학
- E-Mail : hwkim@syu.ac.kr