Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.26, NO.1, Feb. 2016 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2016.26.1.177

안드로이드 접근성(Accessibility) 기능을 이용한 보안키패드의 취약점 공격 및 대응 방안*

이 정 웅,† 김 인 석‡
고려대학교 정보보호대학원

A Study on the Vulnerability of Security Keypads in Android Mobile Using Accessibility Features*

Jung-Woong Lee,† In-Seok Kim‡
Graduate School of Information Security, Korea University

요 약

최근 핀테크(FinTech) 산업의 활성화와 더불어 모바일환경에서의 금융거래가 증가함에 따라 이를 공격 대상으로 하는 악성 어플리케이션이 증가하고 있다. 악성 어플리케이션의 공격대상은 점점 많아지고 그 방법 또한 다양해지고 있어, 이에 따라 새로운 형태의 공격방법에 대해 미리 대비해야 할 필요성이 있다. 본 논문에서는 안드로이드 프레임워크의 접근성 서비스(Accessibility Service)기능을 이용해, 보안키패드의 입력 값을 탈취 할 수 있는 새로운 공격방법을 제시하고자 한다. 이러한 공격방법을 악용할 경우, 사용자가 보안키패드에 입력한 비밀번호 정보가 매우 쉽게 노출될 수 있음을 실험을 통해 보이고 이에 대한 대응 방안으로 접근성 사용에 대한 검증과 모바일 어플리케이션의 접근성 지침에 대한 개선을 제안한다.

ABSTRACT

As the fintech industry is growing at an incredible rate, mobile phones are positioned as the most important tool for financial transaction. However, with a rising number of malware applications, the types of attack and illegal access to mobile device are becoming more diverse and sophisticated. This paper studies the potential keylogger attack by exploiting the Accessibility Service in Android framework. This type of attack allows the malicious individual to use keylogger on the victim's Android mobile phone to steal passwords during mobile financial transaction regardless of security keypad setting. Lastly the paper proposes solutions to counter these types of attack by verifying the accessibility usage and amending the application guideline for accessibility.

Keywords: Accessibility Service, Smartphone Security, Security Keypads

I. 서 론

안드로이드는 전 세계적으로 보급률이 가장 높은 모바일 플랫폼으로 2014년 스마트폰 출하량에서 81.5%의 점유율을 차지한 것으로 나타났다[1]. 안드로이드 사용자가 늘어남에 따라 이를 대상으로 하는 악성코드가 점점 증가하고 있으며, 개인정보가 노출되는 사례 또한 지속적으로 증가하는 추세이다. 안

Received(15. 10. 2015), Modified(02. 01. 2016), Accepted(02. 01. 2016)
* 본 연구는 미래창조과학부 및 한국인터넷진흥원의 "2016년 고용계약형 정보보호 석사과정 지원사업"의 연구결과로 수행되었음
† 주저자, starryzen@korea.ac.kr
‡ 교신저자, iskim11@korea.ac.kr (Corresponding author)

랩의 보고서에 따르면 2014년 기준으로 보고된 스마트폰 악성코드 숫자가 143만 247개에 달한다[2]. 특히 핀테크(FinTech) 산업의 활성화로 모바일을 통해 다양한 금융 활동이 가능해지면서, 사용자의 정보가 위협받을 가능성이 더욱 높아진 상황이다.

금융 어플리케이션에서 사용하는 보안키패드의 입력 값은 사용자의 가장 중요한 정보 중 하나이다. 기존에 모바일 환경에서 보안 키패드의 값을 탈취하는 공격은 스마트폰의 터치좌표를 획득해서, 이를 분석해 보안키패드의 무작위 배열을 계산하는 방식이 제안되었다[3]. 하지만 이러한 방식은 터치좌표를 획득하기 위해서 시스템 권한을 필요로 한다는 점에서 공격대상의 한계가 존재한다.

본 논문에서 소개할 보안 키패드 공격방법은 안드로이드의 지원하는 공식적인 기능인 접근성을 이용한다는 점에서 기존의 키패드 공격방식과 차별점이 있다. 접근성 서비스를 이용한 공격방법은 아직까지 사용되지 않고 있지만, 만약 악용될 경우 스마트폰 보안에 취약점이 발생 할 수 있음을 보이고 이에 대한 대응 방안을 알아보고자 한다.

본 논문의 구성은 다음과 같다. 2장은 배경지식으로 안드로이드 접근성 서비스와 동작방식에 대한 내용을 설명하고, 3장에서 이에 대한 보안 문제에 대해 알아본다. 4장은 이를 이용한 공격 시나리오와 구현내용에 대해서, 5장에서는 이러한 공격에 대한 대응 방법을 소개 하며 마지막으로 결론을 통해 본 논문에 대한 정리와 향후연구에 대해서 설명하고자 한다.

II. 배경지식

본 장에서는 안드로이드에서 제공하는 접근성 기능과 활용 되고 있는 분야에 대해서 소개한다.

2.1 안드로이드 접근성 서비스

안드로이드 접근성 기능과 이를 활용한 어플리케이션은 사용자의 신체적인 필요에 맞게 스마트폰을 사용할 수 있도록 도와준다. 특히 시각적으로 불편한 장애인들이나 고령자에게 스마트폰을 좀 더 쉽게 사용할 수 있도록 도움을 줄 수 있는 기능을 기본적으로 탑재하고 있다[4].

TalkBack은 안드로이드의 대표적인 접근성 서비스로서, 시각장애인들이 화면을 터치하거나 선택했을

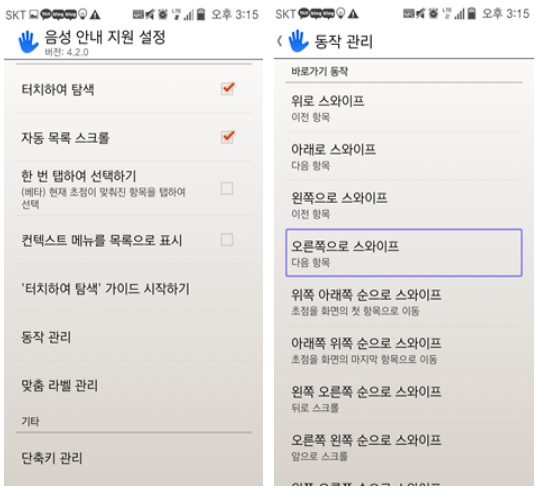

Fig. 1. TalkBack configuration screen

때 활성화되는 항목에 대한 내용을 안드로이드 TTS(Text To Speech)기능을 통해 음성으로 먼저 제공해 준다. 이후 음성으로 화면의 내용을 인지한 사용자가 추가적인 터치를 했을 때, 해당 항목에 대한 실행명령을 내리게 된다.[5] Fig.1.에서처럼 TalkBack설정에서 사용자는 눈으로 보지 않고 스마트폰을 조작할 수 있도록 특정 동작들에 대한 지정을 해줄 수 있다.

2.2 접근성 서비스 구조

안드로이드 플랫폼은 개발자들이 이러한 접근성 서비스를 활용해서 어플리케이션을 만들 수 있도록 API를 제공한다[6]. 접근성 기능은 Accessibility Service 클래스를 상속해서 구현할 수 있으며, Service형태로 동작하기 때문에 백그라운드에서 계속해서 동작할 수 있다. AccessibilityService 클래스에서는 사용자의 터치 동작을 통해 화면의 변화가 발생하면 AccessibilityEvent 값을 수신 받을 수 있으며, 이 이벤트는 Table 1.과 같이 사용자의 조작에 따라 여러 타입으로 구분할 수 있다[7][8]. 이벤트에는 화면을 구성하는 레이아웃의 뷰(View)들에 대한 리소스 정보들이 담겨 있어, 이 정보를 바탕으로 사용자에게 피드백을 줄 수 있는 데이터를 구성한다.

Table 1. Main accessibility event type

Event Type	Description
TYPE_VIEW_CLICKED	Represents the event of clicking on a View like Button
TYPE_VIEW_FOCUSED	Represents the event of focusing a View
TYPE_WINDOWS_CHANGED	Represents the event of changes in the windows
TYPE_VIEW_TEXT_CHANGED	Represents the event of changing the text of an EditText
TYPE_WINDOW_CONTENT_CHANGED	Represents the event of changing the content of a window

2.3 접근성 서비스의 활용

이러한 접근성 서비스는 일반 사용자들을 위해서도 활용될 수 있는데, 그 예로 메신저 서비스인 카카오톡[9]이 있다. 카카오톡에서는 다양한 어플리케이션의 알림을 모아서 잠금화면에 표시하는 "알림커버" 기능을 제공하는데, 접근성 기능은 카카오톡이 다른 어플리케이션의 알림데이터에 접근할 수 있도록 할 수 있다.

또한 스마트폰 바이러스 백신 어플리케이션인 중국의 Qihoo 360 Security[10]에서도 접근성을 활용하고 있다. 이 어플리케이션에서는 사용자 단말에 설치된 어플리케이션의 자동시작 등의 행위를 제어하고 보안 기능의 강화를 위해, Fig.2.에서처럼 접근성 활성화를 위한 권한을 필요로 한다.

III. 접근성 서비스의 보안 문제

본 장에서는 2장에서 살펴본 접근성 서비스의 보안 문제점, 접근성 승인 등에 대해서 알아본다.

3.1 보안상의 문제

2장에서 살펴본 것처럼 Accessibility Service는 Accessibility Event를 수신할 수 있다. 이 이벤트로 부터 AccessibilityNodeInfo 객체 값을 가져올 수 있으며 이로부터 화면상의 정보를 얻어낼 수 있게 된다. 각 Event type 별로 사용할 수 있는 메소드가 다르지만, 보안상 문제가 되는 정보들은

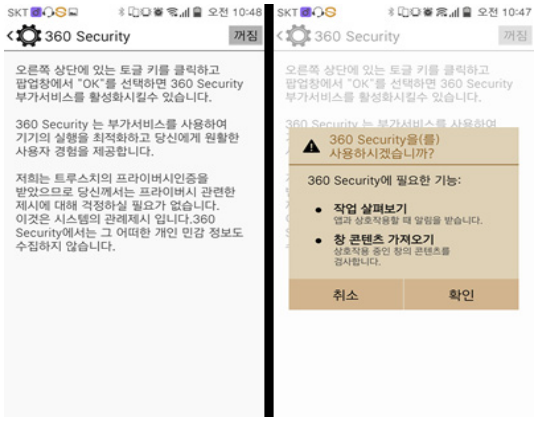

Fig. 2. accessibility service of 360 Security

getText()와 getContentDescription() 메소드만을 사용하여 가져올 수 있다. Fig.3.은 이메일 어플리케이션을 사용 중일 때 백 그라운드에서 동작하는 접근성 서비스가 화면상의 모든 노드에 대한 정보를 getText() 함수를 통해 수집한 후 로그로 표시한 것이며, 화면상의 대부분의 정보가 그대로 노출되는 것을 확인할 수 있다.

이미 표시된 정보 뿐만 아니라 사용자의 터치 입력에 대하여 터치한 뷰에 대해 정보를 받아 올 수 있다. 이것은 사용자가 보고 있는 화면의 정보 뿐 아니라 잠금화면 비밀번호, 모바일 뱅킹, 신용카드 결제 어플리케이션 등에서 입력되는 중요한 개인정보가 다른 어플리케이션으로 노출 될 수 있음을 의미한다. 안드로이드 환경에서는 어플리케이션간의 약속된 통신 이외에 다른 어플리케이션의 정보를 가져올 수 없게 되어있지만 접근성 서비스를 사용하는 경우 예외가 발생하는 것이다.

Fig. 3. Exposure of sensitive information in email application

Table 2. AccessibilityService XML attribute

Attribute	Description
canRetrieveWindowContent	Attribute whether the accessibility service wants to be able to retrieve the active window content.
canRequestEnhancedWebAccessibility	Attribute whether the accessibility service wants to be able to request enhanced web accessibility enhancements
canRequestEnhancedWebAccessibility	Attribute whether the accessibility service wants to be able to request enhanced web accessibility enhancements.
canRequestFilterKeyEvents	Attribute whether the accessibility service wants to be able to request to filter key events.

3.2 접근성 서비스 활성화 승인

3.1절에서 살펴본 보안 문제점을 방지하기 위해 안드로이드에서는 접근성 기능에 대한 설정을 외부 어플리케이션에서 활성화 할 수 없도록 Setting.Secure 클래스에서 관리하고 있으며, 사용자가 직접 설정에 들어가서 활성화 하거나 시스템 권한을 가진 어플리케이션에서만 변경을 허용하도록 되어있다.

사용자가 시스템 설정의 접근성 메뉴에서 접근성 서비스항목을 활성화하려고 할 때, 안드로이드 시스템은 어떠한 접근성 서비스를 사용하게 될 것인지에 대한 권한에 대한 승인을 요청한다. 이러한 권한 목록은 어플리케이션 개발 시 접근성을 시스템에 등록하기 위해 작성하게 되는 XML정보에 따라서 달라진다. Table 2.는 접근성 활성화시 표시되는 XML Attribute와 얻을 수 있는 정보에 대한 설명이다.

사실상 이 승인요청 메시지는 안드로이드에서 접근성 서비스의 보안 취약점에 대한 마지막 방어 수단이라 볼 수 있다. 하지만 사용자는 승인 후 보안 위협에 대해 정확히 인지하지 못할 수 있으며, 일반적인 악성 어플리케이션의 패턴처럼 본래의 기능을 수행하며 추가적으로 접근성 승인을 요청하였을 때 별다른 의심 없이 허용을 할 수 있다. 특히 실제로 장애가 있어 접근성을 활용해야만 하는 사용자의 경우에는 활성화 요청에 승인할 가능성이 더욱 높아진다.

3.3 일반 사용자 대상의 접근성 기능

2.3절 에서 살펴본 것처럼 접근성 서비스는 장애인들을 위한 기능이 아닌 일반 사용자를 대상으로 하는 어플리케이션도 존재한다. 이러한 어플리케이션들은 API 본래 목적에 맞게 사용하지 않음으로서 보안의 취약점을 발생하게 할 수 있다.

특히 Qihoo 360 Security와 같은 백신 어플리케이션은 대부분의 기본 어플리케이션 권한과 더불어 접근성 추가권한까지 요구하고 있다. 보안을 위해서 보안 어플리케이션에게 너무나 많은 권한을 허용하고 있는 상황이다. 현재로서는 이러한 어플리케이션에 대한 권한 허용이 문제가 없을지 모르지만, 자주 업데이트 되는 모바일 어플리케이션의 특성상 민감한 정보를 수집하는 기능이 추가될 가능성이 존재한다.

IV. 접근성 서비스를 이용한 공격모델

본 장에서는 접근성 서비스를 이용해 보안키패드의 비밀번호를 가져오는 공격방법과 실제 금융 어플리케이션에 적용한 결과에 대해서 설명한다.

4.1 공격 시나리오

공격자 어플리케이션은 다른 기능으로 위장한 어플리케이션 형태로 사용자 단말에 설치된다. 어플리케이션에는 AccessibilityService 클래스가 존재하며 Backgroud에서 계속 동작할 수 있다.

Fig.4.와 같이 Foreground에서 Target Application 이 실행될 때 화면의 정보가Android Framework의 Window Manager, Activity Manager 등으로부터 Background에서 동작하고

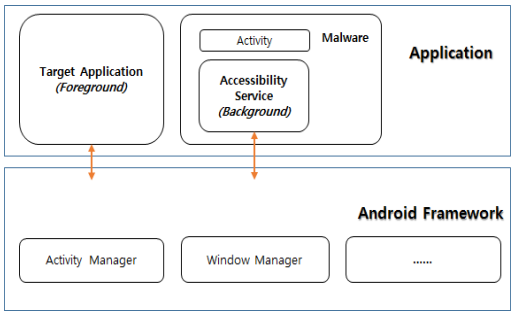

Fig. 4. Attacker's application activation mechanism

있는 Accessibility Service 로 전달된다.

Fig.5.에서는 공격자 어플리케이션이 설치된 이후의 과정을 보여준다. 사용자는 "폰지킴이(Phone Safer)"라는 다른 기능으로 위장한 악성 어플리케이션을 설치하고 어플리케이션을 사용한다. 위장된 기능을 수행함과 동시에 단말의 루팅 (Rooting) 상태를 체크하고, 만일 루팅이 되어있다면 시스템 권한을 이용해 Setting 어플리케이션 DB 의 접근성 컬럼 데이터 값을 조작하여 악성 어플리케이션의 접근성 설정을 사용자 몰래 활성화 할 수 있다.

루팅이 되어있지 않다면 Fig.6.과 같이 주기적으로 사용자에게 알림을 보내 접근성에 대한 활성화를 유도한다. 접근성 권한을 많이 사용할수록 더 많은 정보를 알아낼 수 있지만, 사용자로부터 권한 승인을 받을 수 있는 가능성은 낮다. 따라서 "창 콘텐츠 가져오기" 권한 만을 사용해 권한사용 표시내용을 최소화하고 사용자가 악성 어플리케이션임을 인지하기 어렵게 한다.

사용자가 시스템 설정에서 해당 접근성 서비스를 활성화 하는 시점부터 사용자의 조작을 감시할 수 있게 된다. 모든 상태에 대해서 감시하는 것은 시스템 속도를 저하시킬 수 있으며 저장된 결과에 대한 분석을 어렵게 하기 때문에 ActivityManager클래스의 getRunningAppProcesses() 이용해 금융 어플리

케이션과 같은 공격대상이 실행 중인지 체크하고, 실행 중이라면 onAccessibilityEvent() 메소드에

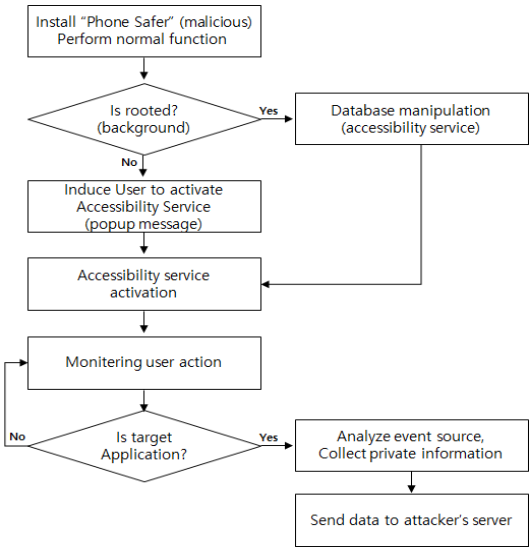

Fig. 5. Attacker's application process

Fig. 6. Requesting Accessibility Service Activation

서 TYPE_VIEW_CLICKED 타입으로 정의된 이벤트정보를 수신한다. 수신한 AccessibilityNodeInfo 객체의 getText(), getContentDescription() 메소드를 이용하면 클릭 이벤트가 발생한 뷰(View)의 Text와 ContentDescription 값을 얻을 수 있다. ContentDescription은 접근성 서비스를 위해 해당 뷰에 지정한 설명 데이터이다. Fig.7.은 Android Studio의 Dump view hierarchy 기능을 이용하면 보안키패드 "6" 버튼의 content-desc 속성에 "숫자 육" 이라는 데이터가 들어 있는 것을 볼 수 있다. 이처럼ContentDescription 값을 이용해 보안키패드의 입력 값을 알아낼 수 있으며, 이것은 클릭 이벤트가 발생할 때마다 공격자 어플리케이션으로 전달된다. 또한 View의 Text 값은 보안키패드 이외에 계좌 번호, 인증서 정보와 같은 데이

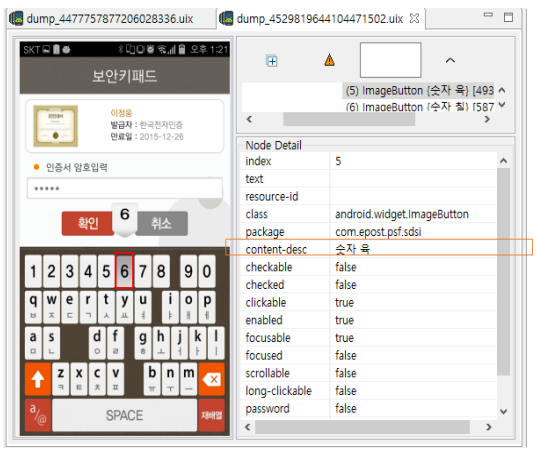

Fig. 7. Security Keypads layout properties

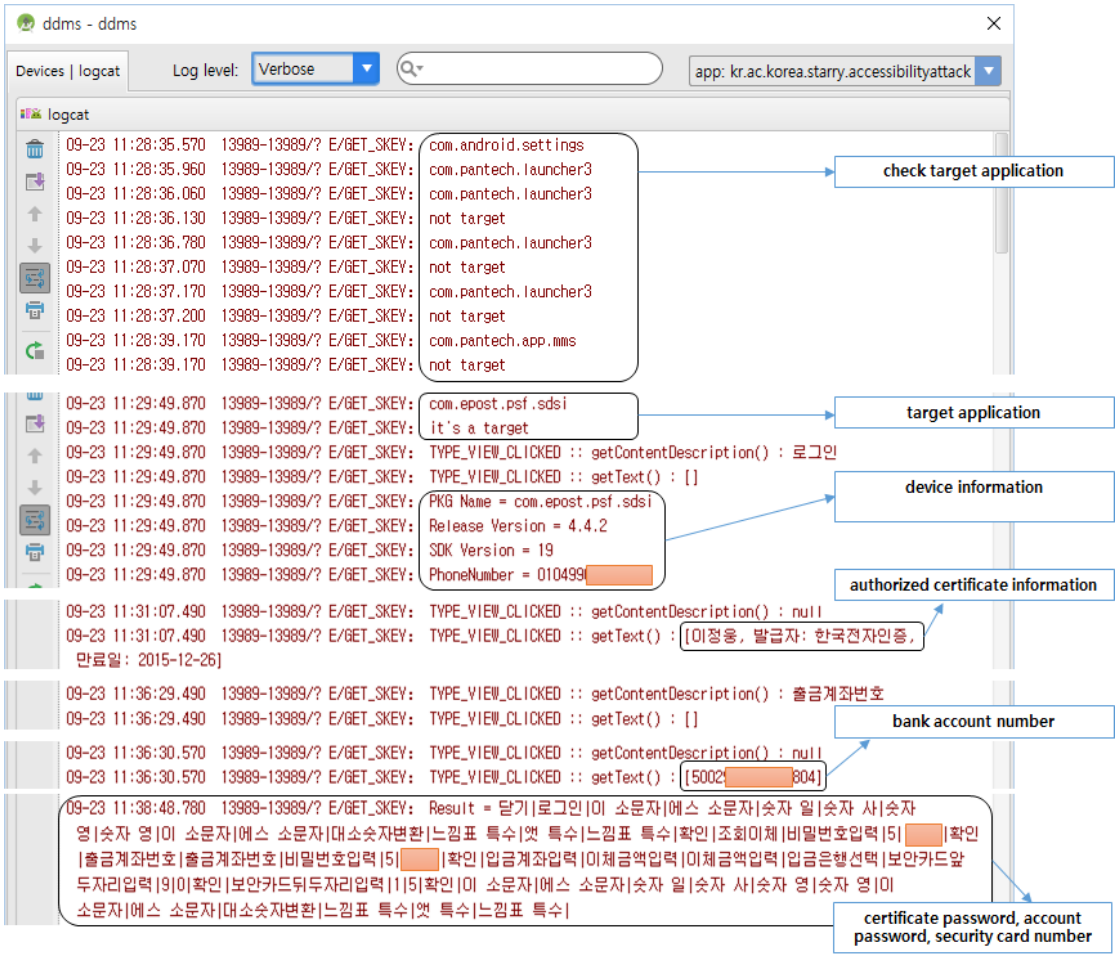

Fig. 8. Acquired information via attacker's application

터를 얻어 낼 수 있다.

4.2 테스트 및 결과

공격자 어플리케이션은 Android Studio로 제작하였으며 테스트 환경은 갤럭시 S5, 베가 아이언2 단말의 안드로이드 4.4 Kitkat OS에서 진행하였다. 제작한 공격자 어플리케이션을 테스트 대상 단말에 설치 후, Fig.9.에서처럼 사용자가 금융 어플리케이션을 실행하여 거래를 하는 과정에서 중요 정보가 노출 되는지를 확인하였다. 대상 금융 어플리케이션은 국내 은행의 모바일 뱅킹 서비스, 증권사 모바일 트레이딩 서비스 및 지불결제 서비스 등, 총 7개 어플리케이션들을 대상으로 하였다. 그리고 각 대상 어플리케이션들의 보안 키패드에서 입력 한 값이 노출되는지와 이러한 행위를 감지할 수 있는지에 대해서 Table 3.에 그 결과를 기록하였다.

Fig.8.에서 표시되고 있는 로그는 테스트 과정에서 공격자 어플리케이션이 획득한 정보를 Android Studio의 DDMS 툴을 이용해 로그로 출력한 것으로 Fig.9.의 거래과정에서 수집되는 정보들이다. 로

Table 3. Banking Application Test Results

Application	Password Disclosure	Malware Detection
Mobile Banking A	O	X
Mobile Banking B	O	X
Mobile Banking C	O	X
Mobile Banking D	O	X
Mobile Trading A	X	X
Mobile Payment A	O	X
Mobile Payment B	O	X

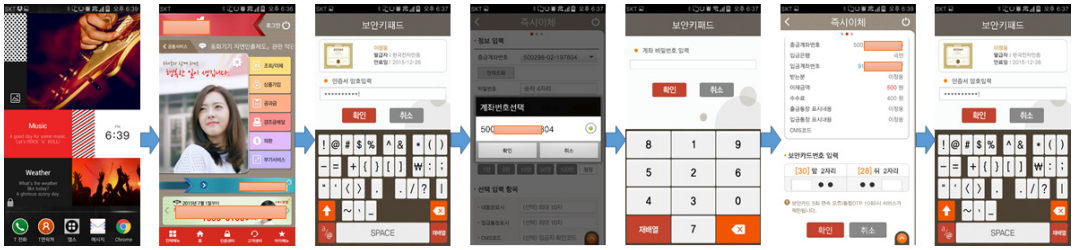

Fig. 9. Mobile Banking Application Transaction Procedure

그 정보를 확인해보면 사용자의 단말 정보, 계좌번호, 공인인증서 만료일, 공인인증서 비밀번호, 계좌 비밀번호, 보안카드 일부번호 등이 그대로 노출 되고 있음을 확인할 수 있다. 공격자 어플리케이션에서는 이렇게 노출되는 정보를 수집하여 공격자 서버로 전송하게 된다.

테스트 결과 Table 3.에서와 같이 대부분의 금융 어플리케이션에서 접근성 이용한 공격에 취약점을 보이고 있는 것을 알 수 있다. 각 어플리케이션의 보안 키패드에서 바로 암호화되어야 할 정보들은 평문 그대로 공격자 어플리케이션으로 전송이 가능했으며, 이러한 공격방법을 가진 접근성 서비스가 동작 중이라도 금융 어플리케이션과 함께 실행되는 보안 어플리케이션에서는 이를 악성행위로 감지하지 못하고 있음을 확인할 수 있었다.

만약 최소한의 권한만 사용하지 않고 더 많은 접근성 권한을 이용한다면, 터치 기반의 모바일 환경에서는 거의 모든 사용자 입력정보가 노출 될 수 있기 때문에 그 위험성이 더욱 커질 수 있다.

V. 대응방안

본 장에서는 4장에서의 공격 시나리오를 바탕으로 테스트 결과에 대해 대응방안을 알아보고자 한다.

5.1 접근성 어플리케이션에 대한 검증

Fig.10.에 작성한 순서도와 같이 금융어플리케이션이나 보안 어플리케이션에서, 현재 어떠한 접근성 서비스를 사용하고 있는지 먼저 확인하여 공격을 차단하는 것이 가능하다. Settings.Secure 클래스의 접근성 값을 확인하면 현재 사용 중인 접근성 서비스의 리스트 값을 받아올 수 있다. 만약 이 리스트에 단말에 기본 탑재되어있거나 검증된 접근성 서비스가 아닌 경우, 사용자에게 이를 알리고 접근성 서비스를 비 활성화하도록 유도한다. 확인되지 않은 접근성 서비스 설정을 해제하는 것만으로도 이러한 공격은 완벽하게 차단할 수 있다.

또한 접근성 서비스를 사용하는 어플리케이션들은 공식적으로 배포하기 전에 접근성을 악용하지 않는다는 인증을 거쳐야할 필요성이 있다. 따라서 Google 플레이 스토어나 통신사 앱 스토어 등에 어플리케이션을 등록하는 과정에서 접근성 서비스를 사용하는 어플리케이션의 경우 이에 대한 검증 절차 방안이 별도로 필요하다.

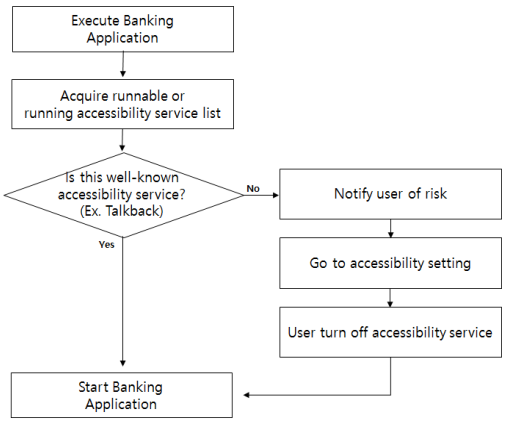

Fig. 10. Verification of accessibility service before starting the banking application

5.2 모바일 어플리케이션 접근성 지침

국가정보화 기본법 제32조 5항에는 장애인·고령자 등의 정보 접근, 이용 편의 증진을 위한 정보통신서비스 및 정보통신제품 등의 종류·지침을 정해 고시하도록 되어있다[11]. 그리고 이는 미래창조과학부 고시 제2013-107호의 모바일 애플리케이션 접근성 지침을 통해 준수하도록 되어있다[12].

모바일 애플리케이션 접근성 지침 제 6조에는 텍

스트 아닌 콘텐츠에 대해 대체 가능한 텍스트와 함께 제공하도록 준수사항으로 명시하고 있기 때문에, 안드로이드에서는 Content Description값을 보안 키패드와 같이 이미지 버튼으로 구성된 View에 대해서도 작성해 주어야 한다. 하지만 본 논문에서 살펴본 것처럼 안드로이드의 접근성을 악용하게 되면 Content Description 값에 접근할 수 있고 이를 통해 보안 키패드의 입력 값을 외부로 노출 시킬 수 있다.

따라서 모바일 애플리케이션 접근성 지침에서는 보안 키패드와 같은 중요 정보를 다루는 화면에서 접근성 기능을 위해 Content Description값을 작성하는 것이 보안 취약점이 될 수 있음을 명시할 필요가 있다. 또한 모바일 접근성 인증을 위한 심사에서 이러한 보안 취약점에 대한 대응 방안마련 여부를 심사 항목에 추가한다면, 개발사들로 하여금 기술적인 대안을 마련할 수 있도록 유도할 수 있을 것이다.

VI. 결 론

최근 핀테크 산업의 성장과 더불어 전자지갑이나 간편결제 어플리케이션 등 다양한 모바일 금융 어플리케이션들이 출시되고 있으며, 이에 따라 모바일 환경의 악성 어플리케이션은 다양한 방식으로 사용자의 정보를 위협할 것으로 예상된다. 사고 사례에 대해서 분석하고 예방을 하는 것도 필요하지만 발생 가능한 취약점에 대해서 미리 연구를 하고 그에 대한 대비를 하는 것도 중요한 일이다. 본 논문에서는 기존의 악성 어플리케이션과 다른 방식을 사용해, 금융 어플리케이션의 보안 키패드의 값을 알아낼 수 있는 공격방법을 제안하였다.

접근성 서비스를 이용한 공격방식은 사용자에게 접근성 활성화에 대한 승인을 받는다는 한계점이 존재하지만, 이를 승인할 경우 사용자의 중요한 정보가 대부분 노출 될 수 있는 치명적인 위험성을 가지고 있다. 따라서 보안키패드를 사용하는 중요한 정보를 입력받는 어플리케이션들은 이러한 위협에 대해서 대응할 수 있는 방안이 반드시 필요하다.

향후에는 모바일 금융 어플리케이션의 보안을 위협할 수 있는 다양한 공격방법과 이에 대한 대응방안에 대해서 연구할 계획이다.

References

〔1〕 “Android and iOS Squeeze the Competition”, IDC, 2015.2.24., http://www.idc.com/getdoc.jsp?containerId=prUS25450615

〔2〕 “2015 Mobile Security Threat Expectation Trend Big 4”, Ahnlab, 2015.1.6., http://asec.ahnlab.com/1018

〔3〕 Yunho Lee, “An Analysis on the Vulnerability of Secure Keypads for Mobile Device,” Journal of Korean Society for Internet Information, 14(3), pp.15-21, June. 2013

〔4〕 Android Accessibility, https://support.google.com/accessibility/android/answer/6006564?hl=ko

〔5〕 Web Standards Darum, “Android Accessibility-TalkBack” http://darum.daum.net/accessibility/tools/android

〔6〕 Android Developers, “Building Accessibility Services” https://developer.android.com/guide/topics/ui/accessibility/services.html

〔7〕 Accessibility Service, https://developer.android.com/reference/android/accessibilityservice/AccessibilityService.html

〔8〕 Accessibility Event, http://developer.android.com/reference/android/view/accessibility/AccessibilityEvent.html

〔9〕 Kakao Talk, https://play.google.com/store/apps/details?id=com.kakao.talk

〔10〕 360 Security, https://play.google.com/store/apps/details?id=com.qihoo.security

〔11〕 MSIP Framework Act on National Informatization, http://www.law.go.kr/lsInfoP.do?lsiSeq=162070&efYd=20141119#AJAX

〔12〕 MSIP Mobile Application Accessibility Guideline, http://www.law.go.kr/conAdmrulByLsPop.do?&lsiSeq=162070&joNo=0032&joBrNo=00&datClsCd=010102&dguBun=DEG&#AJAX

〈저자소개〉

이 정 웅 (Jung-woong Lee) 학생회원
2010년 2월: 전남대학교 컴퓨터공학과 학사
2015년 3월~현재: 고려대학교 정보보호대학원 금융보안학과 석사과정
〈관심분야〉 전자금융보안, 모바일보안

김 인 석 (In-seok Kim) 종신회원
1973년 2월: 홍익대학교 전자계산학과 학사
2003년 2월: 동국대학교 정보보호학과 석사
2008년 2월: 고려대학교 정보경영공학과 박사
1980년~2011년: 한국은행, 금융감독원 근무
2011년~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 전자금융보안, IT감사, 전자금융법규