Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.26, NO.4, Aug. 2016 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2016.26.4.921

클라우드 플랫폼을 이용한 악성 URL 및 수정된 APK 파일 검증 시스템 설계 및 구현*

제 설 아,† 응 웬 부 렁, 정 수 환‡
숭실대학교

Design and Implementation of Verification System for Malicious URL and Modified APK File on Cloud Platform*

Seolah Je,† Nguyen Vu Long, Souhwan Jung‡
Soongsil University

요 약

최근 국내에서 개인정보를 유출하고 금전적 손실 등의 2차 피해를 발생시켜 큰 문제로 대두되고 있는 스미싱 공격 기법은 악성 URL과 악성 어플리케이션이 주요 공격 요소인 사회 공학적 해킹 기법이다. 사용자는 스미싱 공격에 사용되는 문자메세지의 호기심을 유발시킬 수 있는 내용으로 인하여 의심없이 악성 URL에 접속하고 다운로드 되는 APK 파일을 검증 절차 없이 설치하기 때문에 쉽게 스미싱 공격에 노출되고 있다. 하지만 현재 상용되는 스미싱 방지 앱 들의 경우 시그니쳐가 생성된 뒤부터 차단이 가능한 사후처리 방식이므로 빠른 대응이 어렵다는 문제점을 지닌다. 금전적인 2차적 피해를 유발할 수 있는 스미싱의 방지 대책으로 실시간으로 검사 가능하며 다운로드 되는 APK 파일의 수정여부를 확인 할 수 있는 시스템이 필요하다. 따라서 본 논문에서는 클라우드 플랫폼을 이용한 악성 URL 및 악성 APK 파일 검증 시스템과 사용자 단말기에 설치되는 어플리케이션을 설계 및 구현하므로써, 스미싱 공격과 그에 따른 개인정보 유출 등의 2차 피해를 예방하고 한다.

ABSTRACT

Over the past few years, Smishing attacks such as malicious url and malicious application have been emerged as a major problem in South Korea since it caused big problems such as leakage of personal information and financial loss. Users are susceptible to Smishing attacks due to the fact that text message may contain curios content. Because of that reason, user could follow the url, download and install malicious APK file without any doubt or verification process. However currently Anti-Smishing App that adopted post-processing method is difficult to respond quickly. Users need a system that can determine whether the modification of the APK file and malicious url in real time because the Smishing can cause financial damage. This paper present the cloud-based system for verifying malicious url and malicious APK file in user device to prevent secondary damage such as smishing attacks and privacy information leakage.

Keywords: Smishing, cloud-based, malicious URL, modified APK file

Received(11. 30. 2015), Modified(1st: 06. 16. 2016, 2nd: 07. 14. 2016), Accepted(07. 14. 2016)

* 본 논문은 2015년도 하계 학술대회에 발표한 우수논문을 개선 및 확장한 것임. "본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음" (IITP-2016-H8501-16-1008). 또한 이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.R-20160222-002755, 맞춤형 보안서비스 제공을 위한 클라우드 기반 지능형 보안 기술 개발).

† 주저자, seolahje@ssu.ac.kr
‡ 교신저자, souhwanj@ssu.ac.kr(Corresponding author)

I. 서 론

국내의 경우 2013년 이미 스마트폰 보급률이 73%를 넘어섰으며 전 세계 인구 당 스마트폰 보급률은 2014년 24.5%로 20%인 PC 보급률을 넘어섰다[1]. 전 세계적으로 스마트폰 보급률이 증가함에 따라 스마트폰과 관련한 각종 다양한 서비스와 어플리케이션이 개발 및 보급되고 있다. 또한, BYOD(Bring Your Own Device) 환경의 보편화로 의해 본인의 스마트폰을 이용해 회사 업무를 수행할 수 있고, 더 나아가 금융 및 은행 업무가 가능해짐에 따라 다양하고 중요한 사용자의 데이터들이 저장되고 있다. 그로 인해 자연스럽게 스마트폰에 저장되는 데이터들을 타겟으로 한 다양한 공격들이 발생하고 있으며, 국내에서는 악성 URL과 사회 공학적 기법을 이용한 스미싱 공격이 증가하고 있는 추세이다[2].

국내에서 악성 URL을 포함하는 다양한 공격방법으로 인한 피해가 증가함에 따라 보안업체에서 다양한 스미싱 방지 앱이 출시되고 있지만, 이미 신고되어 분석을 마친 URL 또는 관련된 문자열의 패턴 매칭을 통해 사용자에게 알림을 주는 형태이다. 문자열 시그니쳐 기반의 스미싱 방지 알고리즘은 분석을 마쳐 시그니쳐가 형성된 스미싱 공격에는 효과적이지만 보고되지 않은 새로운 스미싱 공격 및 간단한 문자열의 추가로 인하여 시그니쳐와 매칭 되지 않는다면 빠른 대응이 어려운 단점을 지닌다[3]. 또한, 스미싱 방지 어플리케이션들은 동작이 느린 단점으로 인해 스마트폰 사용자의 사용률이 적다. 따라서 대부분의 스마트폰 사용자들은 별 다른 검증 절차 없이 해당 URL에 접속하고 더 나아가 URL을 통해 유포되는 앱 설치 파일(이하 APK 파일)의 수정 여부 및 악성 여부에 대한 확인 절차 없이 설치하게 된다. 실제로, 악성 앱의 일부는 정상적인 앱과 유사하게 제작되어 사용자의 의심을 피하고 악성행위를 수행한다. 이러한 악성 앱의 대부분의 경우 정상적인 앱에 악성행위를 추가하는 리패키징 작업을 통해 제작되기 때문에 APK(Android Package File) 파일의 수정여부가 큰 판단 근거가 된다[4].

Table 1. Smishing percentage of spam SMSs ('15 1Q)[4]

Month	Spam SMSs (included URL)	Smishing	Ratio(%)
Jan.	3,074,071	120,597	3.92
Feb.	3,068,611	570,291	18.58
Mar.	1,279,974	13,293	1.04
total	7,422,656	704,181	9.49

사용자가 전달 받은 URL에 검증 절차 없이 접속하고, APK 파일의 리패키징 여부를 확인하지 않고 APK 파일을 다운 받아 설치하는 행위들은 해커가 손쉽게 사용자 개인 정보를 습득할 수 있는 환경이 된다[4]. 또한 개인 정보 유출은 지인을 사칭하는 사회적 공학 공격 기법을 통해 2차, 3차 피해를 야기 시킬 수 있어 문제점으로 대두되고 있다[5].

악성 URL와 악성 APK 파일로 인한 개인정보 유출 및 악성 행위를 수행하는 다양한 문제를 해결하기 위해 본 연구에서는 클라우드 서버를 이용하여 악성 URL을 검증하고 APK 파일의 수정 여부를 판단하는 시스템을 구축하였다. 본 시스템은 클라우드 플랫폼을 이용한 URL 검증 시스템과 APK 파일 검증 시스템 및 사용자 단말기에 설치되는 안드로이드 어플리케이션으로 구성된다. 사용자는 어플리케이션을 통해 스미싱 등으로 전달 받은 URL을 클라우드 검증 시스템으로 전송하고 URL 일 경우 URL 검증 시스템에서 URL의 악성여부를 판단하고, APK 파일을 다운로드 할 경우 APK 파일 검증 시스템에서 APK 파일의 리패키징 및 악성 여부를 검증 후 사용자 어플리케이션으로 전달한다. 본 시스템을 통해 사용자는 검증결과를 확인하고 접속함으로써, 위협에 대한 판단이 가능하게 된다.

서론에 이어 본 논문은 2장에서는 배경 지식으로 스미싱에 대해 소개하고, 3장에서는 관련 연구로써 스미싱을 탐지 및 차단하기 위한 다양한 연구에 대해 고찰하였으며, 4장에서는 구현된 시스템의 전반전인 동작 과정과 서버 및 클라이언트의 동작 과정에 대해 설명한다. 마지막으로 5장에서는 본 논문의 결론을 제시하며 향후 연구 및 방향에 대하여 기술한다.

II. 배경 지식

2.1 스미싱의 정의

스미싱(Smishing)이란 문자메시지(SMS)와 피싱(Phishing)의 합성단어로, SMS는 메시지 내용과 URL로 구성되어 사용자의 호기심을 유발시켜 URL을 클릭하도록 유도한다. 사용자는 URL에 접속하여 해커의 의도대로 동작하는 악성 앱을 설치하게 되고

개인정보 및 금융정보 등을 탈취당하는 사이버 사기 수법 중 하나이다. 스미싱은 국내에서 2012년 최초 발견된 이후, 2013년부터 급속도로 유포되고 있으며 2015년도 1분기에만 스팸 문자 7,422,656건의 약 9%를 차지하고 있는 공격 기법이다[4].

2.2 스미싱의 공격 시나리오

스미싱의 기본적인 공격시나리오는 Fig.1.과 같이 피해자의 호기심을 유발 할 수 있는 SMS 내용을 이용해 악성 URL의 접속을 유도하는 것이다. 접속한 악성 URL은 파밍된 웹사이트로 연결되며 공격자가 의도하는 악성 행위를 수행하는 악성 앱이 자동으로 다운로드 된다. 피해자에 의해 설치된 악성 앱은 피해자의 스마트폰에서 개인정보 탈취, 공인인증서 탈취, 소액결제, 자동결제, 금액 이체, 요금 과금 등 해커가 사전에 계획한 악성 행위들을 수행한다[6].

특히, 최근에는 스미싱을 통해 유출된 개인정보를 이용한 소액 결제, 금액 이체 등의 금전적 2차 피해까지 발생하고 있어 사회적인 문제로 대두 되고 있다[7].

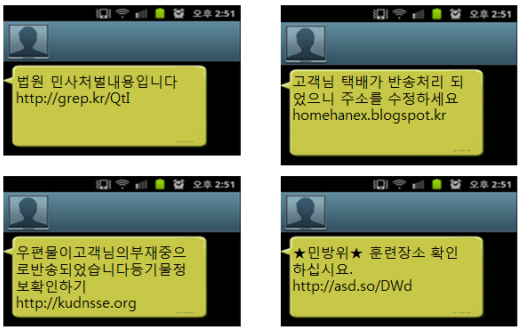


Fig. 1. Actual reported contents[8]

2.3 스미싱 방지 대책

국내 스미싱 관련 피해가 증가함에 따라 정부와 각 기관에서 스미싱 관련 대책들을 발표하고 있다.

한국인터넷진흥원에서 2015년 3월에 발표한 '스미싱 예방 및 대응 가이드'는 평소 스마트폰을 안전하게 관리하기 위한 보안 수칙과 스미싱 피해를 예방하기 위한 보안 수칙 및 스미싱 피해 시 대응 방법까지 포함하고 있다. 스미싱 피해를 방지하기 위한 수칙으로는 다음과 같다[2].

- ○ 공식 앱 마켓 이용하기
- ○ 앱 설치 시 요구 권한의 적절 여부 확인하기
- ○ 휴대폰 관리자 권한 활성화 금지
- ○ 모바일 백신 설치 및 실시간 감시 설정 등
- ○ '웹 발신 확인 서비스', '번호도용 문자 차단 서비스', '소액결재 차단 서비스' 등의 스미싱 차단 서비스 가입
- ○ 스미싱 의심 문자신고 (국번 없이 118)

금융위원회는 전기통신금융사기 범죄에 대한 관련 기관 간 협조체제와 전기통신금융사기 범죄에 대한 대책을 종합적으로 협의·조정하기 위한 협의체인 「전기통신금융사기 방지대책협의회」을 출범한 뒤 2014년 12월 전기통신금융사기방지대책을 발표하였다. 해당 방지 대책에서 3대 핵심 범죄이용수단에 대한 관리 강화, 금융회사의 피해방지 노력 확대, 민관 협력 및 교육 강화를 주요 내용으로 하고 있다[9].

III. 관련 연구

국내 스미싱 범죄가 급증하면서, 모바일 스미싱 관련 악성코드의 분류 방법과 예방·차단 방법에 대한 연구가 활발해지고 있다.

이시영 등은 사용자 기기에 수신된 문자메시지를 수집하여 데이터베이스 내에 문자메시지가 저장되었는지 검사함으로써 악성코드의 설치여부를 판단하였으며, 커널 레벨에서의 시스템 콜 후킹과 소켓 통신을 위한 버퍼의 특정 문자열 확인을 통해 서버와의 통신을 차단 차단하는 기법을 연구하였다[4].

박재경은 웹 엔진인 Webkit의 Webcore 모듈을 수정한 엔진을 사용하였으며 독립적인 실시간 악성 링크 DB을 이용해 악성코드 URL을 판단한 후 별도의 Webkit API를 통해 접근을 제어하는 기법을 제안 및 구현하였다[10].

서길원 등은 앱의 패키지 명을 이용해 앱의 업데이트를 모니터링하고 축약 URL을 긴 URL로 변환하여 사용자에게 보여주어 스미싱을 차단하는 기법을 연구하였다[11].

주춘경 등은 금융기관이 고객의 개인식별코드를 이용하여 암호화를 수행한 프로토콜을 이용하여 안전한 문자 메시지 통신을 제안하였다[12].

이지원 등은 스미싱 문자의 형태소 분석과 SVM (Support Vector machine)을 이용한 기계 학습을 결합하여 스미싱 문자를 차단하는 시스템을 연구하였다[13].

데이터베이스와 앱 패키지명 등을 이용한 탐지 기

법의 경우 지속적인 모니터링이 불가피하며 이는 단말기의 전체적인 동작 속도를 느리게 할 수 있다. 다음으로, 축약 URL을 긴 URL로 변환하여 사용자에게 알려주는 기법의 경우 긴 URL이 특정 단어를 포함하지 않는 한 그 결과를 일반 사용자가 악성행위를 예측하기에는 어려움이 있다. 더 나아가, 개인 식별 코드를 이용하여 암호화 프로토콜의 경우 실제적으로 프로토콜이 채택되지 않는 한 실제적으로 반영되기 어려운 문제점을 지닌다. 또한, 관련 연구의 대부분의 경우 모두 사용자 단말기에서 동작하여 제한된 자원으로 동작하는 안드로이드 단말기에 동작 오버헤드를 증가시킨다.

본 논문에서는 클라우드 플랫폼을 이용하여 안드로이드 단말기의 오버헤드를 줄일 수 있으며 다양한 분석 정보를 사용자에게 직접 제공하여 사용자가 모바일 악성 행위에 대해 인지할 수 있는 클라우드 기반의 악성 URL 및 수정된 APK 파일 검증 시스템을 제안하고자 한다.

IV. 제안하는 기법 및 구현

4.1 제안하는 기법

본 시스템은 악성 URL을 검증하는 URL 검증 시스템, 수정된 APK 파일을 검증하는 APK 파일 검증 시스템, 사용자 단말기에 설치되어 서버로 정보를 전 송하고 검증 결과를 사용자에게 제공하는 안드로이드 어플리케이션으로 구성된다. 본 시스템은 클라우드 플랫폼을 기반으로 구축되었으며, 악성 URL 및 수정된 APK 파일을 검증하고 사용자에게 위협에 대해 판단할 수 있는 정보를 제공하는 시스템이다. 클라우드 플랫폼을 이용해 여러 개의 VM(Virtual Machine)을
이용하여 검증 과정을 수행하므로써 단일 VM 또는 단일 서버에서 수행하였을 때보다 빠르게 동작할 수 있으며 클라우드 플랫폼 위에 동작하는 VM의 경우 사용자의 안드로이드 단말기와 동일한 안드로이드 환경이기 때문에 실제 단말에서의 동작 결과와 유사한 결과를 제공할 수 있다.

사용자는 본 시스템을 통해 스미싱 공격에 의해 특정 URL로 접속을 시도했을 때의 웹페이지 모습과 세 가지 사이드의 URL 검증 서비스의 결과를 제공받을 수 있으며, 직접 연결을 시도할지 차단할지 결정할 수 있다. 또한 APK 파일을 다운로드하는 URL 이라면 다운로드 받은 APK 파일이 구글 플레이 스토어에서 받을 수 없는 수정된 APK 파일인지 APK 파일의 수정 여부도 확인 할 수 있다. 본 분석 시스템을 통해 사용자는 악성 URL에 접속했을 때의 상황을 예측 할 수 있고, 설치하려는 APK 파일의 수정여부를 판단할 수 있다.

4.2 시스템 구성도

본 시스템은 Fig.2.와 같이 사용자의 안드로이드 앱과 클라우드 서버로 구성된 URL 검증 서버와 APK 파일 검증 서버로 구성되어 있다. 안드로이드 앱은 미리 사용자 스마트폰에 설치되어 있어야 하며, AndroidManifest.xml 파일에 Fig.3.의 line 36, line 37 코드를 추가하여 사용자가 URL을 클릭할 때 브라우저가 아닌 본 시스템의 안드로이드 앱으로 연결되도록 구성하였다.

Fig.4.는 전체적인 시스템의 동작 플로우로, 사용자가 URL에 접속하려고 할 때 연결되는 어플리케이

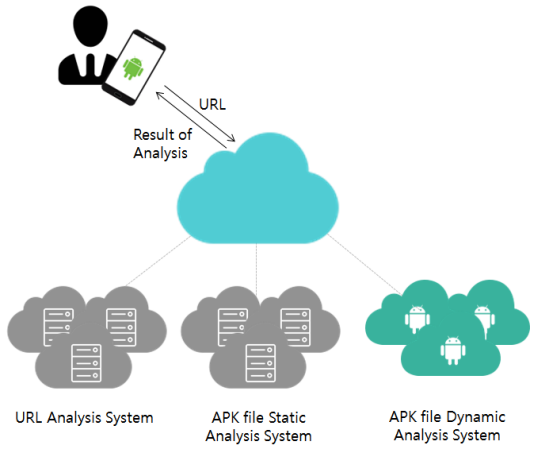


Fig. 2. System Architecture

```
25      <activity
26          android:name=".MainActivity"
27          android:label="@string/app_name" >
28          <intent-filter>
29              <action android:name="android.intent.action.MAIN" />
30              <category android:name="android.intent.category.LAUNCHER" />
31          </intent-filter>
32          <intent-filter>
33              <action android:name="android.intent.action.VIEW" />
34              <category android:name="android.intent.category.DEFAULT" />
35              <category android:name="android.intent.category.BROWSABLE" />
36              <data android:scheme="http" />
37              <data android:scheme="https" />
38          </intent-filter>
39      </activity>
```

Fig. 3. Android code for connecting App

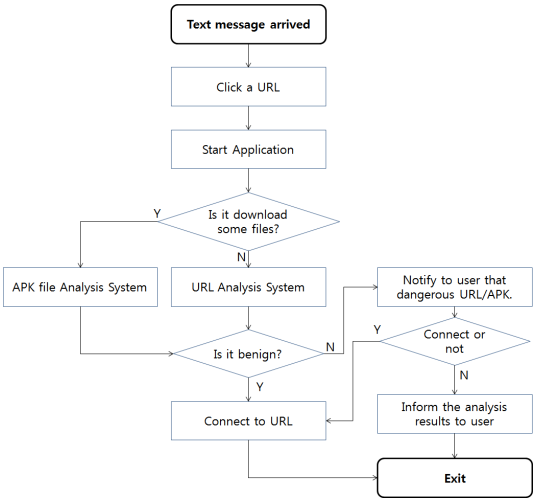


Fig. 4. Flow chart for overall system procedure

션을 해당 앱으로 선택하여 실행되어야 한다. 안드로이드 앱은 URL의 형식을 확인하여 URL인지 APK 파일 다운로드를 유도하는 웹 페이지인지 확인하여 목적에 맞는 서버로 전송한다. 서버는 URL 검증 시스템 또는 APK 파일 검증 시스템을 통해 검증을 수행한 후 안드로이드 앱에게 결과를 전송하며 안드로이드 앱은 전달받은 결과를 토대로 안전성을 측정하여 안전한 URL의 경우 연결하고, 안전하지 않은 URL의 경우 사용자에게 알려 연결여부를 사용자가 선택하도록 한다. 서버에서는 어플리케이션에 검증한 결과를 제공하여 사용자가 URL에 접속 했을 때의 상황을 확인할 수 있도록 한다.

4.3 URL 및 APK 파일 검증 시스템

4.3.1 URL 검증 시스템

URL 검증 시스템은 Fig.5와 같이 클라우드에 세팅된 4개의 안드로이드 VM이 병렬적으로 작업을 수행하여 단일 VM이나 서버에서 수행하였을 때보다 빠르게 동작하며 이는 사용자의 안드로이드 단말기와 동일한 안드로이드 환경이기 때문에 실제 안드로이드에서 확인할 수 있는 화면을 제공함으로써, 실제로 동작되는 실행결과를 미리 확인할 수 있다. 1번 VM은 사용자가 전송한 URL에 직접 접속하여 접속한 웹페이지의 화면을 캡쳐하여 사용자에게 전송하기 때문에 사용자는 해당 URL이 어떤 웹페이지와 매칭 되는지 즉각 가시적인 확인이 가능하다.

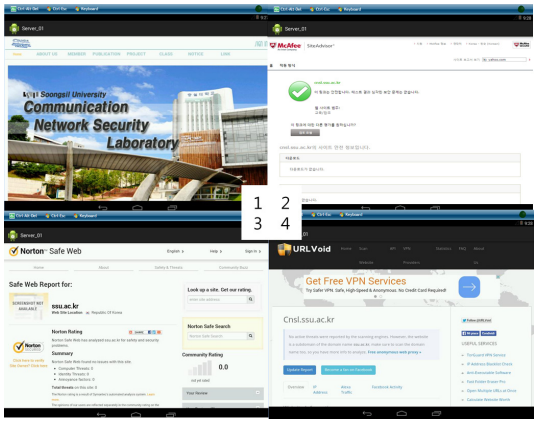


Fig. 5. URL checking system

URL의 악성여부를 확인하기 위해 2~4번 안드로이드 VM은 McAfee사의 SiteAdvisor, Norton사의 Safe Web, NoVirusThanks사의 URLVoid를 이용하여 URL의 악성여부를 판별한다. 3개 서비스 모두 각 기업의 자체적인 서버와 검증 시스템을 구축하고 있으며 사용자가 입력하는 웹사이트를 분석해 사용자 및 사용자 시스템에 미치는 영향을 파악하여 안전도를 제공하는 평판 서비스로, 국내외 악성 도메인을 판별하는데 사용된다.

4.3.2 APK 파일 검증 시스템

APK 파일 크롤러는 유포되는 URL을 통해 다운로드 되는 APK 파일 검증을 위해 구축된 데이터베이스이다. 클라우드 내 구축된 수 개의 크롤러는 구글플레이스토어에서 많이 다운로드 되거나 해커의 앱 리패키징 등의 공격에 많이 사용되는 은행, 경찰청, 공공기관 등의 앱을 위주로 데이터베이스를 구축한다. 또한 블랙마켓이라고 불리는 대표적인 3개의 마켓에서도 APK 파일을 다운로드 하여 데이터베이스를 구축한다.

본 시스템의 앱으로부터 전달 받은 URL을 통해 다운로드한 APK 파일은 안드로이드 정적 분석 무료 툴인 APKTOOL을 이용하여 정적 분석을 수행한다. APKTOOL은 안드로이드 앱 파일인 APK 파일을 디컴파일 할 수 있는 공개된 리버스 엔지니어링 툴로, AndroidManifest.xml 파일을 비롯해 APK 파일로부터 원본에 가까운 형태로 소스 코드를 디코딩할 수 있다. 그리고 APK를 디코딩하여 추출한 리소스에 수정을 가한 후 다시 빌드하여 APK 파일로 패키징하는 것이 가능하다.

정적분석을 통해 추출한 패키지 명 · 버전 값 · 앱의 서명 값을 이용해 본 시스템의 APK 파일 크롤러 데이터베이스에서 동일한 파일을 검색한다. 두 파일을 각각 HASH 함수를 통해 파일 무결성을 확인 및 비교하고 정적 분석을 통해 추출한 값을 비교 분석해 최종적인 APK 파일의 수정 여부를 판단하여 사용자에게 전달한다.

4.4 URL 및 APK 파일 검증 어플리케이션

URL 및 APK 파일 검증 앱은 사용자가 클릭한 URL을 클라우드 서버에게 전송하고 서버에게 받은 검증 결과를 토대로 최종적인 안전성을 판단하여 Fig.6.과 Fig.7.와 같은 간단한 형식으로 사용자에게 알린다.

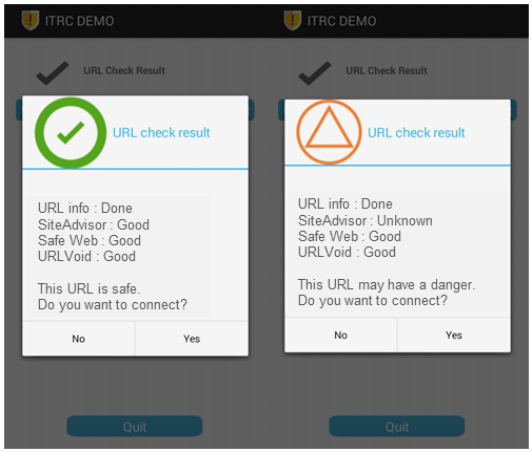


Fig. 6. URL validation results screen in App

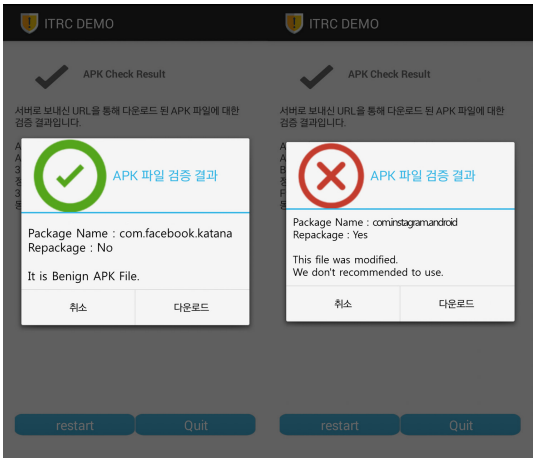


Fig. 7. APK file validation results screen in App

바로 해당 URL로 접속하지 않는다면, 보다 상세한 내용을 확인 할 수 있는 Fig.8.과 Fig.9.와 같은 상세 검증 결과 화면으로 연결된다. URL 검증 결과의 경우, 서버에서 접속한 4개의 웹페이지를 캡쳐한 이미지를 확인 할 수 있어 URL과 연결된 웹페이지, 3개의 URL 검증 서비스에 대한 결과를 확인 할 수 있다. APK 파일 검증 결과의 경우, 앱의 패키지 명과 각각 APK 파일의 해시 값을 제공하여 수정여부 결과를 제공한다. 사용자는 이러한 자세한 결과를 토대로 URL 접속 및 APK 파일 설치의 안전성을 인지할 수 있고, 사용자로 하여금 보다 안전한 스마트폰 사용을 도모할 수 있다.

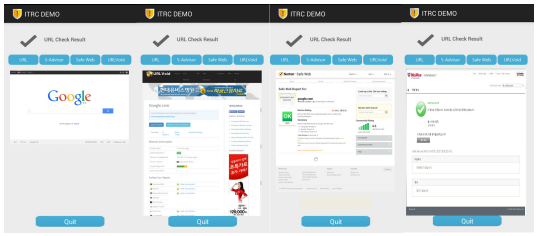


Fig. 8. URL validation detailed results screen in App

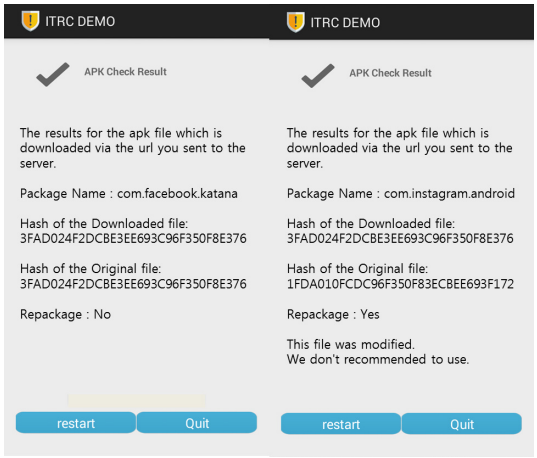


Fig. 9. APK file validation detailed results screen in App

V. 결 론

본 논문에서는 스미싱의 피해 규모 및 공격 시나리오와 특징에 대해 분석하고, 주요 공격 요소인 URL

과 수정된 APK의 악성 여부를 판단하는 검증 시스템을 구현하였다. 이러한 시스템을 통해 사용자는 악성 URL에 접속했을 때의 상황을 예측 할 수 있고, 설치하려는 APK 파일의 수정여부를 판단 할 수 있다. 만일 사용자가 스미싱 공격에 의해 특정 URL로 접속을 시도할 때 본 시스템의 어플리케이션으로 연결되고, 악성 URL일 경우 사용자에게 해당 웹페이지로 접속했을 때의 웹페이지 모습과 각 URL 검증 서비스 3곳의 결과를 제공한다. 사용자는 본 시스템의 앱이 제공하는 이미지들을 근거로 직접 연결을 시도할지 차단할지 결정할 수 있다. 또는 해당 웹페이지가 APK 파일을 다운로드 한다면 사용자는 클라우드 APK 파일 검증 시스템을 통해 분석 된 결과를 받을 수 있다. 다운로드 받은 APK 파일이 구글 플레이 스토어에서 받을 수 있는 정식 앱이 아닌 리패키징 된 APK 파일이라면 수정된 앱임을 확인 할 수 있고, 구글 플레이 스토어에서 찾을 수 없는 앱을 경우 악성 앱으로 사용자에게 전달되므로 사용자는 전달받은 URL로부터 배포되는 APK 파일에 대한 정보를 확인 할 수 있다. 위와 같은 시나리오를 통해 본 시스템을 통해 사용자는 검증결과를 확인하고 접속함으로써, 위협에 대한 판단이 가능하게 된다.

본 시스템은 안드로이드 프로그래밍의 intent filter 기능을 이용해 다양한 경로에서의 URL 접근 행위를 탐지하여 분석할 수 있기 때문에 별도의 커널 변경이 수행되지 않는다. 또한, 사용자 단말기에 설치된 앱과 클라우드 기반의 악성 행위 분석 시스템으로 구성된 본 시스템은 실제 분석 동작은 서버에서 수행하기 때문에 단말기의 오버헤드가 타 시스템에 비해 적으며 사용자가 사용함에 느낄 수 있는 오버헤드를 줄일 수 있다. 또한, URL 검사의 실제 접속 화면 및 여러 사이트의 평판 분석 결과 내용과 APK 파일 검사의 정·동적 분석 결과를 통해 타 시스템과 비교하여 각 객체의 안전도를 사용자가 직접적으로 인지 할 수 있어 잠재적인 악성 행위로부터의 면역력을 높일 수 있을 것으로 기대한다.

스마트폰 단말기가 대중화되고 성능이 고도화됨에 따라 스마트폰 사용자를 대상으로 스미싱 공격은 점차 다양화 및 지능화되고 있으며 피해 사례도 증가하고 있다. 이러한 피해를 막기 위해서는 본 연구에서 구현한 검증 시스템과 같은 다양한 시스템이 구축되어야 하며, 다양화 되고 지능화되는 악성 URL와 APK 파일들에 대한 체계화된 연구가 필요하다. 검증시스템을 우회하려는 다양한 기법에 대해서도 발 빠른 연구와 그에 대한 방어체제 구축에도 힘써야 할 것이다.

References

[1] "Mobile trends in 2015," KT Economic Management Institute, Jan. 2015.

[2] "Smshing Prevention and Response Guide," KISA, Mar. 2015.

[3] Si-young Lee, Hee-soo Kang, and Jong-sub Moon, "A study on smishing block of android platform environment," Journal of The Korea Institute of Information Security & Cryptology, vol.24, Oct. 2014.

[4] "Smshing Analysis of the First Quarter of 2015," KISA, April 2015.

[5] Joonhyouk Jang, Seunghwan Han, Yookun Cho, U Jin Choe, and Jiman Hong, "Survey of security threats and contermeasures on android environment," Journal of Security Engineering, 11(1), pp. 1-12, Jan. 2014.

[6] Youngho Jeong, Kukheon Lee, and Sangjin Lee, "Designing SMS phishing profiling model,"Journal of the Korea Institute of Information Security and Cryptology, 25(2), pp. 293-302, April 2015.

[7] H.S. Moon, B.H. Jung, Y.S. Jeon, and J.N. kim, "A Survey of Mobile Malware Detection Techniques," vol.28, ETRI, June 2013.

[8] http://spam.kisa.or.kr/kor/smishing/sm ishingWay.jsp, visited July 2015.

[9] "Telecommunications Fraud Prevention Plan for Financial Consumer Protection," Telecommunications fraud prevention Council, Dec. 2014.

[10] Jae-Kyung Park, "A study of realtime malware URL detection & prevention in mobile environment," Journal of the Korea Society of Computer and Information, 20(6), pp. 37-42. June 2015.

[11] Gil-won Seo and Il-Young Moon, "A study

of technical countermeasure system for the smishing detection and prevention based on the android platform", Journal of Advanced Navigation Technology, 18(6),pp. 569-575, Dec. 2014.

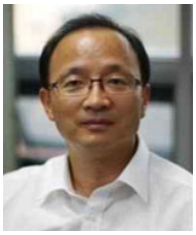
〔12〕 Choon Kyong Joo and Ji Won Yoon, "Discrimination of spam and prevention of smishing by sending personally identified SMS(For financial sector)," Journal of The Korea Institute of Information Security & Cryptology, 24(4), pp.645-653, Aug. 2014.

〔13〕 Ji-Won Lee, Dong-Hoon Lee, and In-Suk Kim, "Method of detecting smiShing using SVM," Journal of Security Engineering, 10(6), pp.655-668, Dec. 2013.

〈저자소개〉

제 설 아 (Seolah Je) 정회원
2015년 2월: 숭실대학교 정보통신전자공학부 졸업
2015년 3월~현재: 숭실대학교 정보통신학과 석사과정
〈관심분야〉 정보보호, 모바일 보안, 클라우드 보안

응웬부렁 (Long Nguyen-Vu) 학생회원
2012년 9월: Vietnam National University of Information Technology
2016년 2월: 숭실대학교 정보통신공학과 석사
2016년 3월~현재: 숭실대학교 정보통신학과 박사과정
〈관심분야〉 클라우드 보안, 모바일 보안, 네트워크 보안



정 수 환 (Souhwan Jung) 종신회원
1985년 2월: 서울대학교 전자공학과 졸업
1987년 2월: 서울대학교 전자공학과 석사
1996년 6월: University of Washington 박사
1988년~1991년: 한국통신 전임 연구원
1997년~현재: 숭실대학교 전자정보공학부 교수
〈관심분야〉 클라우드 보안, 모바일 보안, 네트워크 보안