안드로이드 어플리케이션의 진본 검증 방안

한규천
고려대학교 컴퓨터정보통신대학원 디지털정보 · 미디어공학과
e-mail : quefsa@ korea.ac.kr

Validation Plan of Android applications

Kyu-cheon Han
Korea University

요 약 스마트폰이 출시된 이후 지금까지 개발된 안드로이드 앱은 초기 앱 자체의 문제가 있었으나 스마트폰 시장이 점차 확대 되면서 개발된 안드로이드 앱의 보안 취약점과 악성코드가 삽입된 어플리케이션의 .apk 파일 배포로 인해 무결성이 지켜지지 못하고 보안 문제가 끊임없이 발생하며 안드로이드폰에 설치된 앱의 위• 변조로 악성코드가 삽인된 APK 파일을 이용해 설치하였을 경우 소스코드의 부정 사용과 개인정보유출 등 2 차로 금융사기 유도, 소액결재 등 사회적인 문제가 되고 있다. 본 논문은 위• 변조된 안드로이드 어플리케이션의 진본 여부를 확인하 위해 방안을 제안하며 악의적인 목적으로 만들어진 위• 변조된 안드로이드 어플리케이션의 apk 파일을 이용한 설치로 부정 사용되는 안드로이드 어플리케이션의 진본 설치 여부를 진단할 수 있는 방안을 제안하고자 한다.

1. 서론

스마트폰이 보급된 지 불과 4 년만에 국내 82%가 스마트폰을 사용하고 있다. (*2014 년 9 월 기준 미래창조과학부*) 스마트폰 사용자의 증가 및 다양한 앱의 개발은 과거 PC 사용자에게 발생했던 악성소프트웨어의 공격이 이제는 스마트폰의 보급률이 증가와 함께 스마튼폰으로 전이 되어 악성앱의 위협에서 벗어나기 힘들게 되었다. 특히 상대적으로 apk 파일의 소스코드 변환이 쉬워 안드로이드앱은 스미싱 방법을 이용한 악성 소프트웨어 배포로 URL 정보를 선택하면 사용자의 휴대폰 정보, 문자, 전화번호 등 정보 유출로 이어졌으며 이로 인한 2 차 피해도 발생하여 사회적인 문제가 되고 있어 사용자의 주의가 요구된다.

또한 개인 사이트를 통한 변조된 apk 파일 형태의 앱으로 배포되어 사용자의 안드로이드폰에 설치되어 소액결제 유도하는 등 악의적인 목적으로 사용된다.

위.• 변조된 안드로이드폰 앱의 사용자는 위• 변조된 사실을 모른체 설치하는 경우가 많이 끊임없이 문제시 되고 있다. 본 논문은 앱의 진본 여부를 확인할 수 있는 방안으로 구글 플레이 스토어에서 제공하는 어플리케이션과 사용 중인 안드로이드폰 앱이 동일한지 여부를 판단할 수 있는 안드로이드앱 진본 검증 방안을 제안하고자 한다.

2. 관련 연구

2-1. 악성코드 동향

과거에는 보이스피싱을 통한 다양한 금융사기가 주를 이루었으나 정부의 끊임없는 노력으로 현재는 그 수가 많이 줄어든 반면 스마트폰의 대중화와 함께 생겨난 파밍 스미싱 메모리해킹 중 스미싱은 2013 년 신종범죄로 발생하여 2 년간 피해액이 약 30 억원으로 급증하였고 2014 년 스미싱에 대한 정부의 강력한 대응으로 2014 년 스미싱을 이용한 악성 앱의 배포가 다소 감소하였으나 2015 년에는 꾸준이 증가하는 추세이다.

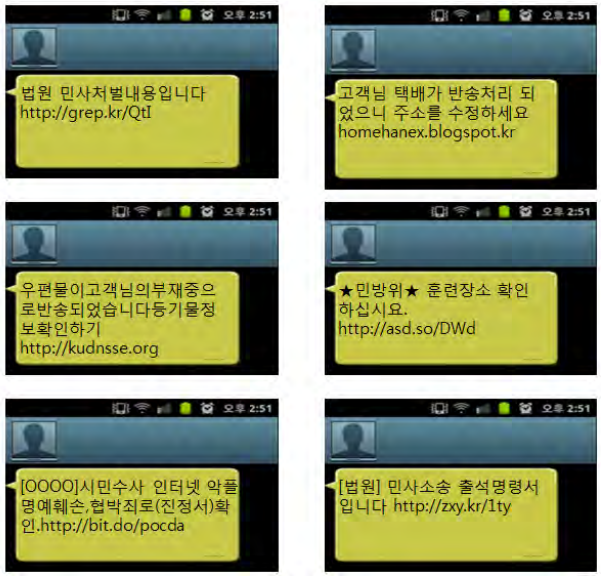
(단위:백만원/건)

	보이스피싱		파밍		스미싱		메모리해킹		총계	
	건수	피해액	건수	피해액	건수	피해액	건수	피해액	건수	피해액
2009	6,720	62,100							6,720	62,100
2010	5,455	55,400							5,455	55,400
2011	8,244	101,900							8,244	101,900
2012	5,709	59,500							5,709	59,500
2013	4,749	55,300	3,218	16,424	76,356	4,807	463	2,762	84,786	79,293
2014.6	2,851	36,900	1,628	6,842	4,459	276	97	522	9,035	44,540
합계	33,728	371,100	4,846	23,266	80,815	5,083	560	3,284	119,949	402,733

그림 1 <출처> 경찰청(스미싱은 전화결제산업협회)

2-2. 스미싱 동향

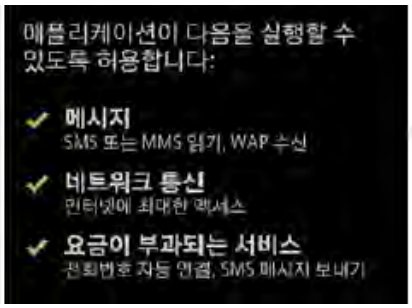
변화하는 스미싱 문자메세지 유형은 최초 모바일청첩장으로 시작하여 최근에는 관계기관 사칭, 정치적. 사회적 이슈 등의 내용을 발송하여 *사람들의 심리를 이용하는 사회공학적 기법을 사용*[1]하여 URL 을 포함한 문자메세지 발송을 통해 설치를 유도하여 피해를 주는 방법으로 기관을 사칭하여 이루어지고 있다.



<그림 2> 출처: 불법스팸대응센터

문자를 통해 악위적인 목적을 가지고 만들어진 위• 변조된 안드로이드앱을 불특정 다수에게 전송하여 설치를 유도하는 방식으로 최근 기사를 보면 “연발정산 환급금 신청”, “연하장”, “쿠폰이벤트”, “택배조회” 등 다양한 방식으로 위• 변조된 앱의 설치를 유도하고 있다.

악성앱은 설치 시 권한 확인이 필요하지만 사용자의 대부분은 이를 확인하지 않은 상태에서 앱을 설치하는 경우가 대부분이어서 피해를 보게 된다.



<그림 3> 앱 설치시 권한

스미싱의 피해단계는 특정문자에 악성 apk 의 링크를 전송하여 발송 설치를 유도하며 악성앱이 설치된 안드로이드폰의 개인정보를 통해 소액결재를 하고 대금결제는 악성앱이 설치된 안드로이드폰 사용자가 지불하게 되는 방식으로 진행된다.[2]

2-3. 악성코드 제작과정

앱의 제작과적은 소스코드를 컴파일하여 실행파일을 생성한 뒤 이를 배포용 패키지로 제작하여 코드사인[3]을 거치면 마켓에 등록 됩니다.

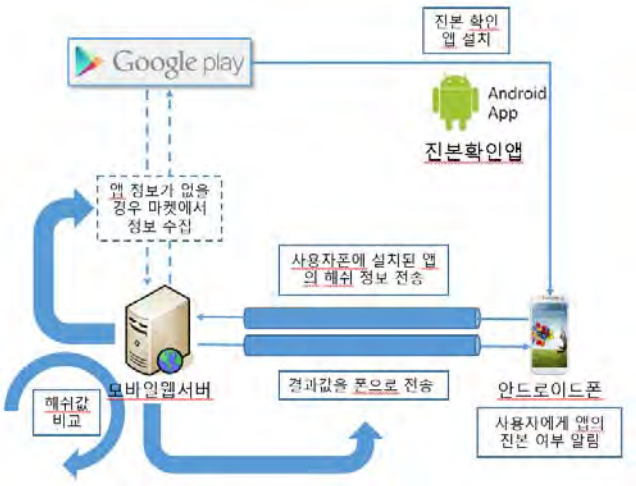
<그림 4> 안드로이드 앱 제작 과정

이러한 앱을 공격자는 다음과 같은 과정을 거쳐 악성코드를 삽입하게 됩니다. Apk 파일을 Unpackaging 하여 바이너리를 생성하고 이를 Decompile 하여 소스코드로 변환합니다. 이때 apktool[4] 같은 툴을 이용하여 decompile 을 하고 dex2jar[5]를 이용하여 원래 소스와 거의 동일하게 소스를 복원하여 소스를 수정한 후 다시 앱의 제작 과정을 반복하여 진행하면 수정된 앱을 마켓에 등록 할 수 있습니다.

<그림 5>안드로이드앱 repackaging 과정

3. 안트로이드 스마트폰 어플리케이션 진본 확인 방안

안드로이드앱의 진본방안은 안드로이드앱 apk[6] 설치파일로부터 구한 해쉬값의 수집을 통해 이루어진다. 진본앱의 해쉬값은 진본 확인 서버에 전송되어 앱의 해쉬값을 비교하게 되며 진본 확인서버에 앱의 해쉬 정보가 없을 경우 구글 마켓을 통해 해쉬값을 수집하여 서버에 기록해 두고 이를 비교 하는 방법을 이용하여 진본 여부를 진단하여 결과 값을 사용자에게 보내준다.



(그림 2) 안드로이드앱 진본확인 방안

사용자는 진본확인 앱을 통해 앱의 버전과 해쉬값이 상이할 경우 패치가 필요하다는 메시지를 보여주고 앱의 버전이 같으나 해쉬값이 다른 경우 진본이 아니라는 결과 값을 사용자에게 보여준다.

4. 주요기술

4-1. 해쉬값 비교

구글플레이를 통한 안드로이드앱의 정식 등록 정보 확인 하는 방법으로 서명검증을 사용하며 검증서버는 구글플레이에서 앱의 해쉬값과 용량 버전 등의 정보를 수집하기 위해 android-market-api[7]를 이용하여 apk 파일을 검증서버서 다운로드하여 apk 파일을 MessageDigest.getInstance()함수를 이용하여 해쉬값을 구하고 android-market-api 의 search query 를 이용하여 앱의 패키지네임정보를 전송하여 package 정보를 request 하여 정보를 받아서 database 에 저장한 뒤 사용자의 폰에 설치된 앱들의 경로를 얻어서 apk 파일의 용량과 META-INF 파일의 버전정보 그리고 PackageManager 를 이용하여 apk 파일로 추출하여 해쉬값 구해 서버로 전송하여 비교하는 방법을 사용하는 방법이다.

4-2. 서명키를 이용한 검증구간 암복호화[8]

해쉬값을 이용한 방법은 이미 다양한 보안 시스템에서 사용중이지만 이러한 방법은 매우강력한 보안 방법이기도 합니다. 해쉬값은 공격자가 알수 없도록 어플의 서명키를 사용하며 같은 값을 비교하는 방법으로 앱과 검증서버간에 데이터 전송방식은 SSL 통신을 이용하여 통신구간에서 발생할 수 있는 스니핑을 이용한 정보 탈취는 없을 것이다.

4-3. 안전한 키관리 방안

Html5 local storage[9]를 이용한 secure local storage[13]의 사용으로 local storage 를 사용할 때와 성능 차이를 비교한 결과 암.복호화를 수행하기 위해 걸리는 시간의 차이는 미미하다는 결과를 확인 하였다. 이를 이용하여 검증구간에 사용하는 서명키를 안전하게 보관 하는데 도움을 줄 것이다.

5. 결 론

최근 스마트폰 사용자의 증가와 함께 시장규모가 큰 안드로이드 운영체제를 사용하는 스마트폰 앱은 해커들의 공격 대상이 되었다.

스마트폰 앱의 역공학을 통해 소스코드를 변경하는 방식으로 악성코드를 삽입하여 재패키징하는 방식으로 스마트폰 내에 개인정보 취득 및 이용으로 피해자가 발생하는 등 사회적으로 심각한 문제를 일으키고 있다.

본 논문에서는 마켓에 등록된 안드로이드 앱의 hash 값을 이용하여 사용자의 안드로이드폰에 설치된 앱들의 진본 여부를 확인하고자 하며 이를 확인 하기 위해 사용자의 스마트폰에 설치된 모든 앱의 hash table 을 구하여 진본 앱과 의 비교를 통해 위• 변조를 확인 할 수 있는 진본 검증 방안을 제안한다.

참고문헌

[1] 바이트코드분석을 통한 안드로이드 플랫폼 스미싱 방지 기법
[2] 경찰학연구소 모바일 범죄의 특성과 예방에 관한 연구
[3]http://developer.android.com/tools/publishing/app-signing.html
[4] https://code.google.com/p/apktool/
[5] https://code.google.com/p/dex2jar/
[6] apk : http://developer.android.com/sdk/index.html
[7] https://code.google.com/p/android-market-api/
[8]안전한 html5 로컬스토리지 구현에 대한 연구
[9]W3C Web Storage W3C Recommendation 30 July 2013 http://www.w3.org/TR/webstorage/#security-localStorage