한국컴퓨터정보학회 하계학술대회 논문집 제23권 제2호 (2015. 7)

앱 내부결제를 위한 인증 서버 시스템

석호준°, 황성민*, 김석규*
°안동대학교 정보통신공학과, *주식회사 니벤스컴퍼니
e-mail : rucrazy99@naver.com°, info@nibens.co.kr*, sgkion@andong.ac.kr*

An Authentication Sever System for In-App Purchase

Ho-Jun Seok°, Sung-Min Hwang*, Seog-Gyu Kim*
*°Dept. of Infomation & Communication, Andong National University, *Nibens Company Inc.

● Abstract ●

본 논문은 앱 내부결제(In-App-Purchase)에 대한 해킹 방법을 방어 할 수 있는 서버 인증 시스템 구축 및 전자 영수증의 검증 항목들을 제안하고 있으며 앱의 위변조 검사를 할 수 있는 앱 무결성에 관해서도 제안하고 있다. 또한 기존의 해킹 방식이 아닌 다른 해킹 방식이 나와 보안 시스템이 공격을 받았을 경우 사후 관리 시스템인 상품 지급 내역과 결제 내역을 대조 할 수 있는 대사 시스템을 제안하고 있다. 앱 개발에 있어 내부결제 기능적 구현이 아닌 해킹 피해를 피할 수 있는 방법을 제안함으로 결제 보안을 높이고자 한다.

키워드: 앱(App), 내부결제(In App Purchase), 인증 서버 시스템(Authentication Sever System)

I. Introduction

스마트폰의 보급에 따라 스마트폰 콘텐츠 시장 또한 성장 하였다. 스마트폰 콘텐츠의 과금 시스템이 초기에는 어플리케이션 자체를 판매하는 것에서 현재는 어플리케이션은 무료로 배포 후 어플리케이션 내에서 상품을 선택하고 구입하는 시스템으로 변화 되었다. 이러한 시스템을 인앱결제 혹은 내부결제라 한다. 하지만 내부결제 시스템의 해킹의 위험에 노출 되어 있고 실제 해킹에 관한 피해사례도 많이 나타나고 있다. 그래서 내부결제 시스템의 구현에 있어 해킹 방지 방안에 관해 논하고자 한다. 서론에는 내부결제 해킹 방식인 역공학 공격과 DNS를 조작하여 전자 허위 영수증 발행에 관해 살펴보고 본론에서는 클라이언트 프로그램 위변조를 검사하여 무결성을 확인할 수 있는 시스템, 전자 영수증 검증 방법과 상품 지급 내역과 결제 내역을 비교 할 수 있는 대사 시스템의 구현 및 설계 방법에 관해 제안 하고자 한다.

II. Preliminaries

1. 내부결제 해킹 방식

내부 결제는 클라이언트에서 결제가 진행되게 된다. 보안을 클라이언트에서 하는 것은 매우 어려운 일이다. 클라이언트는 사용자에게 제공하는 프로그램이기 때문에 해커도 쉽게 공격할 수 있는 프로그램이기도 한 것이다. 클라이언트에서만 보안하기 어렵기 때문에 내부결제를 진행 프로세스에 인증 절차를 서버에서 진행하여 결제 프로세스를 검증 하여야 된다.

1.1 클라이언트 역공학 공격

클라이언트의 소스코드 자체를 변경하는 공격 방법이다.

결제를 진행하는 분기문의 조건에 'or True'를 추가 하게 된다면 결제의 성공 여부와 상관없이 분기문의 결제 성공 쪽으로 프로세스는 진행하게 된다.

이처럼 소스코드 자체를 수정하는 것은 클라이언트 보안에 항상 문제가 되는 요소이기 때문에 패키지 소프트웨어는 불법복제를 막는 것은 쉽지 않다.

1.2 허위 전자 영수증 발행

내부결제 해킹에 가장 사용이 많이 되는 방법이다. DNS를 조작하여 결제 마켓으로 가야되는 결제 요청을 해킹 프록시 서버로 요청이 가게 되고 해킹서버에서는 정상적인 결제가 이뤄지지 않았지만 결제 승인을 보낸다.

Freedom Hacking Tool Mechanism

① 내부결제 적용 앱 검색, 루트권환 획득
② /etc/hosts 파일 변조
③ 앱 실행 및 결제 시도
④ Freedom Proxy Server로 결제 요청
⑤ FreeCard-xxx 생성 및 지불 요청
⑥ 마켓 결제 승인
⑦ Freedom Proxy Server에서 마켓 응답 처리 후 앱에 결과 전달

Freedom이란 해킹 툴을 사용하여 해킹이 가능하지만 루틴이 된 디바이스에서만 사용할 수 있었다. 하지만 최근 루틴 없이도 내부결제를 무력화 할 수 있는 방법이 소개가 되었는데 공유기의 DNS를 조작하는 방법이다. 루틴으로 관리자 권한을 얻지 않고 시스템의 조작 없이 공유기의 DNS만을 조작하는 방식이라 클라이언트만으로 보안이 힘들어 진 것이다.

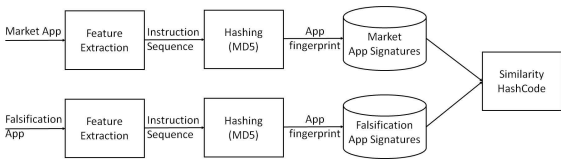
III. The Proposed Scheme

1. 클라이언트 무결성 검사

어플리케이션 위변조 검증 방법은 어플리케이션 실행 시점에 어플리케이션의 위변조 여부를 탐지하는 방법이다. 어플리케이션에서 실행 시점에 어플리케이션 실행 파일인 DEX,SO,DLL파일등을 추출하여 바이너리/바이트 코드를 해쉬값을 생성을 한다. 생성한 해쉬값을 서버에서 알려준 공개키로 암호화를 하고 서버에 전송을 하게 된다. 서버에서는 전송 받은 암호화된 해쉬값을 복호화하여 서버에서 보관한 올바른 해쉬값과 비교하여 위변조 여부를 판단하게 된다. 일치 한다면 어플리케이션이 위변조가 되지 않았다고 판단하고 불일치 한다면 어플리케이션이 위변조가 되었다고 판단하여 원격으로 어플리케이션 사용을 차단하게 된다. 제안하는 시스템에서는 해쉬 함수를 MD5로 생성을 하게된다. MD5가 안정성이 떨어져 SHA 해쉬 알고리즘을 권장하지만 본 시스템에서는 대용량의 데이터인 실행파일의 해쉬값을 추출하여야 되기 때문에 안정성이 떨어지지만 속도에서 뛰어난 MD5를 사용하여 설계를 하였다. 앱의 실행 파일의 구성요소인 DEX파일에 class파일들로 구성되어 있는데 이것들이 앱 실행 파일들이다. 안드로이드 플랫폼에서는 /data/app, /system/app, /dalvik-cache 아래 존재하고 되는데 이 파일 접근은 owner나 system group에게만 접근이 허용된다.

Integrity Check System Process

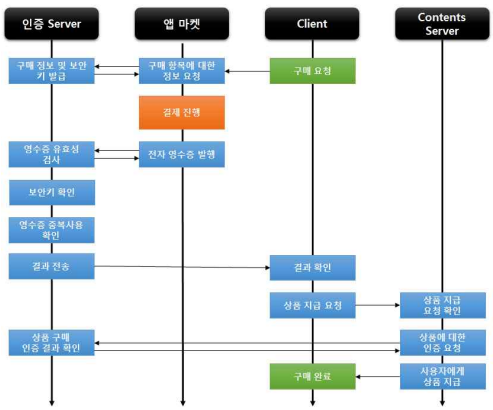
보통 실행 파일의 위변조는 설치된 파일을 변조하는 것이 아니라 APK를 풀어 DEX를 추출하고 dex에서 class파일 추출하게 된다. 그리고 class를 디컴파일 하면 java파일을 손쉽게 구할 수가 있다. 획득한 java파일을 수정 후 역순으로 파파일을 하여 class을 만들고 dex메이커로 dex로 만든후 압축을 하면 APK로 변환 되게 된다. 하지만 java파일을 수정, 추가, 삭제를 하게 되면 dex의 해쉬 값이 다르게 생성이 되게 된다. 그래서 본 시스템에서 해쉬 값으로 무결점성을 확인하는 것이다.



Hash Value Registration Process

안드로이드 플랫폼 마켓은 다양하게 있다. 그 중 몇몇 마켓에서는 업로드 한 APK파일을 마켓에서 다시 패키징을 하는 경우가 종종 있다. 처음 만들어진 APK가 변경이 되는 경우가 있는데 시스템에서 이를 위변조로 판단하고 차단하게 된다. 그래서 시스템설계에서는 APK 버전 따라 생성한 최초의 해쉬 값을 저장하게 되며 다수의 해쉬값의 어플리케이션이 원본 해쉬값으로 판단하게 된다. 만약 변조된 앱에서 해쉬값을 등록을 하더라도 비교 후 변조로 판명된 해쉬값을 가진 앱들은 모두 원격으로 사용 차단을 시키게 된다.

2. 전자 영수증 검증



Digital Receipt Verification Process

내부 결제에 보안에는 영수증 인증 서버, 클라이언트, 콘텐츠 서버, 앱 마켓 4개의 노드에서 정보가 오가며 검증 절차가 이루어진다. 인증 서버에서는 구매 정보 및 보안키를 발급하여 결제 보안에 인증 절차를 책임지는 역할을 하게 된다. 마켓에 결제 요청은 클라이언트 단에서만 보내야 되기 때문에 다중적 보안을 해야 된다. 클라이언트에서 마켓의 결제 결과를 수신한 후 마켓에서 발급한 영수증이 위변조가 행하여진 여부를 서버에서 판단을 하게 된다.

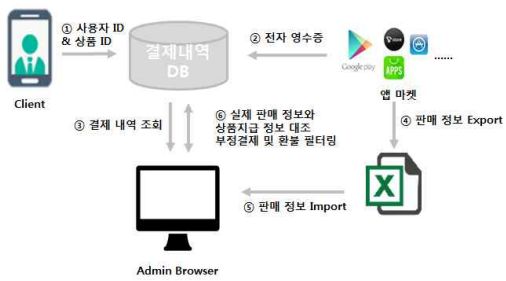
Digital Receipt Structure

Order ID Package Name Product ID Purchase State Purchase Time Purchase Token Developer Payload
Signature

전자 영수증 검증 항목

1. 과거 사용된 이력 DB에서 영수증 중복 확인
2. 패키지네임과 productID가 일치 여부로 타 앱/게임에서 발행된 영수증인지 확인
3. 영수증의 지갑번호와 판매자의 지갑번호가 일치 여부로 판매자의 앱에서 결제가 된 영수증인지 확인
4. 일회용 안전 페이로드 회신 여부 확인과 복호화 후 내용으로 결제 시작 디바이스와 일치 확인
4. 구매 완료된 시간 확인
5. 인증 서버에 보관된 공개키로 영수증을 서명하여 반환된 영수증의 서명과 대조 후 영수증 위변조 검증

3. 상품 지급 내역과 결제 내역 대사시스템



Payment List Compare System

내부결제로 판매된 상품의 종류 및 수량 등을 앱 사용자 ID와 연결하여 관리한다. 개발사에서 사용하는 사용자 ID와 영수 정보를 연결하여 원하는 정보를 쉽게 조회 가능한 도구를 제공하고자 한다. 기존 마켓에서 판매내역을 제공하지만 판매 내역과 앱에서 사용하는 사용자 ID가 함께 제공되지 않기 때문에 개발사에서 판매관리에 어려움을 겪고 있다. 결제 완료 후 결제내역 서버에 영수증의 Order ID와 앱 사용자 ID를 함께 보관한다. 사용자가 개발사로부터 환불을 요청할 때 사용자 ID로 결제내역을 쉽게 조회 하여 결제를 환불해줄 수 있게 된다.

서버사이트 인증 프로세스로 인증을 하더라도 혹시 고려되지 않은 새로운 해킹 수법이 발생하여 내부결제 보안의 결점이 생기더라도 마켓의 실제 결제 내역과 상품지급 내역을 대조하는 기능으로 부정 결제 사용자를 쉽게 색출할 수 있게 된다. 해킹 사고의 사후 관리가 가능하다.

IV. Conclusions

본 논문에서는 스마트 기기에서 앱 내부결제에 관한 인증 및 보안에 관해서 연구하였다. 클라이언트에서 결제를 요청하고 결과를 받아야 되는 앱 마켓의 내부결제 페이먼트의 특성상 보안에 취약하여 본 논문에서 제안하는 서버에서의 인증 절차 및 데이터 저장을 권장한다. 서버에서의 전자 영수증 검증과 데이터 저장을 서버에서 한다면 내부결제 보안에 한해서는 앱 무결성을 검사 하지 않아도 된다. 그리고 전자영수증 서명에 사용되는 공개키는 필수적으로 서버에서 보관하고 전자영수증의 서명을 검증하여야 된다. 클라이언트쪽에서의 보안은 취약하기 때문에 가능한 정보들은 서버에서 처리하고 보관하여야 하며, 불가피하게 클라이언트에서 처리해야 되는 경우는 서버를 통해 위변조 검사를 진행하는 것이 정보를 안전하게 보호 할 수 있다. 또한 해킹의 방법은 항상 진화하기 때문에 해킹 발생 후 관리 시스템 또한 구축하는 것이 중요하다. 해킹 방식 분석에 필요한 기록을 남기는 것 또한 좋은 방법이다.

향후 연구 과제로는 여러 개의 앱 마켓들이 내부결제 모듈이 상이해 모듈 적용의 어려움을 해결 할 수 있는 통합 내부결제 모듈에 관한 설계 및 방법에 관해서 연구하고자 한다.

References

[1] ZonD80, "Apple's in-app purchasing process circumvented by Russian hacker", http://9to5mac. com/

[2] Wonnam,Lee "A Study on the Android Vulnerability of Rooting/App Integrity Verification Module" Dec. 2014.

[3] Google, "In-appBillingSecurityandDesign," http://developer. android.com/google/play/billing/billing_best_practices.htm

[4] Google, "In-appbillingVersion3API," http://developer. android.com/google/play/billing/api.html

[5] Mulliner, Collin, William Robertson, and EnginKirda. "Virtual Swindle : an automat edattack against in-app billingon android," Proceedings of the 9th ACM symposium on Information, computer and communications security, 2014.

[6] Jinsun, Hong " Mobile Game Hacking", http://www.inven.co. kr/webzine/news/?news=128022