Journal of Korea Game Society 2015 Feb; 15(1): 77-88
http://dx.doi.org/10.7583/JKGS.2015.15.1.77

게임 콘텐츠, 산업, 마케팅

소셜 네트워크 게임(SNG) 서비스의 개인정보 유출 및 보안위협 대응방안에 관한 연구

이상원*, 김휘강*, 김은진**
고려대학교 정보보호대학원*, 경기대학교 국제산업정보학과**
{infoprotection, cenda}@korea.ac.kr, ejkim777@kgu.ac.kr

A Study on Countermeasures for Personal Data Breach and Security Threats of Social Network Game

Sang Won Lee*, Huy Kang Kim*, Eun Jin Kim**
Graduate School of Information Security, Korea University*
Department of International Industrial Information, Kyonggi University**

요 약

스마트폰 게임시장의 성장과 함께 모바일 소셜 네트워크 게임(SNG) 서비스의 이용이 크게 증가하고 있다. 이와 더불어 이들 서비스를 대상으로 한 게임 데이터 조작, 결제 부정, 계정도용, 개인정보 유출 등 보안위협이 동시에 증가하고 있다. 모바일 소셜 네트워크 게임의 보안강화를 위해 강력한 개발보안 표준이 요구 되지만 게임의 짧은 생명주기, 추가적인 개발 비용의 발생, 원활한 서비스 제공의 어려움을 이유로 이의 적용이 쉽지 않은 실정이다. 본 논문에서는 소셜 네트워크 게임의 보안 강화 방안으로 발생빈도와 위험성이 높은 공격 방법 중 하나인 메모리 변조에 대한 대응 방안을 제시하고자 한다. 또한 이 방법은 모바일 환경에 맞게 가볍고 강력한 보안을 제공할 것으로 기대 된다.

ABSTRACT

As the smart phone market is drastically expanding, there is a steady growth of recent vicious activities such as data manipulation, billing fraud, identity theft, and leakage of personal information that are security threats to Social Network Games(SNG). Due to the threats, Strong development standard is required for security enhancement of SNG. Nonetheless, short life-spans, additional expenses, and the necessities to provide a sound game service hinders developers from reaching their security goals. Therefore, this research investigates the weak points of SNG through memory manipulation experiments based on the currently provided SNG services. In addition, the research presents counter measures and security enforcements that are light in service load and simplistic which can be applied in the developing process.

Keywords : Social Network Games, Android Game Security, Memory Manipulation

Received: Nov. 13, 2014 Revised: Dec. 15, 2014
Accepted: Dec. 29, 2014
Corresponding Author: Eun Jin Kim(Kyonggi University)
E-mail: ejkim777@kgu.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

1. 서 론

최근 스마트폰의 대중화로 카카오톡, 페이스북, 트위터 등으로 대표되는 소셜 네트워크 서비스(Social Network Service)와 이를 기반으로 하는 소셜 네트워크 게임(Social Network Game) 사용이 증가하고 있다.

한국인터넷진흥원에서 국내 스마트폰 이용자를 대상으로 '2013년 스마트폰 이용 콘텐츠'를 조사한 결과 정보검색, 웹 서핑을 주로 이용하던 스마트폰 사용자들이 최근에는 소셜 네트워크 서비스, 소셜 네트워크 게임, 동영상, 음악 등을 주로 이용하는 것으로 나타났다[1].

또한 "2013년 모바일인터넷 이용 실태조사" 결과 스마트폰 이용자 중 72%는 여가생활로 모바일 게임을 즐기고 있는 것으로 나타났다[2]. 모바일 게임 이용 증가와 함께 모바일 소셜 네트워크 게임(SNG) 이용이 증가하여 for kakao, with band 등 SNS와 연계된 소셜 네트워크 게임들이 리 생활 속에 밀접하게 자리 잡아가고 있다.

소셜 네트워크 게임은 소셜 네트워크 서비스를 이용하는 사람들이 서비스 안에 연결된 사용자들 또는 지인들과 정보를 공유하며 함께 즐길 수 있다는 특징을 가진다[3].

스마트폰의 편리한 휴대성과 소셜 네트워크 게임의 직관적이고 쉬운 조작방법, 시간과 장소의 구애를 받지 않는 특징, 인맥을 통해 급격히 확산되는 바이러스 네트워킹 등으로 소셜 네트워크 게임이 빠르게 확산되었다 할 수 있다[4].

하지만 빠른 시간 안에 개발, 서비스되고 있는 많은 소셜 네트워크 게임들은 해킹 등 보안위협에 취약하게 노출되어 그에 따른 피해사례가 발생하고 있다[5]. 특히 사용자 수가 많은 안드로이드를 중심으로 해킹공격이 많아지고 있다. 안드로이드 게임의 경우 스마트폰이 갖는 취약점과 악성코드에 관대한 앱 마켓 구조, 애플리케이션 개발 및 서비스 취약점 등이 복합적으로 작용하여 주 해킹공격의 대상이 되고 있다. 이러한 공격은 메모리 변조, 결제부정, 패킷변조, 계정탈취 등의 형태로 발생하고 있으며 그 위협 또한 증가하고 있다.

그중 메모리 변조는 발생빈도와 위험성이 높은 공격 방법 중 하나로 본 논문에서는 이에 대한 대응 방안으로 메모리 데이터 암호화 방법과 추가적으로 데이터 변조여부를 검증할 수 있는 방법을 제안하고자 한다. 또한 이 방법은 안드로이드 환경에 맞게 가볍고 강력한 보안을 제공할 것으로 기대 된다.

본 논문의 2장에서는 연구의 배경을 살펴보고 3장에서 소셜 네트워크 게임의 보안동향과 공격동향을 분석하여 4장에서는 분석결과를 바탕으로 가장 많이 발생하고 있는 메모리 변조에 대해 실험을 진행, 이에 대한 대응방안을 제시하고 5장에서는 결론과 향후 연구에 대해 논의하고자 한다.

2. 연구 배경

최근 게임 해킹 공유사이트에서 '프리덤'이라는 해킹 툴을 이용해 가짜 신용카드를 생성, 구글 인앱 결제를 시도하는 안드로이드 게임의 피해사례가 종종 발생하고 있다[6].

이는 안드로이드 소셜 네트워크 게임에서도 해킹을 통해 금전적인 피해를 줄 수 있는 보안위협이 존재함을 보여준 사례라 할 수 있다.

이러한 관점에서 다른 모바일 서비스와 마찬가지로 안드로이드 소셜 네트워크 게임 역시 많은 보안위협과 간과할 수 없는 피해가 예상되며 게임 사용자 및 게임 개발사 측면에서 금전적 피해를 포함한 다양한 피해가 발생할 수 있음을 보여준다. 안드로이드 소셜 네트워크 게임 해킹에 따른 예상 피해는 아래의 [Table 1]과 같이 정리될 수 있다.

[Table 1] Expected damage

Division	Expected damage
Game User	· Carrier charges increase · Physical Crash · Mental damage caused by unfair
Game Developer	· Loyalty decrease · Corporate image tarnished · Server management costs · Costly of Service and marketing

이러한 피해가 예상됨에도 불구하고 소셜 네트워크 게임 보안위협은 스마트폰의 보안위협이나 소셜 네트워크 서비스, 모바일 금융 서비스, 온라인 게임 서비스의 보안위협에 비해 활발히 논의되고 있지 않은 실정이다. 또한 소셜 네트워크 게임은 기존의 스마트폰 자체가 가지는 OS 취약점과 무선네트워크 사용의 취약점, SNS 플랫폼과 소셜 네트워크 게임이 갖는 개발과 서비스 상의 취약점은 물론 소셜 네트워크 게임의 특징인 짧은 생명주기와 보안강화 시 추가비용 발생, 원활한 서비스의 어려움 등이 복합적으로 작용하여 보안 위협에 대한 적극적 대응이 쉽지 않은 상황이다.

온라인 게임의 경우 사용자들에게 키보드 보안 프로그램, 무료백신, 보안패치 서비스, 모바일 OTP 등 온라인게임에 필요한 보안솔루션들을 사용자들에게 무상제공해 주는 형태로 사용자의 보안성을 높이고 있으나, 안드로이드 게임의 경우 이러한 대응이 현실적으로 어렵다.

또한 온라인 게임의 경우에는 고성능 하드웨어로 보안솔루션이 구동되어도 성능상 제약을 거의 받지 않고, 사용자들이 해당 보안솔루션에 대한 이해도가 높은 편이어서 저항감이 상대적으로 덜할 수 있지만 안드로이드 게임의 경우 주 장르가 캐쥬얼 게임으로 짧은 시간에 가벼운 게임을 즐기는 것이 사용자들의 이용목적이기 때문에, 모바일 게임 플레이를 위해 보안 프로그램을 다수 설치하거나 구동하는 것 자체에 거부감을 느끼는 경우가 있다.

더불어 스마트폰은 일반 PC에 비해 하드웨어 성능상 제약이 있어 앞서 설명한 강력한 보안기능을 적용 할 경우 CPU의 발열, 배터리의 빠른 소진, 통신비용 증가 등의 문제가 발생하는 등 많은 제약요소가 따른다.

본 논문은 위와 같이 상대적으로 보안 위협에 취약하게 노출되고 있는 모바일 게임, 특히 소셜 네트워크 게임의 보안 강화 방안에 대해 연구를 진행하였다. 또한 모바일 게임이라는 특성이 고려된 가볍고 강력한 보안기능을 제공하는 대응 방안을 제시하고자 한다.

3. 본 론

3.1 소셜 네트워크 게임 보안동향

최근 소셜 네트워크 게임 해킹의 주요 공격방법은 메모리 해킹으로 볼 수 있다[7]. 이는 소셜 네트워크 게임 시장이 커지면서 해킹을 통해 아이템 부정획득 및 금전적 이득을 목적으로 하는 해킹이 꾸준하게 증가하고 있음을 반증한다 할 수 있다.

메모리 해킹의 경우 게임서버에 심각한 부하와 금전적 피해를 주고 무엇보다도 불공정한 게임 진행으로 게임의 불균형을 초래하는 등 게임 사용자와 게임 개발사에 많은 피해를 줄 수가 있다. 이러한 메모리 해킹은 기술적으로도 해킹수법이 고도화 되고 해킹사례 또한 향후 꾸준하게 증가 할 것으로 예상되며 안드로이드를 기반으로 하는 일반 애플리케이션에도 확대될 것이 우려되고 있다.

이에 본 논문에서는 소셜 네트워크 게임 보안위협 중 메모리 해킹에 대한 보안 대응 방안의 중요성을 인식하고 이를 도출하기 위해 메모리 해킹의 보안동향과 문제점을 다음과 같이 살펴보았다.

메모리 해킹의 공격유형을 살펴보면 메모리 해

킹의 경우 애플리케이션 단에서 발생하며 특정 애플리케이션이 사용하는 메모리를 모니터링 하여 특정 문자열 및 수치 변화를 추적하고 해당 데이터가 저장되는 주소를 알아내어 필요할 때 그 값을 바꿔치기 하는 공격기법을 사용한다.

주로 게임 클라이언트 프로세스의 메모리를 조작하는 해킹방법 이다. 메모리 변조는 크게 게임 데이터나 특정 수치의 값을 변조하는 데이터 영역과, 게임의 내부 구현루틴 즉, 코드 영역 부분에 대한 해킹으로 나눌 수 있다.

데이터 조작의 경우 메모리에 적용되어 있는 레벨, 체력, 게임 아이템, 경험치, 게임머니 등 모든 수치에 대해 메모리에 로드 된 특정 값의 위치만 알아내면, 그 부분을 해커가 원하는 값으로 바꿔 메모리 변조가 가능해지게 된다.

물론 대부분 소셜 네트워크 게임에서 클라이언트에 저장되는 메모리 값은 서버에서 처리한 결과를 보여주는 용도로 사용하고 있다. 이런 경우는 크게 문제가 되지 않으나, 일부 게임에선 게임의 구조상 또는 서버의 오버헤드를 최소화하기 위하여 클라이언트의 데이터를 그대로 신뢰하여 게임에 적용하는 경우도 있다.

코드 영역변조는 게임 개발자가 게임의 룰로 정하여 작성한 코드를 무력화 시키거나 또는 해커가 원하는 대로 게임코드를 바꿀 수 있다. 또한 보안코드 영역을 NOP 처리하여 보안 루틴이 실행되지 않게 우회하는 일이 가능해진다. 이런 경우를 방지하기 위해 클라이언트 실행파일을 암호화(Packing) 하기도 하지만, Unpacker가 공개되어 Unpacking 후 다시 변조하는 경우도 적지 않으며 보안솔루션 자체를 우회하거나 실행되지 않게 만들어 버리기도 한다.

이를 막기 위해서는 각 툴들의 검색 루틴과 변조 루틴을 막는 것이 반드시 필요하다.

3.2 SNG 클라이언트 공격유형 분석

최근 모바일 게임 공격을 살펴보면 약 66%가 메모리 해킹으로 가장 큰 비중을 차지하고 있으며 [9] 파일 변조, 애플리케이션 및 패킷 위·변조, 캐시 정보 및 인증정보 탈취 등 다양한 해킹공격이 발생하고 있다. 최근 발생하고 있는 소셜 네트워크 게임의 공격유형은 다음과 같다.

3.2.1 메모리 해킹

앞서 설명한 바와 같이 메모리 해킹의 경우 주로 클라이언트 프로세스의 메모리를 조작하는 해킹을 가리키며 애플리케이션이 사용하는 메모리를 모니터링 하여 특정 문자열 및 수치 변화를 추적하고 해당 데이터의 주소를 알아내어 값을 변조하는 데이터 조작과 개발자가 코딩한 게임 코드를 무력화 시키거나, 자신이 원하는 대로 코드를 바꾸는 코드영역 변조 공격이 시도된다. 주로 해킹 툴 또는 리버스 엔지니어링에 의해서 공격이 이루어지고 있다. 메모리 해킹에 대해서는 중요 게임데이터에 대한 은닉, 소스코드 난독화, 패킹 등 안티 디버깅 기술을 적용하고 있다[10].

3.2.2 애플리케이션 위·변조

애플리케이션 위·변조는 디버깅 없이 바이트 코드나 소스를 바꿔 애플리케이션이 정상적으로 설치되고 실행은 되나 실제 개발자의 의도와 다르게 작동되어 비정상적인 이득을 취하는 방법이다. 보통 해커들은 게임소스 보단 dalvik 가상환경에서 애플리케이션 실행을 위해 작성된 smali파일을 변조하여 유료 버전의 검증을 우회하여 크랙 버전을 만들거나 게임소스가 컴파일된 SO 파일 혹은 swf 파일을 변조한다. 애플리케이션 위·변조는 무결성 검증을 통해 방지할 수 있으며 무결성 검증 루틴을 난독화 하여 대응하고 있다[11].

3.2.3 패킷 변조

패킷 변조는 통신을 주고받을 때 사용되는 패킷을 변조하여 거짓 정보를 게임에서 받아 실제 게

임에 적용되도록 하는 방법이다. 패킷 캡처 후 패킷의 Header, URL, Body를 복사하여 실제 패킷과 동일하게 만든 후 실제 게임을 하지 않더라도 경험치 및 게임머니 등을 획득하거나 패킷 데이터 중 특정 데이터를 변조하여 대량의 경험치 및 돈을 얻는 방식이 있다. 패킷변조 공격은 송수신 패킷에 대한 인증을 통해 대응하고 있다[12].

3.2.4 게임 계정탈취

해킹 유형 중 게임 사용자들이 직접적으로 피해를 입는 유형이며 대중적이고 일반적인 해킹으로 알려져 있다. 계정을 탈취하는 방법에는 사회공학적 기법에서부터 Key-logger 등 악성코드를 이용한 방법까지 매우 다양한 기술이 있다. 이러한 해킹은 기술적인 방법보다는 사용자의 주의가 필요한 유형이라 볼 수 있다.

초기의 안드로이드 게임해킹의 경우 Cheating을 통해 최종 점수를 조작하는 등 자기과시 정도였으나 이는 통신료를 발생시키거나 금전적인 이익을 제공해 주지 않기 때문에 공격의 가치가 적었지만, 최근 안드로이드 게임들은 초기 설치 시 비용을 지불하는 구매모델에서, 지속적인 게임 이용을 요구하고 아이템 구매를 유도하는 in-app purchases 모델이 주력으로 자리잡아감에 따라 공격의 가치가 높아졌으며 안드로이드 게임에 존재하는 취약점이 온라인을 통해 공유가 될 정도이며, 언론사에서도 안드로이드 게임에 존재하는 취약점을 적극 소개할 정도로 심각성이 증가하고 있는 상황이다[13].

4. 실험 및 실험결과

4.1 실험 개요

본 실험에서는 현재 국내에서 서비스되고 있는 안드로이드 게임 3개를 선택하여 실제 메모리 변조에 어떠한 취약점들을 보이는지를 분석하였다.

안드로이드 게임구동과 메모리 변조 실험을 위해 Blue Stacks라는 안드로이드 스마트폰 환경의 Emulator를 사용하였고, 메모리 검색 및 변조를 위해 Cheat Engine을 사용하여 실험 환경을 구축하였다. 실험에 적용된 게임은 Handy Games의 Clouds and sheep과 DroidHen의 Defender, Frojo Apps의 Moy Farm Day이며 세 게임 모두 메모리 변조 취약점을 가지고 있었다.

게임에서 보이는 게임머니, 게임점수, 아이템, 레벨, 경험치 등 모든 수치는 모두 메모리에 적용되어 있다. 따라서 메모리에 기록된 값이 어떤 부분인지만 알아내면, 그 부분을 바꿔 버림으로써 순식간에 속임(Cheat)이 가능해지게 된다.

이러한 메모리 수치를 알아내기 위해 게임머니의 값을 메모리 영역에서 찾아낸다. 결과 값은 동일한 값이 존재할 수 있으므로 이를 추적하기 위해 게임머니나 점수 값에 변화를 주고 다시 값을 찾아내는 것을 반복하게 되면 실제 게임머니나 점수가 저장되는 메모리 위치의 범위가 좁혀지게 되고 이를 쉽게 찾아낼 수 있다.

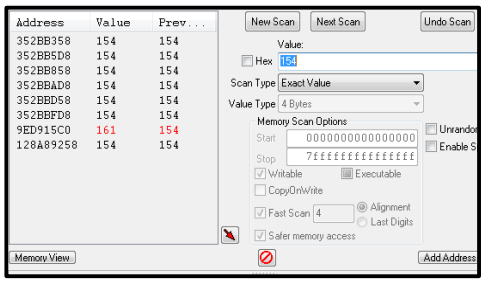
물론 대부분의 게임에서는 클라이언트에 저장되는 메모리 값은 서버에서 처리한 내용을 보여주는 용도 정도로만 사용하고 있는 것도 있다.

이러한 경우는 크게 문제될 것이 없으나 일부 게임에서는 게임의 구조상, 그리고 서버에서의 오버헤드를 줄이기 위해 클라이언트의 데이터를 신뢰하여 그대로 게임에 적용하기도 한다. 이런 경우 메모리 변조가 큰 문제를 야기할 수 있다고 하겠다.

4.2 실험 결과

[Fig. 1] defender

실험을 위해 게임 실행 상태에서 변조할 값을 확인하고 메모리에서 해당 값을 검색하였다.



[Fig. 2] Search

검색 결과 동일한 값을 갖는 값들을 다수 확인되었으며 실제의 값을 찾기 위해 게임을 진행하여 점수에 변화를 주고 변화된 값을 다시 검색하기를 반복하여 실제 값의 범위를 좁혔다.

[Fig. 3] defender modulation

이를 통해 찾아낸 실제 메모리 값을 [Fig. 3]과 같이 원하는 값으로 변경하면 게임을 진행하지 않고도 높은 점수를 얻을 수 있었다.

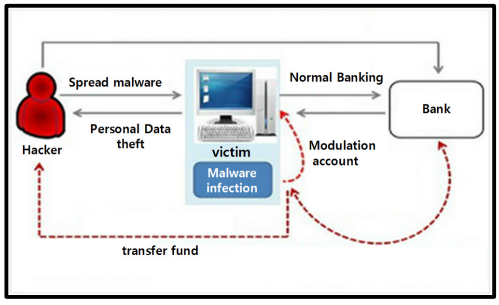
[Fig. 4] Clouds and Sheep modulation

[Fig. 5] Moy Farm Day modulation

[Fig. 4,5]는 다른 실험 대상인 Clouds and Sheep과 Moy Farm Day의 게임머니 변조화면으로 동일한 방법으로 게임머니 변조가 가능한 것을 확인 할 수 있다.

실험 결과 중요 데이터가 메모리에서 보호되지 않고 노출된다면 값의 검색과 값의 변화만으로 쉽게 메모리 위치를 알아낼 수 있으며, 이렇게 찾아낸 값을 원하는 값으로 쉽게 변조가 가능한 취약점이 확인 되었다. 물론 실험 대상을 찾는 과정에서 메모리에서 중요한 데이터를 검색 또는 변조 할 수 없게 데이터를 암호화 하거나 루팅이 된 Emulator가 탐지되면 게임이 진행되지 않는 등의 대응을 하고 있지만 아직까지 그러한 대응이 이뤄지지 않는 게임 또한 많이 존재한다.

이러한 메모리 해킹 위협은 안드로이드 게임에 국한되는 것은 아니다. 안드로이드와 같은 플랫폼과 스마트폰 환경에서 사용하고 있는 모든 애플리케이션이 메모리 해킹의 피해를 받을 수 있으며 특히 최근 이슈가 되고 있는 금융 앱 또한 이와 같은 메모리 변조의 위협이 있다.



[Fig. 6] Finance app Memory hacking overview

[Fig. 6]은 최근 발생하고 있는 금융앱 메모리 해킹 개요도이다. 해킹 절차를 살펴보면, 피해자 PC에 악성코드를 감염시킨 후 인터넷뱅킹 거래를 위해 수취계좌 및 이체금액을 입력하는 과정에서 악성코드에 의해 입력한 정보를 획득하고 인터넷 뱅킹을 중지 시킨다.

피해자는 이체에 필요한 계좌비밀번호, 보안카드번호, 공인인증서 비밀번호 등을 입력한다.

이후 메모리 해킹공격이 시작되며 피해자가 비밀번호 입력을 완료하면 해커는 수취인 정보 및 금액을 메모리에서 해커가 원하는 계좌 및 금액으로 변조하여 은행에 전송한다. 은행은 변조된 계좌로 자금을 이체하며 피해가 발생한다.

이처럼 기존의 공격방법과 메모리 해킹방법이 결합하여 새로운 피해로 발전할 수 있는 위험성이 존재하며 이는 안드로이드 게임 앱 또는 금융앱에 국한되지 않고 유사한 환경과 플랫폼에서는 언제든 발생할 수 있는 잠재적인 위험이 된다.

4.3 대응 방안

게임 개발사들은 이러한 해킹공격에 대해 시큐어 코딩, 패킹 등의 안티 리버싱 기술을 적용하거나 안티바이러스 솔루션을 통해 해킹을 차단하고 있다. 하지만 이러한 보안 솔루션의 경우 대부분 구조가 파악되어 우회기법이 나오면서 효과적인 대응이 어렵게 되었다.

또한 Signature를 기반으로 알려진 해킹 툴을 탐지하거나 Win32 API 기반의 탐지 방식을 사용하고 있지만, 이 또한 우회가 가능하다.

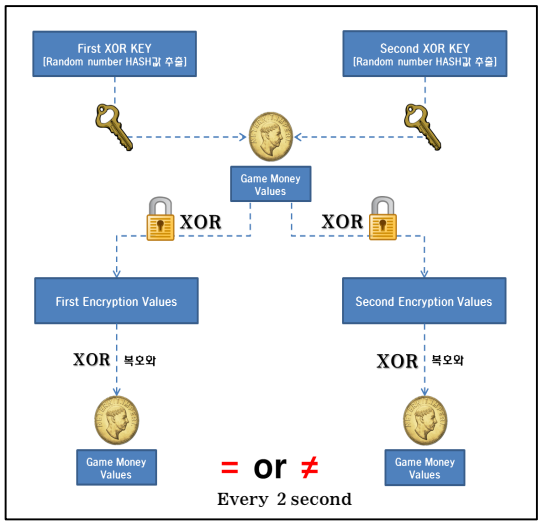
은행의 경우 안전한 거래를 위해 사용자 인증을 강화하고 있다. 하지만 이러한 방법은 비교적 안전하지만 사용자의 불편을 초래할 수 있으며 짧은 안드로이드 게임의 생명주기에 비해 많은 시간과 비용이 소요될 수 있어 안드로이드 기반의 소셜 게임에 적용하기는 어려운 단점이 있다.

따라서 본 연구에서는 현실적으로 적용이 가능한 난수에 SHA512 알고리즘을 적용한 데이터 암호화와 데이터 검증 방법을 제안하고자 한다.

실험에서 증명했듯 클라이언트의 특정 데이터 값이 저장된 위치를 검색해서 원하는 값으로 조작할 수 있음을 확인했다. 이를 방지하기 위해 게임머니, 점수, 경험치, 캐시 등 중요 데이터의 실제 값을 찾을 수 없게 하는 방법이 필수적으로 필요하다.

제안한 방법의 경우 임의의 난수를 생성하고 난수의 합에 복호화가 없고 알고리즘 비도 측면에서 본 값을 추측하기 어려운 SHA512 Digest를 추출하여 XOR 킷값으로 저장한다. 이렇게 추출된 HASH 값은 각각 XOR 키가 되며 이 킷값을 이용하여 은닉할 데이터와 XOR연산을 한다.

또한 검증을 위해 암호화된 게임머니를 2초단위로 복호화(XOR)하여 두 값을 서로 비교하여 일치하지 않을시 메모리 변조로 판단하여 게임을 종료하는 루틴을 적용하는 방식이다.



[Fig. 7] Data Encryption using Random number

[Fig. 7]은 random number HASH 값을 이용한 게임 데이터 암호화 및 비교절차를 그림으로 표현한 것이며 [Fig. 8]은 이를 소스코드로 구현한 것이다.

```
from random import randint
import hashlib
from time import sleep

def sha(randSum):  # SHA-512 해시함수
    randSum = str(randSum).encode()
    m = hashlib.sha256()
    m.update(randSum)
    result = m.hexdigest()
    return result

def randGen():  # 난수 3개를 정의하고 그 합을 반환
    result1 = randint(13421,40127)
    result2 = randint(40397,612767)
    result3 = randint(620123,992767)
    return (result1 + result2 + result3)

def XORMoney(money, key):
    return money ^ key     # 실제 데이터와 XOR킷 값의 XOR연산

gmoney = 100    # 암호화 대상인 게임머니
sum1 = randGen()
sum2 = randGen()

XORKey1 = sha(sum1)  # 난수합에 SHA512 해시값 추출후 XOR키 생성
XORKey2 = sha(sum2)

Encryp1 = XORMoney(gmoney,int(XORKey1,16)) #Data, XOR키 XOR연산
Encryp2 = XORMoney(gmoney,int(XORKey2,16))

while True:     # 무결성 검증 루틴
    sleep(2)
    firstDecryp = XORMoney(Encryp1,int(XORKey1,16))
    secondDecryp = XORMoney(Encryp2,int(XORKey2,16))
    if firstDecryp != secondDecryp:
        print("\n Data manipulated! ending program!!")
        exit(0)
    else:
        print("\n No change in data")
```

[Fig. 8] Development of Encryption Code

결과확인을 위해 출력문을 추가하였으며 그 결과는 [Fig. 9]와 같다. 각각 다른 XOR 킷값을 생성하여 은닉할 데이터와 XOR를 통해 암호화 하며 검증을 위해 복호화(XOR) 했을 땐 실제 데이터 값인 100이 동일하게 출력된다. 이 결과를 서로 비교하여 값이 서로 다를 경우 메모리 변조로 판단하여 메모리 조작에 사용되는 API를 차단하고 프로세스 정보를 관리서버로 전송하며 대상 프로세스를 종료하는 대응을 적용할 수 있다.

```
C:\Python34>python Encrypt.py
Salt_sum1 :  993105
XOR_Key1 :  784c463308bcecab1d6ee28f9b555f89a00660d35f2b89dc0bd7282d4f4233
Salt_sum2 :  1236227
XOR_Key2 :  55665ab58993b0e58e29697cc48e2f54769a004c8f2fb42d6cd67adf639ce7
Game_Money:  100
Encrypt1:  0x784c463308bcecab1d6ee28f9b555f89a00660d35f2b89dc0bd7282d4f423
Encrypt2:  0x55665ab58993b0e58e29697cc48e2f54769a004c8f2fb42d6cd67adf639ce
Decrypt_values :  100 100
No change in data
```

[Fig. 9] Encryption results

이상의 실험에서처럼 메모리에 나타나는 값을 난수와 해시함수(SHA512)를 통해 암호화, 메모리 해킹 툴 탐색과 값의 추측을 차단할 수 있다.

4.4 안전성 검증

제안한 방법의 안전성 검증을 위해 다음 4가지 사항을 고려하여 증명하였다.

첫 번째 서비스 부하방지를 위해 처리가 간단하고 복잡하지 않은 암호화 방식을 적용한다. 두 번째 게임의 중요 데이터에 대해 암호화로 해킹 툴에 의해 검색되지 않아야 한다. 세 번째 암호화 된 데이터는 안전한 암호화 킷값을 적용하여 공격자로부터 추측 또는 복호화가 불가능해야 한다. 네 번째 단방향 해시함수의 단점인 Digest 추측에 의한 Rainbow attack에 대응할 수 있어야 한다.

4.4.1 데이터 암호화로 해킹 툴 검색 차단

서비스 부하 방지를 위해 간단한 암호화 키 생성에 중점을 두었으며 암호화 킷값 생성 시 간단하고 빠르지만 복호화가 불가능한 SHA512 해시 알고리즘을 적용하였다. 이는 DES와 같은 블록암호 알고리즘이 아닌 SHA512 해시 전용 알고리즘으로 블록암호 알고리즘에 비해 처리 속도가 빠른 특징이 있다. SHA512 해시 알고리즘은 다양한 가변 길이의 입력에 대응하는 고정된 길이의 출력을 만드는 특징으로 모든 데이터에 적용이 가능하다.

4.4.2 서비스 부하 방지를 위한 방법

현재 소셜 게임 해킹공격 대응은 실제 데이터를 메모리에서 찾아낼 수 없게 은닉하는 것과 메모리 조작에 필요한 해킹 툴을 차단하는데 맞추어져 있다. 본 연구에서 제안하는 데이터 은닉방법은 암호화 킷값을 생성하여 실 데이터와 XOR 연산을 통해 데이터를 은닉하며 이때 암호화 킷값은 안전성이 검증된 SHA512 해시 함수를 사용하였다.

4.4.3 안전한 암호화 키값 생성

암호학적 해시함수의 경우 일방향성과 충돌회피성을 만족하여야한다.

일방향성이란 H(x) = h의 경우 이미 정의된 해시 값 h에 대응하는 데이터 x를 찾는 것이 계산적으로 불가능해야하는 성질로 x로부터 h의 계산은 쉬우나 h로부터 x의 계산은 불가능한 성질이다.

충돌회피성의 경우 동일한 Digest를 갖는 두 개의 데이터를 찾는 것이 계산적으로 불가능해야하는 성질이다. 즉 H(x) = H(y)인 어떠한 (x, y)의 쌍을 찾는 것이 계산적으로 불가능한 성질이다.

본 연구에 활용된 SHA512 해시 알고리즘은 이러한 암호학적 해시함수의 요구조건을 충족한 것이 이미 검증된 알고리즘으로 안전한 암호화 킷값 생성에 사용되었으며 이 킷값을 찾아내는 것이 사실상 불가능에 가깝기 때문에 실제 데이터를 찾는 것은 불가능에 가깝다.

4.4.4 Rainbow Attack에 대한 대응

Rainbow table Attack이란 가능한 모든 해시의 쌍(원본 : Digest)을 미리 계산하여 무작위 공격을 하는 해킹방법으로 가능한 해시의 쌍이 온라인을 통해 확산될 정도로 보편화 되어있다. 이에 해시함수 적용 시 난수로 생성된 salt값을 추가하여 값을 추측할 수 없게 하였으며 공격이 어렵게 함으로써 Rainbow table Attack에 대응하였다.

5. 결 론

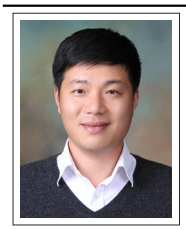
국내 뿐 아니라 해외에서도 소셜 네트워크 게임은 상당한 사용자층을 확보하고 있으나 메모리 해킹, 네트워크 패킷변조, 게임 repackaging 등 다양한 보안위협에 노출되어 있다. 소셜 네트워크 게임의 보안위협은 본 연구에서 제안한 메모리 데이터 암호화만으로 대응하기란 물론 어렵다. 스마트 폰이란 디바이스와 안드로이드 OS가 가지는 보안위협, 애플리케이션 개발 및 서비스에서의 보안위협, 유통단계에서의 보안위협 등 많은 보안위협 요소들이 존재하고 이런 많은 보안위협 요소들에 모두 대응하기란 시간적, 비용적인 부분에서 제약이 많다. 하지만 단계별 보안 대응과 함께 본 연구에서 제안하는 메모리 데이터 암호화를 적용한다면 현재보다 훨씬 더 좋은 대응을 기대 할 수 있으리라 본다. 물론 SHA512를 적용하기 전보다는 성능저하가 다소 발생할 수 있지만, 최근에는 모바일 디바이스의 하드웨어 성능 발전으로 인해 충분히 적용 가능하다[20].

본 연구에서는 안드로이드 환경과 모바일 게임이라는 특성에 맞는 간단한 구현과 서비스에 부하를 주지 않는 방향으로 해결방안을 제시하였다. 본 연구가 향후 안드로이드 기반의 모바일 게임의 메모리 해킹 방지뿐만 아니라 안드로이드를 기반으로 하는 스마트 폰 애플리케이션 보안 분야에 기여할 수 있기를 기대하며 이의 추후 연구가 필요할 것으로 본다.

REFERENCES

[1] KISA, “Statistical Survey Report of Mobile Internet”, 2012.
[2] KISA, “Survey Report of Mobile Internet”, 2013.
[3] KOCCA, “Present and Future of Social network game”, 2010.
[4] DMC Media, “Present and Future of Social network game”, 2010.
[5] Huykang Kim, YoungJun Keum, “Mobile game services security issues on Android environment”, 2013
[6] Yuxue Piao, JinHyuk Jung, JeongHyun Yi, “Structural and Functional Analyses of ProGuard Obfuscation Tool”, The Journal of Korea Information and Communications Society, 2013
[7] Lee Ju Yeob, “A Study on Programs Development of Online Game with Tightened

Security", 2012.
[8] Kaspersky Lab. Security News, "Researchers Find Methods For Bypassing Google's Bouncer Android Security", June 2012
[9] Justin Case, "Report: Google's Android Market License Verification Easily Circumvented, Will Not Stop Pirates", August 2010.
[10] Huykang Kim, YoungJun Keum, "Mobile game services security issues on Android environment", 2013
[11] SangHo Lee, Dayoung Ju, "An analysis of vulnerability and the method to secure on Android SNS applications from alteration of the code segments", 2013
[12] KiSung Lee, Huykang Kim, "Android Game Repackaging Detection Technique using Shortened Instruction Sequence", Korea Game Society, v.13, no.6, pp.85-94, 2013
[13] Soonil Kim, Sunghoon Kim, Dong Hoon Lee, "A study on the vulnerability of integrity verification functions of android - based smartphone banking applications", Journal of the Korean Institute of Information Security and Cryptology v.23 no.4 , pp.743 - 755 , 2013
[14] Yeonbi Chun, Sung Kyun Chang, Tack Woo, "Classification of Smartphone Game based on Mechanics", Korea Game Society, v.12, no.6, pp.15-24, 2012
[15] Justin Case, "Report : Google's Android Market License Verification Easily Circumvented, Will Not Stop Pirates", 2010.
[16] Kaspersky Lab. Security News, "Researchers Find Methods For Bypassing Google's Bouncer Android Security", June 2012
[17] Jiyoung Woo, Ah Reum Kang, Huy Kang Kim, "Modeling of Bot Usage Diffusion across Social Networks in MMORPGs," Workshop at ACM SIGGRAPH ASIA 2012, pp. 13-18, November 2012.
[18] DongYoung Woo, DongNam Seo, HuyKang Kim, JinYoung Choi, "A Study for Effectiveness of Preliminary Security Assessment on Online Game Service Domain", Journal of the Korea society of IT services, 2011
[19] Jiyoung Woo, Huy Kang Kim, "Survey and Research Direction on Online Game Security," Workshop at ACM SIGGRAPH ASIA 2012, pp. 19-25, November 2012.
[20] Hanumantu Rajeswari, Ramesh Yegireddi, Vudumula Govinda Rao, "Performance Analysis of Hash Algorithms and File Integrity", Hanumantu Rajeswari et al, International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7376-7379



이 상 원 (Lee, Sang Won)

2002년 2월 군산대학교 컴퓨터정보과학과 졸업
2013년 9월-현재 고려대학교 정보보호대학원 석사과정

관심분야 : 모바일게임 보안, 네트워크 보안, 안드로이드 애플리케이션 보안, 악성코드 분석

김 휘 강 (Kim, Huy Kang)

1998년 2월 KAIST 산업경영학과 학사
2000년 2월 KAIST 산업공학과 석사
2009년 2월 KAIST 산업및시스템공학과 박사
2004년 2월-2010년 2월 엔씨소프트 정보보안실장
2010년 3월-현재 고려대학교 정보보호대학원 조교수

관심분야 : 온라인게임 보안, 네트워크 보안, 네트워크 포렌식

김 은 진 (Kim, Eun Jin)

1999년 2월 KAIST 산업경영학과
2001년 2월 KAIST 경영공학과 석사
2007년 8월 KAIST 경영공학과 박사
2008년 9월-현재 경기대학교 국제산업정보학과 부교수

관심분야: 경영정보시스템, 보안경제학