情 報 保 護 學 會 誌
第23卷 第2號, 2013. 4

안드로이드 환경에서의 모바일 게임 서비스 보안이슈

김 휘 강*, 금 영 준**

요 약

최근 모바일 게임서비스의 이용자가 증가함에 따라 스미싱 등 결제부정행위에 악용되는 사례들이 증가하고 있다. 이에 따라 안드로이드 환경에서 동작하는 스마트폰 내 모바일 게임 서비스에 대한 보안 요구사항이 게임앱 개발사, 게임서비스 제공사, 유통플랫폼 기업들에 요구되고 있다. 모바일 게임 서비스 보안은 PC 상에서의 게임보안과 어떤 차이점들이 존재하며 모바일 게임 내에 존재하는 취약점들의 유형은 어떤 것들이 있는지 살펴보고, 이에 대한 대안 및 기술적 한계를 살펴보도록 한다.

Ⅰ. 서 론

온라인게임보안은 온라인게임을 대상으로 한 위협이 최근 증가함에 따라 새로이 연구되고 있는 분야라 할 수 있다[11, 12]. 온라인게임을 개발 및 서비스 할 때에 게임개발사 (game studio) 및 게임서비스사 (game publisher)에서 고려해야할 보안 및 게임이용자를 위한 보안, 그리고 게임서버 및 클라이언트 프로그램 내부의 보안 (in-game security) 등 보안이 요구되는 대상에 따라 다양한 보안 기술이 존재한다.

주로 온라인게임서비스는 PC 플랫폼 및 콘솔기종을 대상으로 확산되어 왔으나, 스마트폰의 이용 증가에 따라 모바일게임서비스 역시 증가하고 있다. 다만, 단기간에 성장을 하게 됨에 따라 보안을 고려하지 않고 개발이 이루어지는 경우 역시 존재하며, 스마트폰 OS 자체의 취약점에 기인하여 결제 부정 및 계정 도용 등 피해가 발생하고 있다.

2012년의 ArXan 사의 보고서에 의하면[1] 유료 앱의 90%가 해킹가능하며, iOS기반 유료 앱의 92%가, 안드로이드 기반 유료 앱의 100% 가 모두 해킹 가능한 것으로 조사되었다. 무료 앱 역시 마찬가지로 40%의 iOS 기반 앱이, 80%의 안드로이드 기반 앱이 해킹 가능한 상태로 조사된 바 있다. 전통적으로는 시큐어코딩 기법을 적용하여 개발상의 취약점을 최소화 하는 것으로 보안대책을 강구해 왔으나 모바일 앱의 경우에는 OS 의 취약점, 소셜엔지니어링에 악용될 수 있는 취약점, 앱이 연계되어 구동되는 SNS플랫폼의 취약점이 복합적으로 작용하기 때문에 앱 자체만의 개발보안을 준수하는 것만으로는 적절한 보안수준을 유지하기 어렵다는 문제점이 존재한다.

PC플랫폼 기반에서의 온라인게임의 경우 서비스거부공격이나 웹해킹, 시스템 침입을 통한 데이터베이스 조작과 같은 전통적인 위협 외에도 그림 1에서 보듯이 계정도용을 유발하는 악성코드 및 게임봇과 이를 이용하여 금전적인 이익을 얻으려는 작업장(gold-farming workshop)[8,9] 과 같은 온라인 게임 상에서만 존재하는 위협이 존재해 왔다.

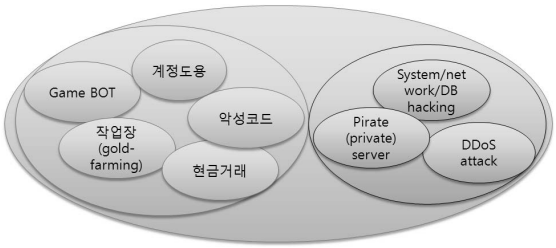
게임개발사 및 게임보안기업들이 이에 대한 대응방안을 그림 3에서 보듯이 법적인 대응 방안, 기술적인 대응방안, 운영적인 대응방안, 게임 기획차원에서의 대응방안을 꾸준히 개발해 온 상태이다.

불법으로 사설서버를 운영하거나 게임봇을 제작하여 판매, 유포를 하는 사이트를 발견 시에 민형사 대응, 사이트 폐쇄, 광고 차단 등의 법적인 대응, 게임운영자들이 상시 게임 내부를 모니터링 하여 사기, 현금거래가 발생하는 것을 탐지해 내는 운영적인 대응방안 및 기술적인 보호 조치가 상당한 발전을 이루고 있다. in-game CAPTCHA 를 이용하여 사용자와 게임봇을 구분하는

* 고려대학교 정보보호대학원 (cenda@korea.ac.kr)
** 고려대학교 정보보호대학원 (0junkum@korea.ac.kr)

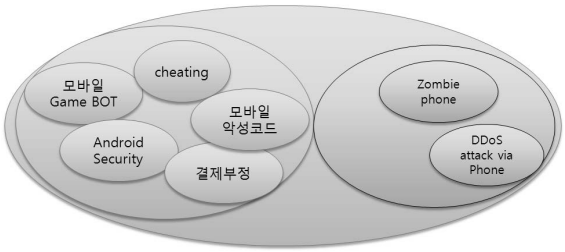
기법이나, 게임액션로그를 분석하여 행위기반으로 사용자와 게임봇을 구분하는 기법 등이 그 예라 할 수 있다. 빅데이터 분석 기법을 이용하여 채팅로그분석을 기반으로 게임봇을 탐지하는 기법[5], 파티플레이로그분석을 기반으로 게임봇을 탐지하는 기법[6], 게임봇을 확산모델(diffusion model)기반으로 분석한 기법[7] 등 다양한 방법이 그간 연구되어 왔다.

그리고 게임 기획 차원에서의 대응방안으로는 아이템을 취득한 사람만이 이용할 수 있고 매매하지 못하도록 귀속 처리해 게임봇이 아이템 거래를 통한 금전적인 이득을 얻지 못하도록 설계하거나, 반복적으로 고정적인 행위 (예: 몬스터를 반복적으로 사냥하는 게임봇 또는 매크로 프로그램을 이용하는 경우)를 할 경우 새로 출현하는 몬스터의 레벨을 점차 강하게 하여 게임봇의 효율을 떨어뜨리도록 한 예를 들 수 있는데, 게임 기획 차원의 대응방법은 사용자들이 별도의 보안솔루션을 설치하지 않더라도 컨텐츠 자체만의 특성을 이용하여 서비스 보안수준을 높인 모범적인 예라 할 수 있다.



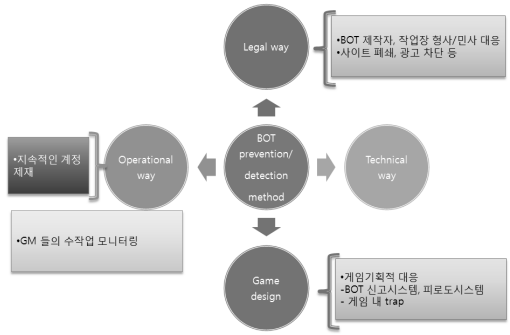
〔그림 1〕 PC 게임 환경에서의 보안 위협[11,12]

PC 플랫폼 기반 게임서비스와 모바일 게임은 다소 위협요소들이 다른데, 그림 2에서 보듯이 안드로이드 플랫폼 자체의 취약점으로 인해 기인하는 문제들이 대부분이며, 이로 인해 스마트폰이 악성코드 및 해킹에 취약해 좀비폰으로 될 수 있는 문제가 존재한다. 최근에는



〔그림 2〕 모바일 게임 환경에서의 보안 위협

스마트폰 및 스마트 단말기기들의 성능이 좋아짐에 따라 게임 점수 정도를 치팅 하는데 그쳤던 게임관련 위협이 게임봇이 출현할 수 있는 환경으로 점차 진화하고 있으며, 많은 공격이 스미싱과 같이 금전적인 이익을 노리는 공격으로 진화되고 있는 추세이다.



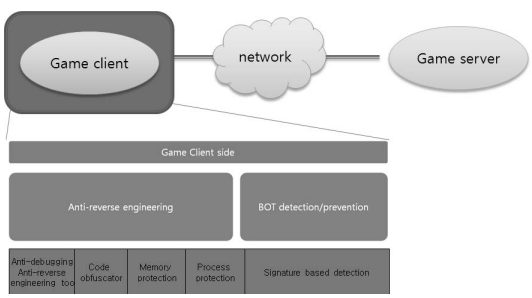
〔그림 3〕 온라인게임서비스에서의 대응 프레임워크

Ⅱ. PC 게임보안과 모바일게임 보안의 차이점

온라인게임서비스를 할 때 보안을 적용할 수 있는 구간은 게임클라이언트 (PC 또는 스마트폰)단, 네트워크의 전송단, 게임서버단으로 나누어 볼 수 있다. 게임클라이언트 구간의 경우 PC 플랫폼에서는 그림 4에서 보듯이 리버스엔지니어링을 이용하여 게임클라이언트를 역으로 분석하기 어렵도록 packer 나 anti-debugging 툴을 이용하여 난독화(obfuscation)를 적용시키는 것에 초점이 맞추어져 있다.

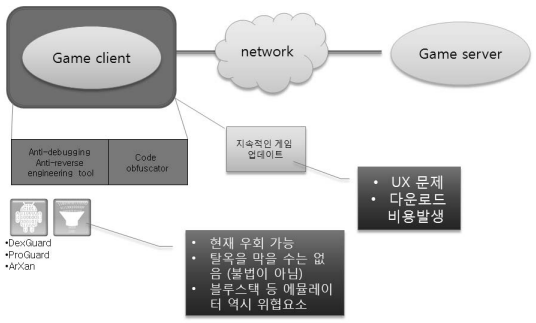
대부분의 게임보안솔루션들이 클라이언트 단에서 게임프로그램의 프로세스 및 메모리를 보호하고, 실행파일 바이너리를 난독화하는 역할을 수행한다.

모바일 게임서비스의 경우에도 게임클라이언트 단에



〔그림 4〕 PC 플랫폼 게임에서 게임클라이언트 단의 safeguard [11,12]

서 적용할 수 있는 보호조치는 주로 anti-debugging, 바이너리 코드 난독화 기술인데, PC플랫폼과 가장 큰 차이점이 있다면 게임앱을 구동할 수 있는 에뮬레이터를 쉽게 구할 수 있어 해커가 디버깅을 하기 용이하고, OS가 탈옥(jail-break)이 되어 플랫폼 보안이 취약해 진 상태라 하더라도 이를 법적으로 제재할 수 있는 근거가 미약해 프로세스 및 메모리 보호를 하는 보호조치들이 동시에 취약해 질 수 있는 문제가 존재한다.

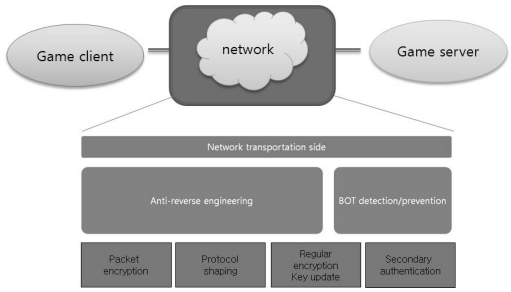


〔그림 5〕 모바일 게임에서 게임클라이언트 단의 safeguard

그리고 PC 플랫폼에서는 게임클라이언트가 해커들에 의해 분석되더라도 분석된 결과를 지속적으로 악용하지 못하도록 정기적으로 게임클라이언트를 업데이트 하여 내부 로직 및 패킷통신 구조를 변경하는 방식을 적용할 수 있는데, 모바일게임의 경우 이런 보호조치가 사용자에게 불편함을 초래할 수도 있고, 다운로드 비용을 발생시키므로 게임이용자가 현재 사용 중인 이동통신사 요금제도에 따라 통신비용을 발생시킬 수도 있다는 점에서 쉽게 적용하기 어렵다.

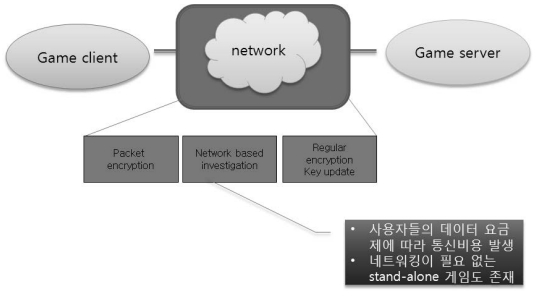
네트워크 단에서는 PC 플랫폼 게임의 경우 그림 6에서 보듯이 네트워크 패킷의 암호화를 통해 기밀성과 무결성을 확보하는 쪽에 초점이 맞추어져 있다. 즉, 해커가 게임데이터를 훔쳐보거나 게임 내 명령을 전송하는 컨트롤 패킷의 구조를 알아볼 수 없도록 하고, 정기적으로 게임클라이언트와 서버간의 통신프로토콜을 변경하여, 이미 리버스엔지니어링에 의해 게임봇이 출현했다 하더라도, 게임봇 제작자가 지속적으로 리버스엔지니어링을 해야만 지속적으로 봇을 구동시킬 수 있도록 리버스엔지니어링의 작업을 방해하도록 한다.

게임클라이언트와 게임서버 간에 별도의 조사패킷(investigation packet)을 정의하여, 원격에서 현재 구동중인 게임클라이언트가 정당한 (genuine) 게임클라이언트인지를 조사하는 방법 역시 네트워크 단을 통해 이루어지게 된다.



〔그림 6〕 PC 플랫폼 게임에서 네트워크 단의 safeguard [11,12]

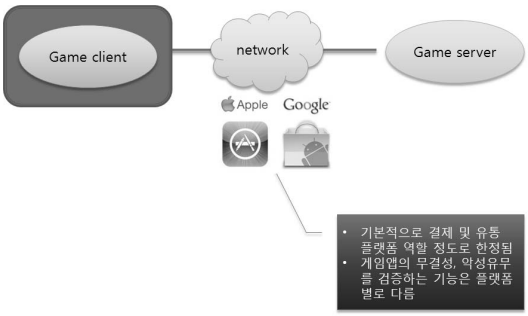
모바일 게임에서는 이러한 작업을 수행하기 어려운데, 그림 7에서 보듯이 모든 모바일게임이 네트워크 플레이를 상시 요구하지 않으므로, 상대적으로 한번 리버스엔지니어링을 통해 게임이 분석될 경우 네트워크 통신 프로토콜을 변경하거나 조사패킷을 정기적으로 보내 확인할 수 있는 가능성이 현저히 줄어들게 된다. 더불어 네트워크 단의 보안기능을 변경할 경우 지속적으로 변경된 게임앱을 앱스토어 등 앱마켓에 업로드 하여 유통시켜야 하는 부담이 발생하게 된다.



〔그림 7〕 모바일 게임네트워크 단의 safeguard

게임클라이언트 단에서 언급된 문제와 마찬가지로 지속적인 통신을 유발시키는 형태로 보안성을 강화하는 것은 가입자의 통신요금제도에 따라 데이터통신비용을 발생시킬 수 있기 때문에 적용하기 어려운 단점이 존재한다.

이와 같이 게임의 구동 시에 네트워크 단에서 게임보안을 강화하는데 한계가 있으므로, 게임의 유통플랫폼

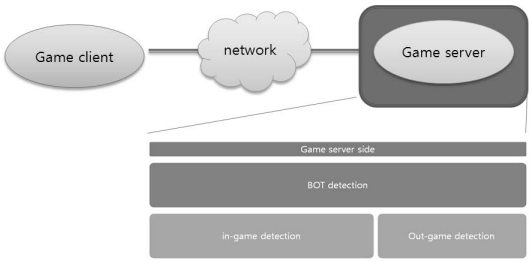


〔그림 8〕 모바일 게임에서 유통플랫폼단의 safeguard

(애플 앱스토어, 구글 플레이스토어 등) 을 통해 최초 설치 시 또는 추가 컨텐츠를 다운로드하여 설치시, 또는 결제 부정행위가 발생 시 조치할 수 있는 기능을 많은 모바일게임개발사들이 희망하고 있으나, 현재 유통플랫폼은 결제와 유통에만 그 역할이 치중되었으며, 게임앱 자체의 기밀성이나 무결성을 검증하는 것은 약하다고 할 수 있다. 구글 플레이스토어의 경우에는 앱의 악성유무를 사전에 다 검증하지는 않으며, 정상적인 유통플랫폼 외에도 apk을 별도로 다운로드하여 설치할 수 있는 방법이 있으므로, 이를 통해 무결성이 담보되지 않은 가짜 앱이 유포되기도 한다.

이 때문에 책임소재가 불분명한 문제 역시 존재한다. 게임 내 아이템 구매 또는 부가컨텐츠 결제 상에 부정이 일어난 경우 온라인게임 서비스 회사와, 이동통신사, 결제대행사 (PG;Payment Gateway) 및 플랫폼사 중 어느 곳이 어떤 사안에 대해서 책임소재가 있는 것이 명확한 상태는 아니다. 예를 들어 게임 내 아이템 중에서 소모성 아이템 (1회 이용 시 사라지는 아이템)을 구매하는 과정에서 결제부정이 발생했을 경우 이동통신사 및 결제대행사는 이러한 결제부정이 대량으로 발생 시 모니터링을 통해 이상증후를 감지할 수 있었음에도 불구하고 적극적인 대응을 하지 않은 것에 대한 책임소재가 존재할 수 있으며, 게임사의 경우에는 게임이용자에게 결제 취소를 승인하고 복원을 해주어야하는데, 소모성 아이템의 경우 이미 사용되어 사라진 경우, 교환 및 환불의 대상 자체가 소멸하여 보상을 하기 어려운 점이 존재한다.

그림 9에서 보듯이 게임서버단에서 적용할 수 있는 보안은 특정 보안솔루션에 의존한다기 보다는 게임개발사 자체의 in-game 로직으로 방어하는 것에 의존하고 있다. 게임패킷을 분석하여 조작된 패킷을 차단하고, 게



〔그림 9〕 PC 플랫폼 게임에서 서버단의 safeguard [11,12]

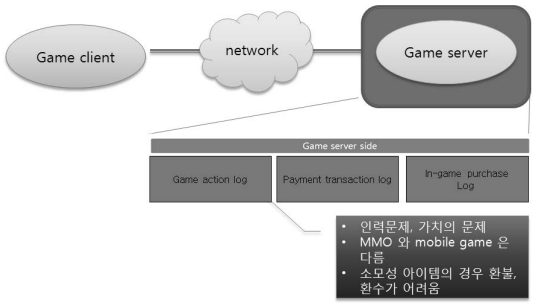
임 내부의 로직을 이용하여 치팅 및 부정행위가 발생하고 있는지를 탐지, 차단하는 것을 들 수 있다. 게임클라이언트 단의 보안은 별도의 보안솔루션을 설치해야 하는 경우가 많아 북미, 유럽권 등 PC 사용자의 문화에 따라 거부감을 살 수 있기도 하며, 커널 단에서 게임봇 프로세스가 게임프로세스를 후킹 하는 것을 보호하는 것이기 때문에 기 설치된 보안솔루션에 따라 충돌이 발생하기도 하는 단점이 존재한다. 그래서 많은 게임개발사들이 게임로그를 분석하여 게임클라이언트의 보안조치와 독립적으로 행위기반분석을 통해 불량사용자를 탐지해 내는 기법을 개발하여 적용하고 있다.

이를 위해 게임 장르별로 정도의 차이는 있지만, 특히 MMORPG (Massively Multiplayer Online Role Playing Game) 의 경우에는 게임 내 사용자별 액션을 상세히 로그를 남겨 빅데이터 기반 데이터마이닝을 통해 사용자의 행위분석을 수행하기 적합한 환경이다.

하지만, 모바일게임은 장르 자체가 대부분 캐쥬얼 게임이어서 스테이지 단위로 구성이 되어 있고, 스테이지당 플레이타임이 기존 PC환경에서의 온라인게임에 비해 짧으므로, 변별력을 충분히 확보할 수 있을 만큼 대량의 로그가 생성되기 어렵다. 더불어 게임 내 행위가 다채롭지 않고, 로그를 매 이벤트가 발생할 때마다 게임서버단에 생성하도록 요청을 할 수 없는 구조이므로, 로그분석을 통한 부정행위 적발이 용이하지 않다고 할 수 있다.

서론에서 언급했듯이 온라인게임보안의 범위에는 게임개발 및 서비스 상의 보안 외에도 게임이용자들을 위한 보안 영역이 존재한다. 많은 온라인게임회사들은 게임이용자들에게 온라인무료백신, 키보드보안, 보안패치서비스, 모바일OTP 등 PC의 보안을 강화할 수 있는 솔루션들을 무상으로 사용자들에게 제공해 주는 형태로 사용자의 보안성을 높이는 노력을 해주고 있다.

다만, 모바일 게임 서비스회사가 모바일용 백신, 모



〔그림 10〕 모바일 게임에서 서버단의 safeguard

바일용 키보드 보안 솔루션을 무상으로 공급하는 것도 무리가 있다. PC 게임의 경우에는 하드웨어 성능이 발전함에 따라 보안솔루션이 구동되어도 성능상 제약을 거의 받지 않으며, 사용자들이 해당 보안솔루션에 대한 이해도 (antivirus 및 Firewall)가 높은 편이어서 저항감이 상대적으로 덜한 편이나, 모바일게임은 주 장르가 MMORPG 가 아닌 캐쥬얼 게임으로 사용자들의 이용목적 자체가 짧은 시간에 가벼운 게임을 즐기는 것이 주목적이기 때문에, 모바일 게임플레이를 위해 보안 프로그램을 다수 설치해야 하거나 구동하는 것 자체에 거부감을 느끼는 경우가 많다. 더불어 스마트폰의 하드웨어 성능상 제약이 있기 때문에 강력한 보안기능을 적용할수록 주기적으로 CPU 사용을 할 경우 발열 및 배터리 소진 문제가 발생하게 된다.

물론, 모바일게임의 역사가 최근 1~2년으로 짧은 것이 아니다. 스마트폰 보급과 앱의 이용이 폭발적으로 증가함에 따라 모바일게임앱의 이용자층이 증가하여 보안이슈가 새로이 발생한 것일 뿐, 스마트폰이 등장하기 이전에도 모바일 게임은 존재했었다. feature phone 이 등장한 10년 전부터 모바일게임들은 존재했었으나 그 당시에는 큰 보안문제가 발생하지 않았던 것이, 데이터통신 요금 때문에 휴대전화는 MUG (Multi-User Game) 모바일게임 플랫폼으로 적합하지 않은 것이 가장 큰 원인이며, 이동통신사의 플랫폼 환경이 폐쇄적이어서 플랫폼에 무관하게 플레이 할 수 있는, 대규모의 사용자층을 확보한 게임이 없었기 때문이다.

그림 11에서 보듯이 해커들이 매력을 느끼는 공격대상으로 되려면 쉽게 해킹할 수 있고 금전적인 이득을 얻을 수 있으며, 공격대상에 접근성이 용이한 “공격의 경제성”이 있어야 하며, 패치가 어렵고 사용자가 심리적 방어기재가 약해 소셜엔지니어링 등 공격기법을 지

〔그림 11〕 공격의 경제성 및 공격의 지속성

속하기 용이한 “공격의 지속성” 이 성립되어야 한다.

과거 feature phone 환경에서의 모바일게임은 국내환경이 이동통신사의 표준에 의해 오프라인 상으로 앱을 등록시킬 때 휴대전화 기기별 이동통신사 검수를 받아야 했던 점이 제약을 발생시키고, 대개는 통신을 요구하지 않는 스탠드얼론으로 구동되는 패키지 게임이었기 때문에 결국은 접근성, 환금성 면에서 해커에게 매력적인 공격대상이 되지 못하였다고 할 수 있다.

스마트폰 환경에서의 모바일 게임 역시 초창기에는 치팅을 통해 최종 점수 정도를 게임스코어보드(leader board)에 조작하여 올리는 정도였으나 이 역시 통신료를 발생시키므로 게임 하이스코어를 굳이 등록하지 않는 경우도 많고, 이것이 금전적인 이익을 제공해 주지도 않으므로 공격의 가치가 적었으나, 최근 모바일게임들은 초기 설치시 비용을 지불하는 구매모델에서, 지속적인 컨텐츠 이용을 유도하고 아이템 구매를 요구하는 부분유료화 모델이 주력으로 자리잡아감에 따라 공격의 가치가 높아졌다.

‘애니팡’, ‘드래곤플라이트’ 와 같이 이용자층이 풍부한 게임이 등장함에 따라 위변조를 한 악성 앱을 배포하기 용이해 졌으며, 특히 대량의 사용자층을 확보한 온라인게임의 경우 SNS 플랫폼 상에서 구동되므로, 이용자가 신뢰하는 사람으로부터 초청메시지가 올 경우 의심 없이 수락하는 심리적인 점을 이용하여 악성코드를 배포하거나 악성 단축 URL 문자를 보내 공격자의 사이트로 접속을 유도하는 것이 점차 쉬워지고 있다. 불어서 현재까지 모바일플랫폼에 대한 방어기재가 거의 존재하지 않는 점, 사용자들의 보안마인드가 낮은 점, 공격대상이 도처에 있어 접근이 용이한 점, 모바일게임 회사들이 영세하여 보안에 투자를 하기 어려운 점 역시 모바일 게임이 “공격의 경제성”과 “공격의 지속성”을 높여주고 있다.

III. 실제 모바일게임 취약점 분석 사례

최근에는 언론사에서도 모바일게임상에 존재하는 취약점을 적극 소개할 정도로 심각성이 증가하고 있는 상태이다 [13, 14]. 본 논문에서는 현재 국내에서 많은 인기를 얻고 있는 모바일게임 중 하나를 선택하여 실제 어떠한 취약점이 존재하는지를 분석한 사례를 소개한다.

게임앱이 동작 시에 무결성을 확인하는지, 앱 내부적으로 코드난독화 처리가 되어 있는지, 사용되는 데이터의 난독화를 하고 있는지, 통신구간에서 암호화를 통해 트래픽의 기밀성을 유지하고 있는지, 기타 앱이 배포될 때 불필요한 디버깅 정보를 포함하여 리버스엔지니어링을 용이하게 하지는 않는지를 점검하였다.

점검 항목	요약	점검 결과
무결성 확인	클라이언트 프로그램 조작 여부를 확인하는지 점검	취약
코드 난독화	프로그램 코드에 대한 난독화 여부 점검	취약
데이터 난독화	주요 데이터에 대한 난독화 여부 점검	취약
네트워크 통신 암호화	통신에 주고 받는 데이터 암호화 여부 점검	보통
필요없는 정보	디버그 정보 등 실행에 필요 없는 추가 정보가 존재하는지 점검	취약

〔그림 12〕 A모 게임의 점검결과

게임앱에서 클라이언트의 코드가 임의로 변경 된 것을 확인하는 무결성 점검이 중요한데, 게임클라이언트는 악의적인 사용자에 의해 언제든지 변경/악용 될 수 있으므로, 초기 구동 시에, 그리고 가능하다면 게임플레이 중간에도 랜덤한 인터벌로 무결성을 점검하는 것이 필요하다.

많은 게임개발사에서 게임앱 배포시 정상프로그램의 해쉬값을 저장하고, 구동 초기에 동작하는 프로그램의 해쉬값을 비교하여 일치하지 않는 경우 게임실행을 종료하는 방식을 따르고 있다. 특히 네트워크를 통해서 해쉬값을 확인하는 방식은 유연성이 높아 보안성을 쉽게 높일 수 있다. 위변조 되지 않은 게임앱을 보유한 경우에만 유료컨텐츠 부가결제를 추가로 할 수 있도록 제약하는 정책 역시 가능한데, 게임개발사들이 수익성을 문제로, 최초 구매가 정상적인 경로로 이루어지지 않은 불법복제 앱이라 하더라도 이 앱에서 추가컨텐츠를 구매하는 것은 허용하는 정책을 고수할 수도 있기 때문에 적용에 있어서는 게임사의 재량에 달린 부분이기도 하다.

코드난독화는 안드로이드앱이 바이트코드로 구성되어 있어 디컴파일을 할 경우 거의 소스코드와 유사한 수준으로 복원을 해내는 것이 가능하여, 난독화를 적용하지 않을 경우에는 내부 로직상 취약점이 그대로 노출되는 문제가 있으므로 난독화 적용은 강력히 권고되는 사항이다. ArXan[2], ProGuard[3], DexGuard[4]와 같은 난독화 솔루션들이 현재 널리 사용되고 있다.

그림 13은 A모 게임을 디컴파일한 결과 암호화에 이용되는 고정키 값과 암호화 방식을 알아낸 예이다.

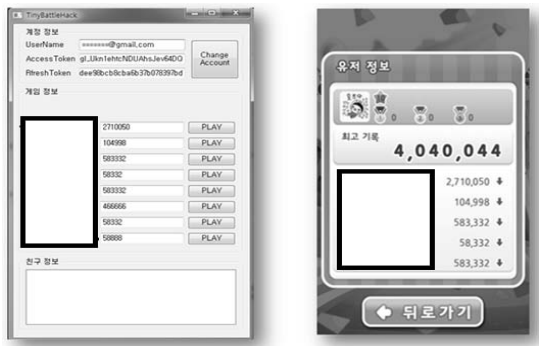
〔그림 13〕 A모 게임을 디컴파일 후 중요정보를 발견한 예

이와 같은 경우에는 현재 통신암호화가 이루어지고 있다 하더라도, 이 암호화키를 이용하여 암호화된 트래픽을 역으로 분석하여 평문을 추정해 내는 공격 역시 가능해 질 수 있다.

네트워크 통신이 암호화 되지 않아 패킷구조가 분석될 경우 점수조작은 물론, 게임 내 명령어를 이용하여 다른 사용자를 차단(ban)하는 공격 역시 가능하다.

A모 게임은 구동시 무결성을 확인하지 않고 있었으며, 난독화 툴을 적용하여 코드나 데이터의 보호를 하지 않고 있는 상태였고, 기본적인 통신암호화는 적용되어 있었지만 코드를 디컴파일 후 암호화키와 암호화 방식을 알아낼 수 있었으며, LogCat 을 통해 게임 구동시 상세하게 남는 정보를 이용 역으로 게임 로직을 추정해 내는 것이 가능했다.

그림 14는 이와 같이 발견된 취약점을 이용하여, 게임플레이를 전혀 하지 않고도 게임서버로 점수를 임의로 조작하여 보내는 공격을 한 예이다.



〔그림 14〕 발견된 취약점을 이용하여 게임플레이 없이 점수조작을 한 예

IV. Discussion

이와 같은 취약점이 다수 존재함에도 현재 확실한 보안대책은 존재하기 어려운 것이 현실이다. 그 이유는 기술적 원인 보다는 시장구조와 모바일게임서비스의 특성에 기인하는데, 모바일게임은 유사한 게임이 많고 참신한 아이디어로 승부하는 경우가 많아 시장에 적시에(in-time) 진입하지 않으면 게임의 완성도가 아무리 높다 하더라도 성공을 담보할 수 없기 때문에 장기간에 걸쳐 개발보안을 강화하기 어려운 것이 현실이다.

모바일 게임개발사 역시 많아지고 앱마켓 내에 워낙 많은 게임앱들이 있어 경쟁이 치열해 져서, 성공작을 만들어 내기 어렵고 설령 인기게임을 만드는데 성공하더라도 모바일게임의 수명은 3개월~4개월을 넘지 못하는 것이 일반적이다. 그래서 취약점이 발견된다 하더라도 수정과 재배포를 하면 어느새 게임 자체의 lifecycle 이 끝나므로, 자금여력이 있는 게임회사들조차 보안대응을 할 동기부여가 안된다는 점 역시 해결해야 할 숙제로 남아 있다. 이와 유사한 문제를 가졌던 게임 장르로는 플래쉬게임과 웹게임이 있었다. 플래쉬 용 난독화 툴이 있었지만, 플래쉬 게임 자체가 무료게임인 경우가 많고, 게임개발사들이 플래쉬 게임에까지 보안을 적용할만한 가치를 느끼지 못했기 때문에 스테이지 당 최대 획득가능한 점수를 계산해 두고 이 점수를 초과한 경우에 한해서 제재를 하는 정도의 간단한 대응이 전부였다. 웹게임 역시 취약점을 수정하는데 들어가는 비용과 새로운 게임을 개발하는데 들어가는 비용을 계산해 보면 게임 수정에 지속적인 노력을 들이는 것보다 새로운 게임을 개발하는 것이 게임사의 수익에 보다 부합하므로 보안상 이슈가 있어도 대응이 쉽지 않아 왔었다.

이와 같은 시장 환경 때문에 그림 3의 대응 프레임워크에서 언급한 법적인 대응, 기술적인 대응, 운영적인 대응, 게임기획상의 대응을 적용하는 것 역시 용이하지 않다.

불법 복제된 게임앱을 이용하였다고 하여 프로그램보호법으로 사용자들을 제재한다는 것은 현실상 많은 문제가 있으며 사용자들의 반발을 살 수 있다. 더불어 악성 앱을 제작하여 배포하는 해커를 추적하는 것 역시 민간게임회사들 입장에서는 수행하기가 어렵다.

기술적인 관점에서는 안드로이드OS가 보안상의 취약점이 아직 많이 존재하여 게임앱을 안전하게 개발하였다 하더라도 OS의 취약점으로 인해 스마트폰이 해킹당한 경우에는 다른 대책이 어렵다. 그리고 보안조치를 강화하면 할수록 데이터통신이 유발되거나 배터리소모속도가 빨라져서 적용이 어려운 면이 있고, PC기반 게임들처럼 많은 운영인원(GM; Game Master)을 투입하여 운영관제를 한다는 것은 게임 장르상, 게임 수익모델상 성립할 수 없다.

사용자층이 많고, SNS플랫폼에서 구동된다는 특성과 소셜엔지니어링 기법과 결합하여 금전적인 피해를 유발하는 많은 공격기법들이 등장하고 있으나, 근본적인 방어수단이 아직 없다는 점에서 최근에 등장한 위협 중 심각도가 높은 위협 중 하나라 볼 수 있다.

V. 결 론

국내 및 해외 모두 스마트폰의 여러 앱 중에서 게임은 상당한 사용자층을 확보한 인기 있는 인터넷 어플리케이션이 되었지만, 리버스엔지니어링에 의한 해킹, 결제부정과 스미싱 등 다양한 위협에 노출되어 있다.

기술적, 법적, 운영적 차원에서 대응이 쉽지 않고 게임클라이언트단, 네트워크단, 유통플랫폼단, 서버단 모두 보안대응수단을 적용하기 어려운 점이 존재하지만, 최소한 결제부정이슈에 대해서는 게임사들의 적극적인 대응이 필요하다. 결제 트랜잭션을 지속적으로 모니터링하면 결제부정을 일으키는 해커의 IP address 를 발견할 수 있고, 도용된 신용카드를 사용자 및 신용카드 회사에 신고하는 등 적극적인 수단을 강구할 수 있다. 결제 트랜잭션을 발생시킨 IP address, 지속적으로 결제부정을 일으키는 account 에 대한 정보를 관리하고 PG

사 및 유통플랫폼제공자와 공조하여 대응력을 높이는 것이 필요하다고 할 수 있다.

참고문헌

[1] “State of Security in the App Economy”. *Mobile Apps Under Attack*, Vol.1, 2012, ArXan Technologies, Inc.

[2] Arxan, http://www.arxan.com/

[3] ProGuard, http://proguard.sourceforge.net/

[4] DexGuad http://www.saikoa.com/dexguard

[5] Ah Reum Kang, Huy Kang Kim, Jiyoung Woo, ”Chatting pattern based game BOT detection: Do they talk like us?,“ *KSII Transactions on Internet and Information Systems*, 6(11), pp. 2866-2879, November 2012.

[6] Ah Reum Kang, Jiyoung Woo, Juyong Park, Huy Kang Kim, “Online Game Bot Detection based on party-play log analysis,” *Computers & Mathematics with Application*, February 2012.

[7] 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.

[8] 서동남, 우지영, 우경문, 김종권, 김휘강, “연결패턴 정보 분석을 통한 온라인 게임 내 불량사용자 그룹 탐지에 관한 연구”, *정보보호학회논문지*, 22(3), pp. 585-600, June 2012.

[9] Kyungmoon Woo, Hyukmin Kwon, Hyun-chul Kim, Chong-kwon Kim, Huy Kang Kim, “What Can Free Money Tell Us on the Virtual Black Market,” *ACM SIGCOMM 2011*, pp. 392-393, August 2011.

[10] Jiyoung Woo, Hwa Jae Choi, Huy Kang Kim, “An automatic and proactive identity theft detection model in MMORPGs,” *Applied Mathematics & Information Sciences*, 6(1S), pp. 291S-302S, January 2012.

[11] 유동영, 서동남, 김휘강, 최진영, “온라인게임 서비스 분야에 정보보호 사전진단 적용시 효과성에 관한 연구”, *한국IT서비스학회지*, 10(2), pp. 293-308, June 2011.

[12] Jiyoung Woo, Huy Kang Kim, “Survey and Research Direction on Online Game Security,” *Workshop at ACM SIGGRAPH ASIA 2012*, pp. 19-25, November 2012.

[13] “인기게임 드래곤플라이트 심각한 보안취약점 발견!”, 데일리시큐, 2012-11-16, http://www.dailysecu.com/news_view.php?article_id=3277

[14] “드래곤 플라이트 게임, 위변조 가능한 취약점 발견!”, 데일리시큐, 2012-11-26, http://www.dailysecu.com/news_view.php?article_id=3218

〈著者紹介〉

김 휘 강 (Huy Kang Kim)
종신회원

1998년 2월 : KAIST 산업경영학과 학사
2000년 2월 : KAIST 산업공학과 석사
2009년 2월: KAIST 산업및시스템공학과 박사
2004년 5월~2010년 2월: 엔씨소프트 정보보안실장, Technical Director
2010년 3월~: 고려대학교 정보보호대학원 조교수
<관심분야> 온라인게임 보안, 네트워크 보안, 네트워크 포렌식

금 영 준 (Young Jun Kum)
학생회원

2008년 2월: 서울과학기술대학교 토목공학과 졸업
2012년 3월~현재: 고려대학교 정보보호대학원 석사과정
<관심분야> 시스템 해킹, 온라인게임 보안, 데이터 마이닝, 네트워크 포렌식