한국컴퓨터정보학회 동계학술대회 논문집 제25권 제1호 (2017. 1)

모바일 게임 보안을 위한 게임내 데이터 난독화에 관한 연구

김효남[O]
[O]청강문화산업대학교 게임전공
e-mail: hnkim@ck.ac.kr[O]

A Study on Obfuscation of the InGame Data for the Mobile Game Security

Hyo-Nam Kim[O]
[O]Dept. of Computer Game, ChungKang College of Culture Industries

● 요 약 ●

현재 국내 모바일 게임 시장 규모와 사용자들이 지속적으로 증가되고 있으며, 스마트폰을 이용하여 게임을 즐기는 시간도 계속 늘어나고 있다. 이런 시장 현황의 이면에는 모바일 게임 시장이 사이버 범죄의 진원지로 급부상하고 있다. 본 논문에서는 모바일 게임을 개발하는데 있어서 게임 내부에서 사용하고 있는 데이터들의 난독화 기술과 관련한 프로그램을 제안하여 게임의 원본 소스 데이터를 해킹으로부터 보호할 수 있는 게임 보안 기술을 제안한다.

키워드: Mobile Game, Obfuscation, Hacking

I. Introduction

현재 국내 모바일 게임의 시장 규모는 약 4조에 달하며, 스마트폰 보급률은 전체적으로 84%이며 20~30대는 99%의 보급률을 보이고 있다. 한국 사용자들의 하루 평균 게임 시간은 43분 정도이며, 스마트폰 사용자 중에 59% 정도는 게임을 8월 한달 동안 한 번 이상 접속한 기록이 있다[1]. 현재 시장 규모와 사용자들의 사용 현황 이면에는 모바일 게임 시장이 사이버 범죄의 진원지로 급부상하고 있다. 전문지식 없이 몇 번의 클릭을 통해 해킹할 수 있는 해킹 툴의 등장과 이에 대한 공유가 활발해지면서 모바일 게임 시장에 비상등이 켜졌다.

이런 문제점들을 안고 있는 환경에서 모바일 게임 해킹을 방지하기 위해서는 개발 과정에서 보안이 필수적으로 고려돼야 한다. 모바일 게임 해킹을 방지하는 대표적인 기술로 난독화(Obfuscation)를 꼽을 수 있다. 본 논문에서는 모바일 게임을 개발하는데 있어서 난독화 기술과 관련한 프로그램 예를 제안하여 게임의 원본 소스 데이터를 알아볼 수 없도록 하는 게임보안 기술을 제안한다.

II. The Main Subject

실시간으로 모바일 게임 해킹을 막기 위해 고려하고 적용해 볼 수 있는 방법들로 Obfuscation (난독화)와 Anti-decompile 기법들을 사용할 수 있다. 난독화는 Dotfuscator, Crypto Obfuscator, Unity3D Obfuscator 등 여러 기법이 사용된다. 이 방법들은 코드를 해석하기 어렵게 만드는 기법으로 메소드나 변수의 명칭을 읽기 어렵게 변경하거나, IL Code의 실행 순서를 뒤섞는 기법이다[2]. Anti-decompile 방법은 decompile 도구를 이용하여 IL Code를 얻지 못하도록 Assembly-CSharp.dll 파일을 조작하여 안전하게 하는 것이다. 본 논문에서는 게임 안에서 사용하는 데이터들을 보호를 위해서 난독화 방법으로 모바일 게임을 개발하는 과정에서 코드를 해석하기 어렵게 만드는 한 가지 방법을 제시하고자 한다.

모바일 게임에서는 기본적으로 어떤 게임이라도 프로세스 메모리 편집기를 사용한 해킹이 쉽게 이루어진다. 프로세스 메모리 편집기란 프로그램 실행 중 메모리에 있는 원본 DLL 덤프를 통해 메모리 내용을 해석하여 바꾸어서 프로그램의 동작을 변화시킬 수 있는

에디터이다.

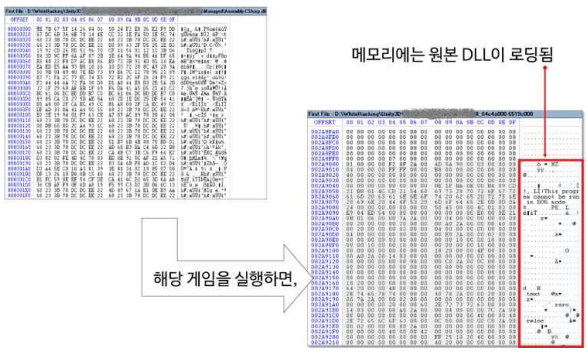

Fig 1. Memory Dump of Mobile Game DLL

난독화의 대표적인 방법은 디컴파일 시 주요 소스 코드를 알아볼 수 없도록 임의의 값으로 바꾸는 것이다. 개발 언어에서 통상적으로 사용되는 구문은 그대로 두고, 중요한 내용만 바꾸기 때문에 해커가 소스 코드를 손에 넣더라도 이 코드가 어떤 기능을 수행하는지 알아보기 어렵게 만든다.

게임 안에서는 다양한 데이터들을 이용해서 게임성을 높이려고 기획하는 부분이 있다. 여기서 게임에서 중요하게 사용하는 게임 캐시 데이터에 대해서 안전하게 사용하는 방법을 예로 들어보고자 한다. 게임 내의 캐시 조작으로 발생한 문제 사례를 보면 초기 게임 캐시가 100,000이 있다고 하자 메모리 내에서 100,000을 가리키고 있는 곳을 찾아서 해커는 감시하게 된다.

게임 내 상점에서 거래와 같은 행위를 하면서 캐시가 50,000으로 변화를 게임 내에서 관찰할 수 있다. 메모리 내에서 캐시가 상점에서 거래하기 전에 100,000이 상점에서 거래 후에 50,000이 되어 있는 곳을 알게 된다. 그리고 이후에 해커는 캐시가 가리키고 있는 곳을 지정하여 1,000,000으로 고쳐 써서 이득을 볼 수 있게 된다. 이 문제를 해결하기 위한 난독화 방법은 프로그램 내에서 캐시 변수 처리를 난수와 연계시켜 캐시 데이터를 알아볼 수 없게 하는 방법이다. 아래 난독화 방법이 적용된 프로그램을 제시해본다.

탐지부에서는 정규화가 완료된 링크정보가 정상인지 비정상인지 여부를 판단하게 되는데 이에 대한 판단기준은 그림 1의 중앙처리 부분에서 비정상 iFrame 탐지, 취약한 Object 탐지, 비정상 문자열 카운팅, 스크립트 앤트로피 계산, 휴리스틱 패턴 등을 사용하여 판단하게 되며 난독화 된 코드는 자동으로 Unpacking하여 악성여부를 판단하게 된다.

```
class GameCacheClass
{
 private Integer securityKey=0;
 private Integer haveCache=0;
 private Integer encryptCache(Integer cache)
 {
    securityKey=rand.nextInt(100000);
    return cache+securityKey;
 }
 private void displayCache (Integer cache,
  GameData data )
 {
  data.displayCache(cache-securityKey);
 }
 public void inGameCache(GameData data)
 {
    haveCache=encryptCache(data.getCache());
    displayCache(haveCache);
    saveCache(securityKey);
  }
}
```

Fig 2. Obfuscation Example of Ingame Data

난독화를 위하여 제시한고 있는 프로그램에서 캐시가 변동할 때마다 랜덤으로 할 필요가 있다. 해커가 캐시 변동치 폭을 감시하고 있어도 캐시의 상수 값이 알려지지 않도록 하기 위함이다.

III. Conclusions

본 논문에서는 사용자가 게임 콘텐츠로 위장한 악성링크를 클릭했더라도 사전에 악성링크 정보를 통해 사용자 소프트웨어의 최신 보안 업데이트를 설치해둔 경우라면 악성 파일이 다운로드 되지 않도록 하기 위하여 악성코드 탐지엔진에서 수집된 트래픽 정보로부터 악성링크를 판단할 수 있는 실시간 악성링크 탐지 기능을 제시하였다.

References

[1] http://www.inven.co.kr/webzine/news
[2] http://ndcreplay.nexon.com